

O'ZBEKSTAN RESPUBLIKASI JOQARI HA'M
O'RTA ARNAWLI BILIMLENDIRIW MINISTRILIGI

QARAQALPAQ MA'MLEKETLIK UNIVERSITETI

«A'meliy matematika» kafedrası

PROGRAMMALASTIRIW TIYKARLARI

pa'ninen lekciyalar teksti

5480100 "A'meliy matematika ha'm informatika" ta'lim bag'dari ushin

Kurs: I-II

Semestr: 1,2,3.

Duziwshiler:

Sh.Eshmuratov
Q.Xudaybergenov

Nokis – 2016

Annotaciya

Programmalaştırıw tiykarları pa'ni studentlerge qoyılǵan ma'seleni sheshuwshi komp'yuter programmasın du'ziwge u'yretiw bolıp tabıladı. Usı maqsette programmalaştırıw tilleri ha'm ortalıqları haqqında ulıwma tu'sinikler beriledi ha'm bul tillerden paydalanıwǵa u'yretiledi.

Lekciyada programmalaştırıwǵa kirisiwdin' teoriyalıq tiykari bolǵan algoritmlerge o'z aldına itibar qaratılǵan. Bul jerde algoritmlerdi klassifikatsiyalaw da keyingi waqıtlarda kompyuterde a'melge asırıw ushın za'ru'r bolǵan bir qatar matematikalıq tu'sinikler – ta'kirarlaw, ja'rdemshi algoritm, rekursiya, yad, massiv, indeks, parametr ha'm t.b. kiritilip, ha'r qıylı klass ma'selelerinin' algoritmleri du'ziledi.

MAZMUNI

1-Lekciya. Programmalastiriw tiykarlari pa'nine kirisiw. Pa'nnin' maqseti ha'm waziypalari.....	4
2-Lekciya. Algoritmalar ha'm onn' qasiyetleri. Algoritmnin' beriliw usillari.....	11
3-Lekciya. Algoritmni du'ziwdin' basqishlari'	24
4-Lekciya. Programmalastiriw tilleri. C\C++ tilleri.....	34
5-Lekciya. C++ tili ha'm woni'n' leksik tiykarlari	38
6-Lekciya: C++ tilinde berilgenler ha'm wolardi'n' tu'rleri.....	45
7-Lekciya: C++ tilinde a'meller ha'm ma'nis beriw operatori'.....	59
8-Lekciya: Razryadli, logikali'q ha'mde ta'kirarlaw a'melleri	64
9-Lekciya: Sha'rt ha'm tan'law operatorlari'	75
10-Lekciya: Ta'kirarlaw operatorlari'	91
11-Lekciya: break, continue ha'mde goto operatorlari'	104
12-Lekciya: Funktsiyalar.....	116
13-Lekciya: C++ tili funktsiyalari'ndag'i' mu'mkinshilikler	130
14-Lekciya: Rekursiv ha'm qayta ju'kleniwshi funktsiyalar.....	143
15-Lekciya: Ko'rsetkishler ha'm murajatlar.....	152
16-Lekciya: Massivler	170
17-Lekciya: Qatarlar	190
18-Lekciya: Qatarlar u'stinde a'meller.....	202
19-Lekciya: Qatarlarda funktsiyalar	209
20-Lekciya: Qatarlar u'stinde a'meller	221
21-Lekciya: Qatarlar menen isleytug'i'n funktsiyalar	228
22-Lekciya: A'piwayi' strukturalar	235
23-Lekciya: Dinamikali'q strukturalar	245
24-Lekciya: Makroslar	263
25-Lekciya: Fayllar ha'm ag'i'mlar.....	271
26-Lekciya: Woqi'w-jazi'w funktsiyalari'.....	279
27-Leksiya. Obyektke bag'darlang'an programmalastiriw texnologiyalari. Inkapsulyatsiya, Merasliq, Polimorfizm.	303
28-Leksiya. Klasslar. Klassti ha'm obektlardi sa'wlelendiriwi. Klass maydanlari ha'm metodlari.	327
29-Leksiya. Klass konstruktorlari ha'm destruktoralari	347
30-Leksiya. Dos funktsiyalar ha'm dos klasslar.....	359
31-Leksiya. Miyrasli'q. Apiwayi miyrasli'q. Koplikli miyrasli'q.....	368
32-Leksiya. Funktsiyalarni qayta juklew. Klasstin' funktsiya-ag'zalarin qayta juklew	386
33-Leksiya. Operatorlardin qayta juklew.....	397
34-Leksiya. Polimorfizm xa'm virtual funktsiyalar.....	423
35-Leksiya. Polimorfizm ha'm virtual funktsiyalar.....	436
36-Leksiya. Ayriqsha jag'daylar	446
37-Leksiya. Shablonlar	468

1-Leksiya. Programmalaştırıw tiykarları pa'nine kirisiw. Pa'nnin' maqseti ha'm waziypaları

Joba:

- 1. Programmalaştırıw tiykarları pa'ninin' maqseti**
- 2. Programmalaştırıw tili**
- 3. Pa'nnin' waziypaları**
- 4. Pa'n boyınsha studentler bilimine qoyılatug'ın talaplar**
- 5. Basqa pa'nler menen baylanısı**

Programmalaştırıw tili – du'zilg'en algoritmdi kompyuter a'melge asırıw ushın qural bolıp tabıladı. Bul orında tu'rli quramalılıqtag'ı sintaksis ha'm semantikag'a iye bolg'an tillerden paydalanıw mu'mkin. Pa'nde C++ tilinin' alfaviti, til qurılmalarının' ulıwma sintaksisi, berilgenler tu'rleri, operatsiyalar ha'm funktsiyalar qaraladı, patoklar menen islew, ko'rsetkishler, a'mellerdi qayta ju'klewhler, funktsiyalardı qayta ju'klew, qoliplar ha'm modulli programmalaştırıw ha'm grafikalıq moduli menen islew uyretiledi.

Ha'zirgi ku'ndegi programmalaştırıw tilleri eki toparg'a bo'linedi, birinishisine Assembler tili kirse, ekinshisine qalg'an barlıq tiller kiredi. Joqarı basqısh tilleri ol yamasa bul a'meliy ma'selelerdi (ekonomikalıq, matematikalıq ha'm basqalar) sheshiwge arnalg'an bolıp, olar bunday ma'selelerdi sheshiw algoritmin an'latiw quralların o'z ishine aladı. Assembler bolsa protsessor arxitekturasının' sa'wlesi, yag`nıy onın' logikalıq qurılmaları, jumıs hjag'dayları ha'm a'mel qılıw sxemaların o`zinde sa'wlelendiredi. Assemblerde jazılğ'an programma joqarı effektivligi, minimal ko'lemi ha'm orınlanıwının' maksimal tezligi menen ayrılıp turadı. Sonın' ushın Assembler tezlik ha'm yad sheklewler qo'yılğ'an jag'daylarda, qurılmalg'a “qattı” baylanisqan drayverler

jaratıwda ha'm kompyuterge standart emes apparat qurılmaların jalg'aw ma'selelerinde qol keledi. Assembler tilin u'yreniw metodikalıq ko'z qarastan ha'm a'hmiyetli bolıp tabıladı. Assembler tilin u'yreniw – protsessor, ha'm ulıwma kompyuter islewin tolıq u'yreniw, C/C++ ha'm basqa tillerde programma du'ziwsi ushın joqarı basqısh tiller formalizmleri “artında” qanday protsessler ju'z berip atırǵanın an'law imkaniyatın beradi.

Protsessordın' qorg'alg'an rejiminde (Windows OS da) a'mel qılıwshı 32-razryadlı protsessorlar printsipial jaqtan jan'a programmalaştırıw texnologiyaların ju'zege keliwine sebep boldı. Obektke bag'darlang'an programmalaştırıwǵa tiykarlang'an C++ usı studentlerge juwap beriwshi programmalaştırıw tili bolıp tabıladı. C++ tili tiykarında Visual C++ vizual programmalaştırıw ortalıǵı jaratıldı ha'm du'nyada o`zinin' salmaqlı ornına iye boldı. Visual C++ ortalıǵında programmalaştırıw obektke bag'darlang'an programmalaştırıwdın' teoriyalıq tiykarların tu'sindiriw, klasslar ha'm olar shejiresin payda etiw, vizual programmalaştırıw ortalıǵı, forma, komponentalar, prilojenieler jaratıw ha'm usı kibi tayanısh tu'sinikler haqqında mag'lıwmat beriw ha'm olar menen islew sabaqların beriw arqalı a'melg'e ashırıladı.

Programmalaştırıwǵa u'yretiw sa'ykes ma'selelerge tayang'an jag'dayda alıp barıladı. Pa'nnin' a'meliy bo'liminde algoritmler, C++ ha'm Assembler programmalaştırıw tili, Visual C++ ortalıǵında islew boyınsha ma'selelerdi o`z ishine aladi.

Programmalaştırıw tiykarlari pa'ni tikkelew ha'm tuwrıdan tuwtı sistemalı programmalaştırıw, mag'lıwmatlar bazasın basqarıw sistemaları, kompyuter grafikasi ha'm kompyuter tarmaqlari pa'nleri menen u'zliksiz

baylanisli ha'm bul pa'n komp'yuter texnologiyalari boyinsha qa'niyge tayarlawda uliwma tiykar rolin oynaydı.

Pa'ndi oqıtıwdan maqset – a'meliy matematika ha'm informatika bag'darının' bakalavr basqıshı studentlerine programmalastırıw tiykarların jeterli da'rejede oqıtıw, usı bilimlerge tayang'an halda kompyuter modellestiriwge keletug'ın a'meliy ma'selelerdin' programmalıq ta'miynleniwin a'melge asırıwg'a u'yretiw ha'm qa'niygelik pa'nların o'zlestiriwde tayanış bilimlerge iye bolıw.

Pa'nnin' wazıypaları: Masala sheshiwdin' algoritmik tiykarların u'yreniw, kompyuter islewinin' printsipi, programmalastırıw tillerin klassqa bo'liw, kompyuterde berilgenler ha'm buyırıqlardın' su'wreniwi, C++ tilinde programmalastırıw, Assembler tili, obektke bag'darlang'an programmalastırıw texnologiyaları, vizual programmalastırıw ortalıg'ında islew bul pa'nnin' tiykarı wazıypaları bolıp esaplanadı.

Pa'n boyınsha studentler bilimne qoyılatug'ın talaplar

«Programmalastırıw tiykarları» pa'nin o'zlestiriw barısında a'melg'e asırılatug'ın ma'seleler toparında bakalavr:

C++ programmalastırıw tili. Til alfaviti, sintaksisi ha'm semantikasi, programma du'zilisi, berilgenlerdin' tiykarı tu'rleri, ma'nis beriw, tarmaqlanuwsı ha'm ta'kirarlanıwsı operatorlar, funktsiyalar, ko'rsetkishler, oqıw-jazıw patokları, ayırıqsha jag'daylardı qayta islew, klasslar, obektke bag'darlang'an programmalastırıw texnologiyaları, konteynerlardı tu'siniwi ha'm olardı qollag'an halda a'meliy ma'selelerdi sheshe alıwı kerek;

Assembler tili. Protsessor, registr ha'm shina tu'siniklari, real ha'm qorg'alg'an rejim arxitekturalari, programmalarin' segment du'zilisi, stek, u'ziliwler sistemasi, oqiw-jaziw sistemalari, Assembler makroqurallari, LDT ha'm GDT tablitsalari, qorg'alg'an rejimde adreslew usillari, qorg'alg'an rejimde u'zilislerdi qayta islew, Windows prilojeniesin jaratiw haqqinda uliwma tu'sinikke iye boliw ha'm arnawli jag'daylar ushin assembler programmasin jarata oliwi kerek;

Obektke bag'darlang'an ha'm vizual programmalastriw. Klasslar, klass maydanlari ha'm metodlari, konstruktor ha'm destruktorglar, miyras, virtual metodlar, prilojeniyeg jaratiw ustasi, konsol ha'm dialog prilojeniyeleri. Ko`p aynali ha'm bir aynali prilojenieler, xabarlar ha'm buyriqlar, WIN32 sistemasi xabarlar kartasin isletiw, grafikaliq interfeys penen islew, NET texnologiyasi, C# tilinin' o'zgesheliklerin biliwi ha'm programmalastriw ortalig'ında islew aliwı ha'm Windows prilojenielerin jarata aliwı kerek.

Basqa pa'nler menen baylanısı: Programmalastriw tiykarlari pa'ni tikkelew ha'm tuwrıdan tuwrı sistemalı programmalastriw, informatika, obyektke bag'darlang'an programmalastriw, mag'liwmatlar bazasin basqariw sistemalari, kompyuter grafikasi ha'm kompyuter tarmaqlari pa'nleri menen u'zliksiz baylanıslı ha'm bul pa'n komp'yuter texnologiyalari boyınsha qa'niyge tayarlawda uliwma tiykar rolin oynaydı. Bul oqiw pa'ni matematika, matematikalıq logika ha'm statistika, matematikalıq modellestiriw, pa'nleri ha'm basqa da tabiyg'ıy pa'nler menen ha'm u'zliksiz baylanıslı.

Pa'nnin' metodikalıq ta'repten u'zliksiz izbe-izligi: Tiykarg'ı bo'limde pa'nnin' temalari logikalıq tu'rde izbe-izlikte keltiriledi. Ha'r bir

temanın' mazmunı tiykarg'ı tu'sinikler ha'm tezisler arqalı ashıp beriledi. Bunda tema boyınsha studentlerge DTS tiykarında jetkeriliwi kerek bolg'an bilim ha'm ko'nlikpeler toliq iyelep alınıwı kerek.

Tiykarg'ı bo'lim sapasına qoyilatug'ın talaplar temalardın' aktuallıg'ı, olardın' jumıs beriwshiler talapları ha'm o'ndiris mu'tajliklerine sa'ykesligi, ma'mleketimizde bolıp atırg'an sotsiallıq-siyasiy ha'm demokratiyalıq o'zgerisler, ekonomikanı erkinlestiriw, ekonomikalıq-huqıqiy ha'm basqa tarawlardag'ı reformalardın' baslı ma'selelerin qamtıp alıwı ha'mde pa'n ha'm texnologiyalardın' son'g'ı jetiskenlikleri itibarg'a alınıwı usınıs etiledi.

Pa'nnin' isleb shig'ariwdag'i o'rni: Usi' bag'darlama tiykarında bul pa'n ishinde ko'plep model maseleler u'yrenilip, bul pa'ndi teren' u'yrengen har bir bakalavr alg'an bilim ha'm ko'nikmelerin ilmiy-izertlew islerinde, bank diziminde, islep shig'ariwdin' turli oblastlarında, sonday-aq, ta'lim diziminde sapalı' paydalaniw imkanin beredi.

Programma mashina kodlari'ni'n' qanday da izbe-izligi boli'p, ani'q bir yesaplaw qurali'n a'mel yetiwin basqaradi. Programma ta'miynati'n jarati'w procesin an'satlasti'ri'w ushi'n ju'zlegen programmalasti'ri'w tilleri jarati'lg'an. Barli'q programmalasti'ri'w tillerin yeki kategoriyag'a aji'rati'w mu'mkin:

- ✓ to'men da'rejedegi programmalasti'ri'w tilleri;
- ✓ joqari' da'rejedegi programmalasti'ri'w tilleri.

To'men da'rejedegi programmalasti'ri'w tillerine Assembler tu'rindagi tiller kiredi. Bul tillerdi sali'sti'rg'anda qi'sqa ha'm operativ atqari'li'wshi' kodlardi' jarati'w mu'mkinshiligini beredi. Lekin, Assembler tilinde programma du'ziw qi'yi'n, sali'sti'rg'anda uzaq dawam yetetug'i'n

proces boli'p tabi'ladi'. Bug'an keru tu'rde joqari' basqi'sh tilleri jarati'lg'an boli'p, wolar da ta'biy tildin' sheklengen ko'rinisinden paydalang'an halda programma du'ziledi. Joqari' basqi'sh tilleridegi operatorlar, berilgenlerdin' tu'rleri, wo'zgeriwshiler ha'm programma jazi'wdi'n' ha'r qi'yli' usi'llari' tildin' an'lati'w mu'mkinshiligin asi'radi' ha'm programmanin' «o'qimishli» boli'wi'n ta'miyinleydi. Joqari' basqi'sh tillerine Fortran, PL/1, Prolog, Lisp, Basic, Pascal, C ha'm basqa tillerdi mi'sal keltiriw mu'mkin, Kompyuter arxitekturasinin' rawajlani'wi, kompyuter tarmag'i'ni'n' rawajlani'wi' saykes tu'rde joqari' basqi'sh tillerinin' jan'a variantlari' ju'zege keliwine, jan'a tillerdin' payda boli'wi'na, ayi'ri'm tillerdin' jog'ali'p ketiwine ali'p keldi. Ha'zirde ken' tarqalg'an tillerge Object Pascal, C++, C#, PHP, Java, ASP tilleri yesaplanadi'. Atap aytqanda, C tilinin' rawajlang'an varianti' si'pati'nda C++ tilin ali'wi'imiz mu'mkin. 1972 ji'lda Denis Ritch ha'm Brayan Kernegi ta'repinen C tili jarati'ldi'. 1980 ji'lda Byarn Straustrop C tilinin' a'wladi C++ tilin jaratti'p, wol jag'dayda strukturali' ha'm obyektga jo'neltirilgen programmalasti'ri'w texnologiyasi'na tayangan halda programma jarati'w mu'mkinshiligi tuwi'ldi'.

High-Level Languages

There are many languages for writing programs. In this text we will discuss the C++ programming language and use it to write our programs. C++ is a high-level language, as are most of the other programming languages you are likely to have heard of, such as C, C#, Java, Python, PHP, Pascal, Visual Basic, FORTRAN, COBOL, Lisp, Scheme, and Ada. High-level languages resemble human languages in many ways. They are designed to be easy for human beings to write programs in and to be easy for human beings to read. A high-level language, such as C++, contains instructions that are much more complicated than the simple instructions a computer's processor (CPU) is capable of following.

Compilers

A program that translates a high-level language like C++ to a machine language is called a **compiler**. A compiler is thus a somewhat peculiar sort of program, in that its input or data is some other program, and its output is yet another program. To avoid confusion, the input program is usually called the **source program** or **source code**, and the translated version produced by the compiler is called the **object program** or **object code**. The word **code** is frequently used to mean a program or a part of a program, and this usage is particularly common when referring to object programs. Now, suppose you want to run a C++ program that you have written. In order to get the computer to follow your C++ instructions, proceed as follows. First, run the compiler using your C++ program as data. Notice that in this case, your C++ program is not being treated as a set of instructions. To the compiler, your C++ program is just a long string of characters. The output will be another long string of characters, which is the machine-language equivalent of your C++ program. Next, run this machine-language program on what we normally think of as the data for the C++ program. The output will be what we normally conceptualize as the output of the C++ program. The basic process is easier to visualize if you have two computers available, as diagrammed in Display 1.4. In reality, the entire process is accomplished by using one computer two times.

2-Leksiya. Algoritmlar ha'm onin' qasiiyetleri. Algoritmnin' beriliw usillari

Joba:

1. Ma'sele sheshiw basqishlari
2. Ma'selenin' qoyiliwi
3. Sanli usildi tan'law
4. Esaplaw basqishlarin algoritmlaw
5. Kompyuter programmasin du'ziw
6. Algoritm tu'sinigi
7. Algoritmnin' beriliw usillari

Ma'sele sheshiw basqishlari: H'ar tu'rli ma'selelerdi jeke kompyuterlerde tuwridan-tuwri sheshiw basqishlari to'mendegishe beriledi:

1. Ma'selenin' qoyiliwi
2. Sanli usildi tan'law
3. Esaplaw basqishinin' algoritmin du'ziw
4. Kompyuter programmasin du'ziw
5. Programma u'stinde islew
6. Ma'seleni jeke kompyuterde sheshiw ha'm onin' na'tiyjesin talqilaw

Ma'selenin' qoyiliwi: Ha'r tu'rli ma'selelerdi sheshiwdin' da'slepki basqishi bolip, onin' qanday qoyiliwi esapqa alinadi.

Ma'selenin' qoyiliwi dep-en' aldin oni aniq ha'm toliq an'latiw kerek, ha'm oni duris sheshiw ushin qoyilip atirg'an ha'mme sha'rtlerdi aniqlaw tu'siniledi. Ma'seleni sheshiw ushin kerek bolg'an qosimsha mag'liwmatlar boliwi kerek.

Sanli usildi tan'law: Sanli matematikaliq usillar esaplaw basqishlarin elementar operatsiyalarg'a ajiratiwg'a ja'rdem beredi. Yag'niy qa'legen ma'seleni sheshiwshi operatsiyalardi orinlaw izbe-izligine alip keledi. Olardin' quramina arifmetikaliq operatsiyalar (\geq , $<$, $>$, \leq , $=$) kiredi. Aniq bir ma'seleni sheshiw ushin sonday sanli usildi tan'lawimiz kerek, ol qoyilg'an ma'seleni toliq sheshiwi kerek. Belgili bir sanli usildi tan'lawda to'mendegishe sha'rtler boliwi mu'mkin: esaplawlar na'tiyjesinin'

aniqlig'i, jeke kompyuterde ma'seleni sheshiw waqiti, talap qilinip atirg'an yadtin' ko'lemi ha'm basqalar.

Esaplaw basqishlarin algoritmlaw: Algoritmlew basqishi ma'seleni sheshiw din' algoritmin du'ziwdi ta'miyn- leydi, baslang'ish mag'liwmatlardi na'tiyjelilikke aylandiriw basqishlarinin' izbe-izligin aniqlaw ha'm an'latiw ushin xizmet qiladi. Ma'seleni sheshliw algoritmi ku'tilip atirg'an na'tiyjeni aliw ushin, da'slepki mag'liwmatlar ha'm araliq na'tiyjeler u'stinen orinlanatug'in elementar operatsiyalar ta'rtibin aniq belgilep beriwi kerek. Algoritm ma'seleni sheshiw usili esaplanbaydi, tan'lang'an sanli usildi a'melge asiradi. Eger bir sanli usildi a'melge asiriw ushin, bir neshe algoritmler bar bolsa, onda olar arasinan jeke kompyuterden en' jaqsi paydalaniw ushin xizmet qilatu'g'in usil tan'lap alinadi. (quraminda operatsiyalar bari ha'm az yad ko'lemin talap etetug'in)

Kompyuter programmasin du'ziw:

Programmaw-bul jeke kompyuter programmaw tilinde algoritmdi ko'rsetiw basqishi. Eger da'stu'r kompyuter tilinde jazilg'an bolsa, onda da'stu'r tuwridan-tuwri jeke kompyuterde a'melge asiriladi. Da'stu'r simvolliq tu'rde jazilsa, onda da'stu'r tekstli translyator arqali, kompyuter tiline awdariladi ha'm jeke kompyuterde sheshiledi.

Da'stu'r di tekseriw: Da'stu'rdegi qa'teliklerdi aniqlaw ha'm olardi tuwriw basqishi tekseriw delinedi. Da'stu'rlerde sintaksislik ha'm logikaliq qa'telikler boliwi mu'mkin. Translyator ta'repinen da'stu'r di sintaksis baqlaw na'tiyjesinde programmaw astiriw tili ta'repinen olardi du'ziwde ruxsat etilmegen tu'rli konstruktsiyalar aniqlanadi. Sintaksis qa'teliklerdin' barlig'i da'stu'r din' duris islewine mu'mkinshilik bermeydi. Sintaksis qa'teliklerdi durislag'annan keyin da'stu'r shi kompyuterde aling'an na'tiyjeler tiykarinda duris islep atirg'anin aniqlawg'a boladi. Eger aling'an na'tiyjeler ku'tilgen na'tiyjeler menen sa'ykes kelse, onda da'stu'r di jetilistirip ja'ne bir ma'rte tekserip shig'iw kerek. Eger na'tiyjeler qa'te bolip shiqsa, onda biz ju'da u'lken logikaliq qa'teliklerge jol qoyg'an bolamiz.

Logikaliq qa'teliklerge tiykarinan da'stu'r din' ideasinin' jobasin du'zgende jol qoyiladi. Ma'selenin' ideyasin talqilag'anda, ma'selenin'

algoritmin du'ziwde yamasa programmanin' o'zin programmastiriw tilinde jaziw waqtında program -mist ta'repinen onin' ma'nisi boyınsha qa'telikke jol qoyiliwi mu'mkin. Sonin' ushin da'stu'rđi islewđin' birinshi basqishina qoyilg'an ma'selelerdi,talaplardı jaqsilap u'yreniw kerek. Logikaliq qa'teliklerdin' bar ekenligi da'stu'rđi ja'ne bir ma'rte tekserip ko'riwđi talap etedi.Da'stu'r tekserilip bolg'annan keyin,oni jeke kompyuterde tekserip ko'riw kerek.

Algoritm tu'sinigi: EEMge tapsırma tayarlap atırg'anda esaplaw protsessin yaki EEMde orınlanatug'in qa'legen basqa a'meller izbe-izligin anıq ha'm tolıq su'wretlew za'ru'riyatı payda boladı. A'ne usı su'wretlew sheshiw algoritmi menen beriledi. Ma'seleni sheshiw algoritmin izlew, islep shıg'ıw ha'm su'wretlew algoritmlaw delinedi.

Algoritm so'zi ha'm tu'sinigi IX a'sirde jasadp do'retiwshilik etken ullı oyshıl Muhammed al-Xorezmiy atı menen baylanıslı. Algoritm so'zi Al-Xorezmiy atın Evropa alımları ta'repinen buzıp aytılıwınan ju'zege kelgen. Al-Xorezmiy birinshi bolıp onlıq sanaq sistemasın'ın' printsiplerin ha'm ondag'ı to'rt a'mellerdi orınlaw qag'ıydaların tiykarlap bergen. Algoritm degende qandayda bir maqsetke erisiwge eki berilgen tu'rdegi qa'legen ma'selenin' sheshiliwin ta'miynleytug'in ko'rsetpelerdin' (buyruqlardın') anıq, tu'sinerli, shekli ha'mde tolıq sisteması tu'siniledi.

Algoritm tu'sinigi

Algoritm — orınlawshı ushın ma'lim bir ma'seleni sheshiwge qaratilg'an ko'rsetpelerdin' anıq izbe-izligi.

«Algoritm orınlawshısı».

Algoritm orınlawshısı— algoritmde ko'rsetilgen buyruqlardı orınlay alatug'in abstrakt yaki real (texnik, biologik yaki biotexnik) sistema.

Algoritmınin' bes tiykarg'ı qa'siyeti bar.

1. Diskretlilik. Bul qa'siyet algoritmlerdi barlıq waqıtta shekli adımlardan ibarat etip bo'leklew imkaniyatı barlıg'ında, yag'niy onı shekli sandag'ı a'piwayı ko'rsetpeler izbe-izligi formasında an'latıwda.

2. Tu'siniklilik. Algoritm onı orınlawshısına (adam yaki mashinag'a) tu'sinikli mazmunda bolıwı sha'rt, bolmasa orınlawshı a'piwayı g'ana a'meldi de orınlay almawı mu'mkin. A'dette ha'r bir orınlawshının' (durısırag'ı olardıń tu'rlerinin') orınlay alıwı mu'mkin bolg'an ko'rsetpeler yaki buyırıqlar ortalıg'ı bar, ol orınlawshının' ko'rsetpeler sisteması delinedi. Demek, orınlawshı ushın berilip atırg'an ha'r bir ko'rsetpe orınlawshının' ko'rsetpeler sistemasına tiyisli bolıwı kerek.

3. Anıqlılıq. Orınlawshıg'a berilip atırg'an ko'rsetpeler anıq mazmunda bolıwı za'ru'r. Sebebi ko'rsetpedegi anıqsızlıq mo'lsruherdegi maqsetke erisiwge alıp kelmeydi.

4. Ulıwmalıq. Usı qa'siyeti berilgen algoritmnin' baslang'ish mag'lıwmatlarının' ruxsat etilgen iqtıyarıy ma'nislerine an'latadı. Basqasha aytqanda, algoritm qandayda bir klassqa tiyisli ma'selelerdi sheshiwge arnalg'an bolıwı ha'm baslang'ish mag'lıwmatlardın' o'zgeritiw arqalı usı klassqa tiyisli turli ma'selelerdi sheshe alıwı kerek.

5. Na'tiyjelilik. Ha'r bir algoritm shekli sandag'ı adımlardan son' a'lbette natiyje beriwı sha'rt. Eger ko'rilip atırg'an protsess sheksiz dawam etip natiyje bermese, onı algoritm dep atay almaymız.

Qoyilg'an bazi bir ma'seleni sheshiw ushin aldın onin' EEM da matematikaliq modelin, keyin algoritimin ha'm programmasın du'ziwge boladı. Algoritm aldımızg'a qoyilg'an ma'seleni sheshiw ushin kerek bolg'an a'meller izbe-izligi. Misali, kvadrat ten'lemenı sheshiw ushin to'mendegi a'meller izbe-izligi orınlaniwı kerek.

1) a, b, c koefiyetsiyentler berilgen bolsin. Berilgen a, b, c koefiyetsiyentler ja'rdeminde $D = b^2 - 4ac$ (diskriminant) esaplanadı.

2) $D > 0$ bolsa eki koreng'e iye boladı

3) $D < 0$ bolsa haqiyqiy sheshimi joq ekenligin bildiredi

Ja'ne bir misal retinde a, b, c ta'repleri boyınsha u'shmu'yeshliktin' maydanin Geron formulasi boyınsha tabamız.

1. a, b, c —u'shmu'yeshliktin' ta'replerinin' uzinliqlari;

2. $P = (a + b + c) / 2$ —perimetrdin' yarimi esaplansin.

3. $T = P(P - a)(P - b)(P - c)$ esaplansin

4. $S = \sqrt{T}$ esaplansin.

Algoritm so'zi ha'm tu'sinigi **IX a'sirde jasap**, ilmiy miynet etken ulli ulama **Muhammed al-Xorezmiy** ati menen baylanisli. Algoritm so'zi Al-Xorezmiy atin Evropa alimlari ta'repinen basqasha aytiliwinan kelip shiqqan. Al-Xorezmiy birinshi bolip onliq sanaq sistemasinin' belgilerin ha'm onliq sanaq sistemasi boyinsha to'rt arifmetik a'mellerdi orinlaw qag'iydalarin tiykarlap bergen. Algoritmdi an'latqanda, oni ko'rgezbeli, aniq ha'm kompakt ko'riniske keltiriw ha'm a'meliyatta qollaw ushin qolayli jag'dayda an'latiw talap etiledi.

Algoritmdi an'latiwda tiykarinan u'sh tu'rli usildan paydalaniladi:

1. Algoritmdi so'z arqali an'latiw
2. Algoritmdi operator ko'riniside an'latiw
3. Algoritmdi grafik ko'riniste, yag'niy blok-sxemalar ko'riniside an'latiw.

Algoritmdi so'z arqali an'latiwg'a misal ko'rsek:

1-misal. A,B,C,D ha'm x sanlari berilgen. Eger $Cx - D \neq 0$ sha'rt berilgen sanlar ushin orinli bolsa, Y tın' to'mendegi ma'nisleri esaplansin:

$$y = (Ax+B):(Cx-D);$$

Sheshiw: Da'slep Y funktsiyasin esaplaw algoritimi tu'sinikli boliwi ushin so'z arqali ha'm formula ko'rinisine keltiremiz:

1-keste:

So'z arqali berilgeni	Formula ko'rinisi
1) A ni X qa ko'beytip, na'tiyje R menen belgilensin;	1. $A * x = R$
2) R di B g'a qosip, na'tiyje R menen belgilensin;	2. $R + B = R$
3) S ti x qa ko'beytip, na'tiyje R1 menen belgilensin;	3. $S * X = R1$
4) R1 den D ni alip, na'tiyje R1 menen belgilensin;	4. $R1 - D = R1$
5) R di R1 ge bo'lup, na'tiyje Y dep belgilensin;	5. $R / R1 = Y$

Bul misal esaplawg'a tiyisli ma'seleler sheshiw dın' en' a'piwayi algoritmi. Algoritm dın' so'zler arqali beriliwi dim onday ko'rgezbeli ha'm qolayli bolmag'anliqtan oni an'latiwda tiykarinan grafik usilda yaki blok-sxema ko'riniside an'latiw ken' tarqalg'an.

Algoritmnin' beriliw usılları ha'r tu'rli, biz solardan en' ko'p ushıraytug'ınları menen tanısamız.

1. Algoritmnin' so'zler arqalı an'latılıwı. Bunda orınlawshı ushın beriletug'ın ha'r bir ko'rsetpe tabiiy til arqalı buyırıq mazmunıda beriledi.

2. Algoritmnin' formulalar arqalı beriliwi. Algoritm matematikalı'q (esaplaw) formulalar sisteması arqalı beriliwi mu'mkin. A'dette bul usıldan matematika, fizika, ximiya kibi anıq pa'nlerdi u'yreniwde ko'plep paydalanamız.

3. Algoritmin' tablitsa ko'riniste beriliwi. Ma'selen, mektepte qollanıp kelinetug'ın to'rt qanalı matematikalıq tablitsalar yaki tu'rli lotareyalardıń utıslar tablitsaları.

4. Algoritmnin' programma formasında an'latılıwı.

5. Algoritmnin' arnawlı tilde ko'rinisleri. Bunday tiller algoritmlik tiller delinedi ha'm onda algoritmdi bir qıylı ko'riniste ha'm anıq an'latılıwı, orınlaw ushın qollanatuğ'ın belgilew ha'm qag'ıydalar tiykarında jazıw mu'mkin.

6. Algoritmlerdi grafik formada suwretlew. Algoritmnin' bul forması bizge a'welden tanıs, sebebi matematika kursında sızılğ'an grafiklerdin' ko'pshiligi algoritmnin' grafik usılda beriliwine misal boladı.

Amelde qollanatuğ'ın algoritmlik tillerdin' ko'pshiligi algoritmlerdi jazıwdın' formula-so'z usılına ju'da jaqın. Bunda bir bo'lim ko'rsetpeler matematikalı'q formulalar ja'rdeminde basqa bo'limi a'piwayı so'zler ja'rdeminde beriledi. Ma'selen, to'mende berilgen eki natural m ha'm n sanlardın' en' u'lken ulıwma bo'liwshisin (EUUB) tabıwdın' algoritmi adımları keltirilgen:

Eki sandı kiritin';

Egerde bul sanlar ten' bolsa, onda olardan birin juwap sıpatında alın' ha'm jumıstı toqtatın', keri jag'dayda bolsa dawam etin';

Eki san ishinde u'lkenin anıqlan';

U'lken ha'm kishi sanlardın' ayırmasın u'lken san t menen almastırın';

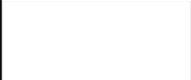
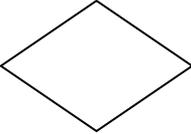
Algoritmdi 2-adımnan baslap qaytarın'.

Keltirilgen algoritmda ha' r qanday natural sanlardin' EKUB in tabiw ushin isletiw mu' mkin.

1. $A=n, B=m$.
2. Eger $A=B$ bolsa, onda 5 aimg'a o'tilsin.
3. Eger $A>B$ bolsa, onda $A=A-B$, keru jag'dayda $B=B-A$ dep alinsin.
4. 2- adimg'a o'tilsin.
5. $EKUB=A$ ha'm esaplaw toqtatilsin.

Algoritmnerdi jaziwda blok-sxema dep atalwshu grafik ko'rinisten paydalanw mu' mkin. Blok-sxemada algoritmnin' ha' r bir adımı arnawlı geometrik formalar menen belgilenedi ha'm onin' ishine a' piwayı a' meller jaziladi. Algoritmnin' orinlanw bag' itin bag' itlawshu sızıqlar ja' rdeminde belgilenedi.

To' mende biz **blok-sxemalardin' tiykarg' i elementleri** menen tanisamiz.

	algoritmnin' baslanıwı yaki tamamlanıwın belgileydi
	bag' itlang' an sızıq, blok-sxemadag' i ha' reket bag' itın ko' rsetedi.
	To' rtmu' yesh, ma' nis beriw yaki tiyisli ko' rsetpelerdi orinlaw protsessin belgileydi
	romb, sha' rtti tekseriwdi belgileydi, onin' bag' itlawshıları bolıp, tarmaqlar boyınsha biri "awa" (sha' rt orinlag' anda), ekinshisi "yaq" (sha' rt orinlanbag' anda) hareket bag' itların beredi.
	parallelogramm, mag' liwmatlardı kiritiw yaki shıg' arıwdı belgileydi.

Algoritm du' ziwge qoyilg' an tiykarg' i talaplar

Algoritm blok-sxemalar arqali ta' riyplengende, har bir blok ishinde orinlanatug' in a' mel jaziladi. Bloklar o' zara sızıqlar arqali birlestiriledi. Bul sızıqlar esaplaw bag' darin belgileydi. Izbe-izlikti aniq ko' rsetiw

ma'qsetinde ha'r bir bloktin' joqari shep sizig'inda nomer qoyiw ha'm mu'mkin. [3-6,8]

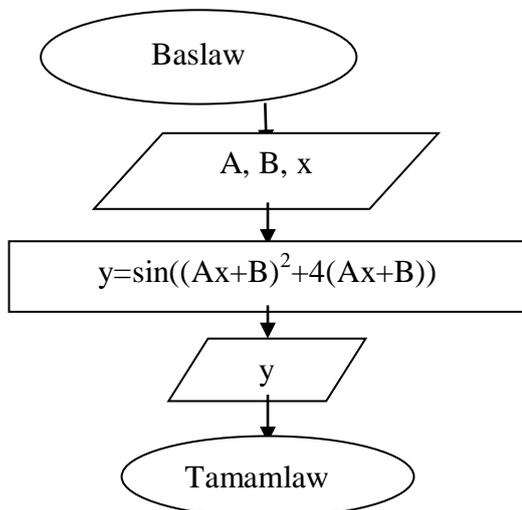
Algoritm tu'rleri

Siziqli algoritmler: Ha'r qanday quramali algoritmler a'melde u'sh tu'rli strukturann' kombinatsiyasinan ibarat boladi. Bular siziqli, tarmaqlaniwshi ha'm ta'krarlaniwshi algoritmlar boladi. Olardın' en' apiwayisi siziqli algoritmler bolip esaplanadi. Siziqli algoritmlar a'meller izbe-izligin payda qilip, esaplaw protsessinde bul a'meller ta'rtibi boyinsha tek g'ana bir ret orinlanadi. A'melde siziqli algoritmler ju'da kem ushiraydi. Bunday algoritmlerden, tiykarinan, u'lken formulalardi joqari aniqliqta esaplawda paydalaniw mu'mkin. Siziqli algoritmge misal ko'reyik. Berilgen $y = \sin((Ax+B)^2 + 4(Ax+B))$ funktsiya ma'nisin esaplaw talap qilinsin. Bul ma'seleni sheshiw protsessin bir neshe basqishqa bo'lip, onun' algoritmin so'zler arqali ta'riyplew usuli menen ha'm blok-sxema ko'rinishinde ta'riypleymiz:

1) So'z arqali' tariyplew algoritmi:

1. A, B, x - o'zgeriwshilerdin' ma'nisin kiritiw;
2. $Z = Ax + B$ esaplaw;
3. $S = Z^2$ esaplaw;
4. $T = 4Z$ esaplaw;
5. $R = S+T$ esaplaw;
6. $y = \sin(R)$ esaplaw;
7. Natijje – "y" di shig'ariw.

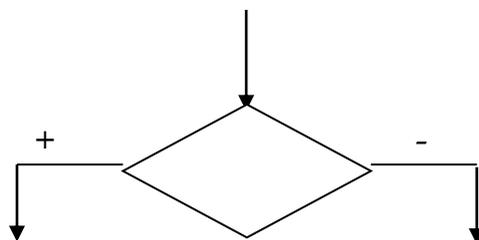
2) Blok-sxema ko'rinishidegi algoritm:



Keltirilgen barliq basqishlar arifmetik a'mellerdi izbe-iz orinlawdi ha'm har bir natiyje keyingi esaplawlar ushin isletiliwin belgilep beredi.

Tarmaqlaniwshi algoritmler

Tarmaqlaniwshi algoritmler o'z ishinde bir yaki bir neshe sha'rtti tekseriwshi bloktin' qatnasiwi menen xarakterlenedi. Bul sha'rttin' orinlaniwin yaki orinlanbawinin' na'tiyjesine qarap ol yaki bul a'meldin' orinlawin ta'miynleydi. Sonin' ushin sha'rttin' orinlaniwina qarap esaplaw protsesi a'lbette tarmaqlanadi. Sol menen birge tarmaqlaniwshi algoritm toliq yaki qisqa boliwi mu'mkin.



Tarmaqlaniwdin' toliq formasi. Tarmaqlaniwdin' qisqa formasi

Tarmaqlaniwshi algoritmlerdin' xarakterli ta'repi bazi bir a'meller blok-sxemada keltirilgen sha'rttin' orinlaniwi yaki orinlanbawina qarap aniqlanadi. Tarmaqlaniwshi algoritmge misallar ko'rip shig'amiz.

1-misal. A,B,C sanlari berilgen bolsin. Bul sanlar ishinde en' u'lkeninin' kvadrati tabilsin.

Sheshiw: Bul misaldi sheshiw ushin eki dana san A ha'm B salistiriladi.

Son' aling'an natiyjege qarap, yag'niy olardin' u'lkeni u'shinshi san bolg'an C menen salistiriladi. Eger S sani salistirilg'an sannan u'lken bolsa, onda S sani kvadratqa asiriladi, kerisinshe jag'dayda salistirilg'an sannin' o'zi kvadratg'a asiriladi.

Ma'seleni sheshiw algoritmi:

1) So'zler arqali algoritmdi ta'riyplew.

1. A,B,C ma'nislerin beriw.

2. $y = A$

3. eger $y \geq B$ bolsa, 5 ge barin'.
4. $y = B$
5. $y \geq C$ bolsa, 7 ge barin'.
6. $y = C$
7. $y = y^2$
8. "y" ti shıg'arın'.
9. esaplawdı tamamlan'.

Ta'kirarlaniwshi algoritmler

A'meliyatta quramalı protseslerdi programmalaştırıwda belgili bir buyırıqlar izbe-izligin sha'rtler tiykarında qayta orınlaw za'ru'riyati tuwiladi. Belgili bir o'zgeriwshinın' tu'rli ma'nislerinde belgili bir buyırıqlar sistemasınin' bazi bir nizamliliqqa tiykarlanip ta'krarlanip orınlaniwi ta'krarlaniwshi esaplaw protsesi (tsikl) dep ataladi. Ma'seleni sheshiw protsesinde bir neshe ma'rte orınlanatug'in operatorlar topari tsikl yaki tsiklii da'stu'r (ta'krarlaniwshi) deb ataladi. Ta'krarlaniwshi esaplaw protsesinin' ta'krar esaplanatug'in bo'limin ta'krarlaniwdın' denesi dep ataladi. Ta'krarlaniw ishinde ma'nisleri o'zgerip baratug'in o'zgeriwshini ta'krarlaniw o'zgeriwshisi yaki ta'krarlaniwdın' basqariwshi o'zgeriwshisi (tsikl parametri) deb ju'rgiziledi. Ta'krarlaniwshi protsestin' algoritmi ulıwma jag'dayda to'mendegilerdi o'z ishine aliwi kerek:

Ta'kirarlaniwlar jeke kompyuterde ma'selelerdi algoritmllestiriw ha'm programmalaştırıwdın' tiykarin saladı. Ta'kirarlaniwshi algoritmler to'mendegilerge bo'linedi:

- 1) Ta'kirarlaniw denesinin' tu'ri boyınsha;
- 2) Ta'kirarlaniw basqishin basqariw qa'siyetine baylanisli;

Ta'kirarlaniw denesi algoritmi siziqli, tarmaqlaniwshi ha'm ta'kirarlaniwshi bolıwi mu'mkin. Algoritmınin' tu'rine baylanisli ta'kirarlaniwshiliq denesindegi

Ta'kirarlaniwshi algoritmler a'piwayi ha'm quramali bolıwi mu'mkin.

Eger ta'kirarlaniwdın' denesi siziqli yaki tarmaqlang'an esaplaw basqishlarinan ibarat bolsa, onda ol **a'piwayi ta'kirarlaniwshi algoritm** delinedi.

Eger ta'kirarlaniwdin' denesi ta'kirarlaniwshi esaplaw basqishlarinan ibarat bolsa

Onda ol **quramali ta'kirarlaniwshi algoritm** delinedi.

Takrarlaniw denesindegi baslang'ish mag'liwmatlar turaqli shama, a'piwayi o'zgeriwshi, indeksli o'zgeriwshi ko'rinisinde boliwi kerek.

1. Ta'krarlaniwdi tayarlaw - ta'krarlaniwdi baslawdan aldın, ta'krarlaniwda qatnasatug'in o'zgeriwshilerdin' baslang'ish ma'nisleri yaki ta'krarlaniw o'zgeriwshisinın' baslang'ish ma'nisi ornatiladi, ta'krarlaniw o'zgeriwshisinın' o'zgeriw adimi belgilenedi.

2. Ta'krarlaniw denesi - ta'krarlaniw o'zgeriwshilerinin' tu'rli ma'nisleri ushin ta'krar orinlanatug'in a'meller izbe-izligin ko'rsetedi.

3. Ta'krarlaniw o'zgeriwshisine taza ma'nis beriw - ha'r bir ta'krarlaniwdan aldın o'zgeriwshige o'zgeriw adimina sa'ykes tu'rde taza ma'nis beriledi.

4. Ta'krarlaniwdi basqariwda ta'krarlaniwdi dawam ettiriw sha'rti tekseriledi.

Algoritmlestiriwdin' tiykarg'i tu'siniklari

Orinlawshi bul adam, mekeme, mexanikaliq yamasa elektron qurilma, robot. Bular qandayda bir a'mellerdin' ko'pligin orinlaydi. Ha'r bir orinlanatug'in a'mel bul buyriq bolip esaplanadi. Ha'r bir a'mel qandayda bir obiekt u'stinde orinlanadi ha'm sol obiektin' jag'dayin o'zgerledi. Obiektler ishki ha'm sirtqi bolip ekige bo'linedi. Orinlawshinin' ishki obiekti basil obiekt dep ataladi. Ma'selen yad qurilmasi, bul basqa obiektler menen baylanista bolip barliq buyriqlarg'a sa'ykes juwap beredi ha'm orinlawshinin' barliq ha'reketin yadta saqlaydi. Sirtqi obiektler bul ko'rsetpelerdin' parametri dep ataladi. Ol orinlaw shi menen belgili bir ko'rsetpenin' shaqirig'ina sa'ykes baylanisadi.

Algoritmderdi du'ziw ha'm analizlew usillari

Algoritmni toliq du'ziw algoritmlestiriw protsesinde en' tiykarg'i mashqala bolip esaplanadi. Bunin' tiykarg'i etapleri to'mendegiler bolip esaplanadi:

1.) Ma'selenin' qoyiliwi

- 2.) Model du'ziw
- 3.) Algoritmdi islep shig'iw
- 4.) Algoritmnin' durislig'in tekseriw
- 5.) Algoritmdi iske asiriw
- 6.) Algoritmdi ha'm onin' quramalilig'in talqilaw
- 7.) Programmani tekseriw
- 8.) Hu'jjetlestiriw

Algoritm du'zgende waqitta orinlawshinin' mu'mkinshiligi de esapqa alinadi,sebebi qa'legen adam qa'legen ma'seleni sheshiw algoritmin du'ze almawi mu'mkin.Qa'legen algoritmdi matemetikaliq formula tu'rinde jaza almawimiz mu'mkin.Bunday jag'dayda algoritmdi ta'riyplew ushin algoritmlik tillerden paydalaniwg'a boladi,al bazi bir paydalaniwshilar o'zinshe belgilerdi qabil etip, algoritmdi sol arnawli belgiler arqali jazadi.

The word *algorithm* has a long history. It derives from the name al-Khowarizmi, a ninth-century Persian mathematician and astronomer. He wrote a famous textbook on the manipulation of numbers and equations. The book was entitled *Kitab al-jabr w'almuqabala*, which can be translated as *Rules for Reuniting and Reducing*. The similar-sounding word *algebra* was derived from the Arabic word *al-jabr*, which appears in the title of the book and which is often translated as *reuniting* or *restoring*. The meanings of the words *algebra* and *algorithm* used to be much more intimately related than they are today. Indeed, until modern times, the word *algorithm* usually referred only to algebraic rules for solving numerical equations. Today, the word *algorithm* can be applied to a wide variety of kinds of instructions for manipulating symbolic as well as numeric data. The properties that qualify a set of instructions as an algorithm now are determined by the nature of the instructions rather than by the things manipulated by the instructions. To qualify as an algorithm, a set of instructions must completely and unambiguously specify the steps to be taken and the order in which they are taken. The person or machine carrying out the algorithm does exactly what the algorithm says, neither more nor less.

Algorithm

An **algorithm** is a sequence of precise instructions that leads to a solution.

Program Design

Designing a program is often a difficult task. There is no complete set of rules, no algorithm to tell you how to write programs. Program design is a creative process. Still, there is the outline of a plan to follow. The outline is given in diagrammatic form in Display 1.7. As indicated there, the entire program design process can be divided into two phases, the *problem-solving phase* and the *implementation phase*. The result of the *problem-solving phase* is an algorithm, expressed in English, for solving the problem. To produce a program in a programming language such as C++, the algorithm is translated into the programming language. Producing the final program from the algorithm is called the *implementation phase*.

3-Lekciya. Algoritmda du'ziddin' basqishlari'

Joba:

1. Ma'selenin' qoyiliwi
2. Matematikaliq model du'ziw
3. Algoritm du'ziw
4. Algoritmning durislig'i
5. Algoritmning orinlaniwi
6. Algoritmda ha'm onin' quramalilig'in analizlew
7. Maselelerdi kompyuterde sheshiw basqishlari

1. Ma'selenin' qoyiliwi:

Ba'riyin burin biz ma'selenin' duris qoyiliwi, ondag'i sorawlardin' duris ekenligine ko'zimiz jetiwi kerek. Ma'selenin' qoyiliwinda tekseriletug'in birneshe sorawlardi atap o'teyik: Ma'selenin' qoyiliwindag'i terminler tu'siniklime? Qanday na'rse kerek? Neni tabiw kerek? Sheshimdi qalay aniqlaw kerek? Qanday mag'liwmatlar jetispeydi? Artiqsha mag'liwmatlar barma? Ma'selenin' sheshimine paydasi joq mag'liwmatlar barma? Qanday artiqsha na'rselerga jol qoyilg'an?

2. Matematikaliq model du'ziw:

Ma'sele aniq qoyilg'annan keyin matematikaliq model du'ziw kerek. Modeldi tan'lap aliw qalg'an etaplarg'a tikkeley ta'sir etedi. Model du'ziw ushin tiykarinan to'mendegi sorawlarga juwap beriw kerek:

- 1) Ma'seleni sheshiw ushin tiykarinan qanday matematikaliq struktura kerek boladi?
 - 2) Qoyilg'an ma'selege uqsas ma'selelerdin' sheshimi barma? Degen sorawlar qoyiladi.
- 2-shi soraw en' tiykarg'isi esaplanadi. Sebebi qoyilg'an ma'sele burin sheshilgen ma'selelerdi tu'rlendiriw ha'm jetilistiriw arqali sheshiliwi mu'mkin.

3. Algoritm du'ziw:

Tapsirmanin' matematikaliq modelin tan'lap alg'annan keyin, og'an algoritm du'ziw kerek boladi. Algoritm birdey ma'seleler ushin ha'r qiyli bolip du'ziliwi mu'mkin. Ha'r qanday algoritm du'ziwshi adam algoritmda tez tamamlap programma du'ziwge kirisedi. Biraq bul natuwri. Sebebi

algoritm du'ziw o'zinen aldin'g'i etaplar menen tig'iz baylanisli. Ha'r bir etapni bir-birinen g'arezsiz tu'rde aniqlawg'a bolmaydi. Sonin' ushin du'zilgen algoritmdi qayta-qayta tekseriw kerek.

4. Algoritmni' durislig'i

Algoritmni' durislig'in dalillew en' qiyini ha'm sharshatatug'in etap bolip esaplanadi. Algoritmni' durislig'in programmadan araliq na'tiyjelerdi baspag'a shig'ariw joli menende tekseredi. Biraq bul protsess barliq waqitta da duris bola bermeydi.

5. Algoritmni' orinlaniwi:

Algoritmni' orinlaniwi degenimiz ol boyinsha programma du'ziw degendi an'latadi. Bul qiyin boliwi mu'mkin, sebebi, bazi bir bloklardi EEM mashinasinda jaziw qiyin boladi. Algoritmdi orinlaw ushin to'mendegi sorawlarg'a juwap izlew kerek:

- 1) Tiykarg'i o'zgeriwshiler qaysisi?
- 2) Olar qanday tu'rde boliwi mu'mkin?
- 3) Massivtin' o'lshemleri qanday?
- 4) Tiykarg'i programma menen qanday baylanista boladi?
- 5) Qanday ja'rdemshi programmalar paydalaniladi?
- 6) Qanday programmalastiriw tili kerek?

Programma da'slep joqaridan to'menge qaray izbe-izlikte tekseriledi.

8. Algoritmdi ha'm onin' quramalilig'in analizlew

Algoritmdi analizlew ushin ha'r qiyli a'meldegi misallardi keltiriwge boladi. Solardin' biri programma ko'lemin ha'm mashiyina waqtin boljaw bolip esaplanadi. Sebebi ha'r qanday quramali algoritmdi du'zgende bul EEM di paydalaniwshilar ushin en' tiykarg'i problema bolip esaplanadi. Algoritmdi analizlew bir-birine uqsas eki algoritmni' mu'mkinshiliklerin salistiriw bolip esaplanadi. Sebebi bir ma'selenin' o'zin ha'r qiyli algoritm boyinsha sheshiwge boladi.

7. Programmanin' durislig'in tekseriw:

Programmadan na'tiyje alindi, biraq oni tekseriw kerek, na'tiyje mu'mkin bolg'an da'slepki mag'liwmatlar ushin alinama yamasa joqpa? Bunin' ushin test programma du'ziw kerek. Bul programma barliq mu'mkin bolg'an variantlar ushin na'tiyje beriw kerek. Test programmani

du'ziw ,programmanin' variantlarin tekseriw jumisi en' da'slep a'piwayi programmalar di tekseriwden baslanadi.

Onnan aling'an na'tiyjenin' durislig'ina toliq isengennen keyin izbe-izlik penen ma'selenin' qoyiliwi quramaliraq formag'a o'tkeriledi ha'm en' aqirinda ha'mme sha'rtler o'z ishine alatug'in uliwma programma tekseriledi. Bunday uliwma programmalar ushin da'slepki mag'liwmatlar tan'lap aling'anda programmanin' ha'mme bloklarina ta'sir etetug'in mag'liwmatlar tan'lap alinadi. Programmani teksergende onin' sapasina da kewil bo'liw kerek. Ko'pshilik programmalar din' sapasin (orinlanatug'in operatsiyalar din' sanin ha'm alinatug'in na'tiyjeler din' shamasin) analitikaliq usilda aniqlaw mu'mkin bolmaydi. Sonin' ushin olardi programmaliq usilda aniqlag'an qolayli boladi. Programma bazi bir da'slepki mag'liwmatlar ko'pligi ushin jaqsi na'tiyje ,al bazilari ushin biz ku'tpegen na'tiyje beriw mu'mkin.

8. Hu'jjetlestiriw:

Programmani hu'jjetlestiriw ushin to'mendegi sorawlarg'a juwap beriw kerek:

- 1) Programma qanday ma'seleler di sheshiw ushin qollaniladi;
- 2) Da'slepki mag'liwmatlar ko'pligi, atamalari, formalari ha'm shegaralari;
- 3) Test programmanin' toliq ta'riypi;
- 4) Programmada qollanilatug'in tiykarg'I ha'm araliq o'zgeriwshiler olardin' o'zgeshelikleri;
- 5) Da'slepki mag'liwmatlardi jaziw formalari;
- 6) Programmada qollanilatugin bloklardin' xizmetleri;
- 7) Alinatug'in na'tiyjeler din' atamalari, baspag'a shig'ariw formalari ha'm formatlari;

Algoritm tu'sinigin formallastiriw:

Insan o'zinin' barliq turmis tarawlarinda, a'sirese mag'liwmatlardi qayta islewde de ma'seleler di sheshiw din' tu'rli usil ha'm qurallari menen soqlig'isadi. Olar son'g'i na'tiyjege erisiw ushin orinlanatug'in ha'reketler ta'rtibin aniqlaydi. Algoritm – qandayda tilde berilgen ma'seleni sheshiw ushin orinlanatug'in baslang'ish berilgenler u'stinde orinlanatug'i

a'mellerdin' shekli izbe-izligi. Ilimpazlardin' izertlewlerine qaramay, Algoritm tu'sinigine aniq bir aniqlama beriwidin' imkani bolmadi. Algoritmter teoriyasinda algoritmnin' tu'rli formal aniqlamalari kiritilgen bolip, olardin' ekvivalentligi esaplanadi. Tu'rli aniqlamalari bir qatar talaplarg'a juwap beriw kerak:

- algoritm shekli sandag'i elementar orinlawshi ko'rsetpelerden ibarat boliw kerak;

- algoritm shekli sandag'i adimlardan ibarat boliw kerak;

- algoritm barliq baslang'ish berilgenler ushin uliwma boliw kerak;

- algoritm tuwri sheshimge alip keliw kerak;

EEM jardeminde ma'seleni sheshiwdi tomendegi basqishlarga ajratiwg'a boladi:

1) Obiekt, problema, ma'sele;

2) Matematikaliq model

3) Diskret model

4) Algoritmnin' sheshiw usillari

5) Programma

6) Programmani durislaw

7) Na'tiyje ha'm onin' analizi

Maselelerdi kompyuterde sheshiw basqishlari

Kompyuterden paydalanip "ilmiy - texnik maseleni sheshiw" tusinigi ken' ma'nidegi so'z bo'lip, to'mendegi basqishlarga bo'linedi. Ma'qset – basqishlardin' qaysi birin qa'niyge kompyuterden paydalanbay ha'm qaysi basqishlardi kompyuterden paydalanip orinlawdi aniqlaw ha'm basqishlardi toliq u'yrenip shig'iwdan ibarat. Ilmiy - texnik ma'selelerdi (ITM) kompyuterden paydalanip sheshiw basqishlari:

1. Ma'selenin' qoyiliwi ha'm ma'qsettin' aniqlaniwi;

2. Ma'seleni matematik an'latiw;

3. Ma'seleni sheshiw usilin islep shig'iw, sanli usillardi tan'law;

4. Ma'seleni sheshiw algoritmin islep shig'iw;

5. Mag'liwmatlardi tayarlaw ha'm aniqlaw (tan'law);

6. Programmalaw;
7. Programma tekstin ha'm mag'liwmatlardi xabar tasiwshig'a o'tkiziw;
8. Programmanin' qa'teliklerin durislaw;
9. Programmanin' avtomatik ta'rizde kompyuterde orinlaniwi;
10. Aling'an na'tiyjelerdi tu'sindiriw ha'm programmadan paydalanuw ushin ko'rsetpe jaziw;

"Informatika" kursinda 1- 4 basqishlar qisqa ma'nide, arnawli jag'daylar, ko'p ushrasatug'in quramali bolmag'an mashqalalar ushin tu'sindiriledi. 8- ha'm 9-basqishlardi orinlawda qa'niyge kompyuterden paydalanadi. 7-basqishta kompyuterden paydalanuw ha'm, paydalanbaw da mu'mkin.

1-basqish. Ma'selenin' qoyiliwi ha'm ma'qsetti aniqlaw

Xaliq xojalig'inin' belgili bir tarawi (texnika, ekonomika, lingvistika, ta'lim) boyinsha islep atirg'an joqari da'rejeli qa'niyge ta'repinen orinlanatug'in jumis, ma'seleni qoyiw ha'm ma'qsetti aniqlaw ushin qa'niyge bir neshe ku'n, ay, jillap izleniwi mu'mkin. Qoyilg'an ma'qsetti a'melge asiriw ushin kerekli mag'liwmatlar qurami (strukturasi), du'zilisi, an'latiliwi aniqlang'an bolip, olar arasindag'i baylanislar aniq an'latilg'an bolsa, "ma'sele qoyilg'an" dep aytiladi.

2-basqish. Ma'seleni matematik an'latiw

Bul basqishta ma'seleni sheshiw ushin kerekli ha'm jeterli bolg'an da'slepki mag'liwmatlardin' qurami, tu'ri, du'zilisi esapqa aling'an jag'dayda matematikaliq terminlerde an'latiladi ha'm ma'seleni sheshiw din' matematikaliq modeli jaratiladi. Bunin' ushin ha'r tu'rli matematik apparat isletiliwi mu'mkin. Ma'selen, ekonomika tarawinda qa'niygeler siziqli da'stu'rlew, dinamik da'stu'rlew, stoxastik da'stu'rlew menen baylanisli ma'selelerdi sheshiw matematik apparatti biliw kerek; texnika tarawindag'i qa'niygeler differentsial ten'lemeler ha'm olardin' dizimleri, gaz dinamikasina tiyisli ma'selelerdi an'latiw ha'm sheshiw ushin isletiletug'in matematikaliq apparatti toliq tu'sinip jetken boliwi kerek.

Qa'niyge o'z tarawin ha'r ta'repleme jaqsi u'yrenge ha'm a'meliy jaqtan puxta o'zlestirgen ha'm qollanilatug'in ha'r tu'rli apparattin' barliq imkaniyatlarin toliq tu'sinip jetken ha'm a'meliyatta qollay alatug'in boliwi kerek.

3-basqish. Ma'sele sheshiw usilin aniqlaw, sanli usildi tan'law

Eger da'slepki mag'liwmatlar menen izlenip atirg'an na'tiyjeler ortasinda aniq bir baylanislilik (nizamliq) ornatilg'an bolip ha'm ma'seleni sheshiw usili islep shig'ilg'an bolsa yaki usi baylanisti a'melge asiriw ushin tayar sanli usil tan'lap alinip (ma'sele ushin, ma'selenin' bir bo'limi ushin) ma'selenin' sheshiw usili jaratilg'an bolsa, "ma'seleni sheshiw usili islep shig'ilg'an" delinedi.

Bunda: X – da'slepki mag'liwmatlar; Y – na'tiyje, ma'qset funktsiyasi, izlenip atirg'an mug'darlar bolsa, olar arasindag'i baylanis $Y = f(X)$ siyaqli aliniwi mu'mkin. f – da'slepki mag'liwmatlar menen na'tiyjeni baylanistiriwshi nizamliq, nizamlar jiyindisi, yag'niy X mag'liwmatlar u'stinde orinlanatug'in a'meller izbe-izligi.

Ma'seleni sheshiw din' islep shig'ilg'an usili yaki tan'lap aling'an usildin' durislig'i, na'tiyjeliligi keyingi basqishlarda tekseriledi.

4-basqish. Ma'sele sheshiw algoritmin jaratiw tiykarinan ma'seleni sheshiw algoritmi jaratiladi. Ma'seleni sheshiw algoritmi kompyuter din' imkaniyatlarin, sheshiw aniqlig'in ha'm ma'seleni kompyuterde sheshiw waqtin ha'm mug'darin esapqa alip jaratilsa ma'qsetke muwapiq boladi. Ma'selenin' algoritmin jaratiwda araliq mag'liwmatlardi ila'ji barinsha kemeytiw, sirtqi qurilmalar menen bolatug'in baylanislardi minimumg'a keltiriw kerek. Da'stu'rdin' na'tiyjeli ha'm o'nimliligi, ma'seleni sheshiw algoritmnin' puxta islengenine baylanisli, 3-4 basqishlar bir-biri menen tig'iz baylanisqan. Yag'niy, jaratilg'an usilin ha'r tu'rli usillar menen a'melge asiriw mu'mkin, usi sebepten ma'seleni sheshiw usilin ha'm algoritmnin' bir neshe variantlari boliwi mu'mkin ha'm keregi tan'lap alinadi. Quramali ma'selenin' algoritmin jaratiwda adimba-adim aniqlastiriw usilinan paydalang'an maqul, ha'r bir adimda algoritmnin' qurami a'piwayi ha'm tu'sinerli bolip qaliwg'a erisiw kerek.

Ma'seleni algoritmler protsesinde algoritmnin' bazi bir bo'leklerin, logikaliq ta'repten ayriqsha bo'leklerin an'latiwda tipik algoritmler ha'm a'meliyatta tekserilgen algoritmlerden ilaji barinsha ko'p paydalang'an maqul. Algoritmlewde modullik printsipten paydalanuw algoritmdi oqiwda ha'm programmalawda qolayliqlar jaratadi. Na'tiyjede ma'seleni sheshiw algoritmi jumisshi jag'dayg'a keltiriledi, yag'niy algoritm grafik ko'riniste bazi bir algoritmik til quralinda an'latiw da'rejesine keltiriw mu'mkin. Ma'seleni algoritmler – ma'seleni kompyuterden paydalanip sheshiw algoritmin jaratiw protsesi.

Algoritmlew – ma'seleni sheshiw basqishi bolip, ma'selege qoyilg'an sha'rt ha'm talaplar tiykarinda axirg'i na'tiyjeni, ma'selenin' sheshimin aliw ushin islep shig'ilg'an algoritmlerdi jaratiw menen shug'illanatug'in informatikanin' bo'limi.

5-basqish. Mag'liwmatlardi tayarlaw ha'm quramin aniqlaw

Mag'liwmatlardi suwretlew usulin tan'law algoritmnin' orinlaniwi menen tig'iz baylanisqan. Sol sebepten mag'liwmatni suwretlew din' sonday tu'rin, usulin tan'law kerek, ma'seleni sheshiw protsesi a'piwayi ha'm tu'sinikli bolsin.

Mag'liwmatlar a'piwayi o'zgeriwshiler ko'rinisinde, massiv ko'rinisindide, bo'lek mag'liwmat fayllari (izbe-iz oqilatug'in) ko'rinisinde xabar tasiwshida jaylasqan boliwi mu'mkin.

6-basqish. Da'stu'rlew

Ma'seleni jumisshi jag'dayg'a keltirilgen sheshiw algoritmin tan'lang'an algoritmik til quralinda an'latiw "da'stu'rlew" delinedi. Algoritmnin' ha'r bir mayda bo'legi algoritmik tildin' operatorlari ja'rdeminda, tildin' sintaksis ha'm semantika qag'iydalari tiykarinda jazip shig'iladi. Algoritm quramali du'zilgan bolsa da'stu'rlewde qiyinshiliq tuwilmaydi. Da'stu'rlew protsesinde to'mendegi usinlar itibarg'a alinsa qa'telerdi durislaw protsesi jen'illesedi.

1. Da'stu'r uliwmalig boliwi kerek, yag'niy mag'liwmatlar aniq bazi bir tu'rine baylanisli bolmawi kerek, massivtin' shegara parametrlerin tekseriw kerek. Massiv elementlerinin' sani 0 yaki 1 bolip qaliw, yaki joqari shegarasinan asip ketiw jag'dayi.

2. Turaqli shama ha'm o'zgeriwshi shama ko'rinisinde islewi mu'mkin.

3. Da'stu'rdegi arifmetikaliq a'mellerdi kemeytiw ha'm da'stu'rdin' islewin tezletiw ushin:

- da'rejege ko'teriw a'melin ko'beytiw a'meli menen almastirg'an maqul.

- Birdey mag'liwmatlar menen esaplanatug'in arifmetik a'mellerdi bir ma'rte esaplap mug'darin bazi bir o'zgeriwshide saqlap isletiw;

- Qaytalaniwlardi payda etiwde qaytalaniwdin' shegarasin tekseriw ushin an'latpalardan emes a'piwayi o'zgeriwshilerden paydalaniv;

Da'stu'rdin' ha'r bir bo'legi, moduli bo'legine tu'sindirmeler jazilg'an boliwi kerek. Da'stu'rdegi tu'sindirmeler, ma'seleni sheshiw izbe-izligin an'latiwshi logikaliq izbe-izlikten ibarat boliwi kerek. Da'stu'rdegi moduller, bo'limler, aniq ko'rsetilgen boliwi kerek. Ta'krarlaniw basi ha'm ta'krarlaniw axiri bo'lek qatarda turg'ani maqul.

7-basqish. Da'stu'r tekstin ha'm mag'liwmatlardi xabar tasiwshig'a jetkiziw

Kompyuter ushin xabar tasiwshi qurallar bolip: perfokarta, perfolenta, magnitli tasma, magnitli disk (iyiliwshen' magnitli disk, magnitli karta), fleshkalar xizmet qiliwi mu'mkin. Da'stu'r teksti bo'lek arnawli qurilmalar ja'rdeminde yaki kompyuterden paydalanip xabar tasiwshig'a o'tkiziledi.

8-basqish. Da'stu'rdin' qa'tesin aniqlaw

Bul basqish ma'seleni kompyuterde sheshiw basqishlari ishindegini ko'p waqit talap qilaturg'in, qa'niygeden sabir, qanaat, shidam, aqil, logikaliq tez pikirlew, imkaniyatlarin, durislaw usilin, jollarin, ma'selenin' ma'nisin teren' biliwdi talap qilaturg'in quramali protsess. Bul basqish "da'stu'rdi test boyinsha tekseriw" dep te ju'rgiziledi. Da'stu'rdin' tuwri islewi ham jol qoyilg'an qateliklerdi aniqlap du'zetiw algoritmdin jaratiwda jol qoyilg'an kemshiliklerdi joq qiliw ham'de tan'lang'an usildin' jaramli yaki jaramsiz ekenligin aniqlap beriwshi protsess. Test – arnawli tayarlang'an da'slepki mag'liwmatlar bolip, olar u'stinde a'meller orinlaw menen masele sheshim-na'tiyje alinadi. Test tayarlaw ju'da quramali jumis bolip, qolda

esap-kitap jumislari orinlawdi talap etiledi ha'm ha'mme jumislardi tekseriw sha'rt. Da'stu'rdin' qa'tesin du'zetiwi boyinsha jol-jobalar:

1. Arnawli tayarlang'an mag'liwmatlar tiykarinda da'stu'rdi qolda esaplap shig'iw (imkani bolsa) yaki logikaliq ayriqsha bolg'an bo'lekler modullerini qolda esaplaw.

2. Da'stu'rdi ha'm onin' bo'leklerin, moduller test ja'rdeminde tekseriw.

3. Da'stu'rdin' kerekli orinlarini basip shig'ariw buyrig'in qoyiw. (Duzetiwlerden keyin alip taslanadi).

4. Da'stu'rdin' qa'telerdi du'zetiwde, so'ylesiw rejimi orinlang'anda (stop) .Toqtatiw buyrig'inan paydalaniw.

5. Da'stu'rlew tilin ha'm a'meldi orinlaw diziminin' arnawli qa'telerin du'zetiwi imkaniyatlarini paydalaniw.

6. Qa'telerdi du'zetiwi protsesinde kem ko'lemdegi mag'liwmatlar menen islewdi payda etiw.

9-basqish. Da'stu'rdin' avtomatik ta'rizde kompyuterde orinlaniwi

Kompyuter qa'teleri du'zetilip tayarlang'an da'slepki mag'liwmatlardan paydalang'an tu'rde maselenin' (sheshimlerin) avtomatik tarzide esaplaydi. Eger natijeler ma'selenin' sheshimi ushin jaramli dep tabilsa ma'seleni sheshiw tamamlang'an esaplanadi, bolmasa joqaridag'i basqishlar qaytadan korip shig'iladi.

10-basqish. Aling'an mag'liwmatlardi tastiyoqlaw, boljaw ha'm da'stu'rden paydalaniw ushin jollanba jaziw: Ma'seleni sheshiw na'tijesinde aling'an sanlar massivi, temalar yaki tema ko'rinishidegi massivlar ha'r tarepleme tastiyoqlanip, tusindiriledi. Da'sturden paydalaniw ushin ko'rgezbe jaziw to'mendegilerdi o'z ishine aladi:

- Dastu'r islew ushin mag'liwmatlardi tayarlaw, du'zilisi aniq belgilengen;

- Dastu'rdin' islewi ushin kompyuterdi so'ylew jollari;

- Dastu'rdi iske tusiriw ham islew waqtinda bolatug'in soraw juwaplar;

- Dastu'rdi islew protsesinde kelip shig'atug'in har tu'rli jag'daylardi joq etiw jollari aniq ham puxta tusinikli bolip jazilg'an bolivi kerek.

The first step is to be certain that the task—what you want your program to do—is completely and precisely specified. Do not take this step lightly. If you do not know exactly what you want as the output of your program, you may be surprised at what your program produces. Be certain that you know what the input to the program will be and exactly what information is supposed to be in the output, as well as what form that information should be in. For example, if the program is a bank accounting program, you must know not only the interest rate but also whether interest is to be compounded annually, monthly, daily, or whatever. If the program is supposed to write poetry, you need to determine whether the poems can be in free verse or must be in iambic pentameter or some other meter.

Algorithms

When learning your first programming language, it is easy to get the impression that the hard part of solving a problem on a computer is translating your ideas into the specific language that will be fed into the computer. This definitely is not the case. The most difficult part of solving a problem on a computer is discovering the method of solution. After you come up with a method of solution, it is routine to translate your method into the required language, be it C++ or some other programming language. It is therefore helpful to temporarily ignore the programming language and to concentrate instead on formulating the steps of the solution and writing them down in plain English, as if the instructions were to be given to a human being rather than a computer. A sequence of instructions expressed in this way is frequently referred to as an *algorithm*.

A sequence of precise instructions which leads to a solution is called an **algorithm**. Some approximately equivalent words are *recipe*, *method*,

4-Leksiya. Programmalastiriw tilleri. C\C++ tilleri
Joba:

1.

C++ is the single most important language that any programmer can learn. This is a strong statement, but it is not an exaggeration. C++ is the center of gravity around which all of modern programming revolves. Its syntax and design philosophy define the essence of object-oriented programming. Moreover, C++ charts the course for future language development. For example, both Java and C# are directly descended from C++. C++ is also the universal language of programming; it is the language in which programmers share ideas with one another. To be a professional programmer today implies competency in C++. It is that fundamental and that important. C++ is the gateway to all of modern programming.

Before beginning your study of C++, it is important for you to know how C++ fits into the historical context of computer languages. Understanding the forces that drove its creation, the design philosophy it represents, and the legacy that it inherits makes it easier to appreciate the many innovative and unique features of C++. With this in mind, this chapter presents a brief history of the C++ programming language, its origins, its relationship to its predecessor (C), its uses, and the programming philosophies that it supports. It also puts C++ into perspective relative to other programming languages.

The Origins of C++

The story of C++ begins with C. The reason for this is simple: C++ is built upon the foundation of C. In fact, C++ is a superset of C. (Indeed, all C++ compilers can also be used to compile C programs!) Specifically, C++ is an expanded and enhanced version of C that embodies the philosophy of object-oriented programming (which is described later in this chapter). C++ also includes several other improvements to the C language, including an extended set of library routines. However, much of the spirit and flavor of C++ is inherited directly from C. To fully understand and appreciate C++, you need to understand the “how and why” behind C.

The Creation of C

The C language shook the computer world. Its impact should not be underestimated because it fundamentally changed the way programming was approached and thought about. C is considered by many to be the first modern “programmer’s language.” Prior to the invention of C, computer languages were generally designed either as academic exercises or by bureaucratic committees. C is different. C was designed, implemented, and developed by real, working programmers, and it reflected the way they approached the job of programming. Its features were honed, tested, thought about, and rethought by the people who actually used the language. The result of this process was a language that programmers liked to use. Indeed, C quickly attracted many followers who had a near-religious zeal for it, and it found wide and rapid acceptance in the programmer community. In short, C is a language designed by and for programmers.

C++ Is Born

In response to the need to manage greater complexity, C++ was born. It was invented by Bjarne Stroustrup in 1979 at Bell Laboratories in Murray Hill, New Jersey. He initially called the new language "C with Classes." However, in 1983 the name was changed to C++.

C++ contains the entire C language. As stated earlier, C is the foundation upon which C++ is built. C++ includes all of C's features, attributes, and benefits. It also adheres to C's philosophy that the programmer, not the language, is in charge. At this point, it is critical to understand that the invention of C++ was not an attempt to create a new programming language. Instead, it was an enhancement to an already highly successful language.

Most of the additions that Stroustrup made to C were designed to support object-oriented programming. In essence, C++ is the object-oriented version of C. By building upon the foundation of C, Stroustrup provided a smooth migration path to OOP. Instead of having to learn an entirely new language, a C programmer needed to learn only a few new features to reap the benefits of the object-oriented methodology.

But C is not the only language that influenced C++. Stroustrup states that some of its object-oriented features were inspired by another object-oriented language called Simula67. Therefore, C++ represents the blending of two powerful programming methods.

When creating C++, Stroustrup knew that it was important to maintain the original spirit of C, including its efficiency, flexibility, and philosophy, while at the same time adding support for object-oriented programming. Happily, his goal was accomplished. C++ still provides the programmer with the freedom and control of C, coupled with the power of objects.

Although C++ was initially designed to aid in the management of very large programs, it is in no way limited to this use. In fact, the object-oriented attributes of C++ can be effectively applied to virtually any programming task. It is not uncommon to see C++ used for projects such as compilers, editors, programmer tools, games, and networking programs. Because C++ shares C's efficiency, much high-performance systems software is constructed using C++. Also, C++ is frequently the language of choice for Windows programming.

One important point to remember is this: Because C++ is a superset of C, once you can program in C++, you can also program in C! Thus, you will actually be learning two programming languages at the same time, with the same effort that you would use to learn only one.

The Evolution of C++

Since C++ was first invented, it has undergone three major revisions, with each revision adding to and altering the language. The first revision was in 1985 and the second occurred in 1990. The third revision occurred during the C++ standardization process. In the early 1990s, work began on a standard for C++. Towards that end, a joint ANSI and ISO (International Standards Organization) standardization committee was formed. The first draft of the proposed standard was created on January 25, 1994. In that draft, the ANSI/ISO C++ committee (of which I was a member) kept the features first defined by Stroustrup and added some new ones as well. But, in general, this initial draft reflected the state of C++ at the time.

Soon after the completion of the first draft of the C++ standard, an event occurred that caused the standard to expand greatly: the creation of the Standard Template Library (STL) by Alexander Stepanov. As you will learn, the STL is a set of generic routines that you can use to manipulate data. It is both powerful and elegant. But the STL is also quite large. Subsequent to the first draft, the committee voted to include the STL in the specification for C++. The addition of the STL expanded the scope of C++ well beyond its original definition. While important, the inclusion of the STL, among other things, slowed the standardization of C++.

It is fair to say that the standardization of C++ took far longer than any one had expected when it began. In the process, many new features were added to the language and many small changes were made. In fact, the version of C++ defined by the C++ committee is much larger and more complex than Stroustrup's original design. The final draft was passed out of committee on November 14, 1997, and an ANSI/ISO standard for C++ became a reality in 1998. This specification for C++ is commonly referred to as *Standard C++*.

The material in this book describes Standard C++. This is the version of C++ supported by all mainstream C++ compilers, including Microsoft's Visual C++ and Borland's C++ Builder. Therefore, the code and information in this book is fully applicable to all modern C++ environments.

5-Lekciya. C++ tili ha'm woni'n' leksik tiykarlari

Joba:

1. C++ tilidagi programma du'zilisi ha'm woni'n' kompilyatsiyasi.
2. C++ tili alfaviti ha'm leksemalar.
3. Identifikatorlar ha'm gilt so'zler.

Tayansh so'zler: Kompilyatsiya, fayl ken'eytpesi, direktevalar, leksemalar, tu'sinikler, ag'i'mli' woqi'w ha'm jazi'w funksiyalari', til alfaviti, identifikatorlar, gilt so'zler.

C++ tilidagi programma du'zilisi ha'm woni'n' kompilyatsiyasi:
C++ tilinde programma jarati'w bir neshe basqi'shlardan ibarat boladi'. Da'slep, tekst redakciyasi'nda(a'detde programmalasti'ri'w wortali'g'i'ni'n' redakciyasi'nda) programma teksti teriladi, bul fayldi'n' ken'eytpesi «.cpp» boladi'. Keyingi basqi'shda programma teksti jazi'lg'an fayl kompilyatorga uzati'ladi', yegerde programmada aljasi'qlar bo'lmasa, kompilyator «.obj» ken'eytpeli obyekt modul fayli'n payda yetadi'. Son'g'i' qa'demde komponovka(yig'uvchi) ja'rdeminde «.exe» ken'eytpeli atqari'li'wshi' fayl programma payda boladi'. Basqi'shlarda ju'zege keliwshi fayllardi'n' atlari' baslang'i'sh tekst fayli'ni'n' ati' menen birdey boladi'.

Kompilyatsiya procesinin' wo'zi da yeki basqi'shdan tashkil tabadi'. Basi'nda preprotessor isleydi, wol tekstdegi kompilyatsiya direktivalarini atqaradi', atap aytqanda #include direktivasi boyi'nsha ko'rsetilgen kitapxanalardan C++ tilinde jazi'lg'an modullardi programma qurami'na kiritedi. Sonnan keyin kengaytirilgan programma teksti kompilyatorga uzati'ladi'.

Kompilyator wo'zi da programma boli'p, woni'n' ushi'n kiretug'i'n mag'li'wmat boli'p, C++ tilinde jazi'lg'an programma teksti yesaplanadi'. Kompilyator programma tekstin leksema(atomar) elementlerge ajratadi' ha'm woni' leksik, keyinirek sintaktik analiz yetedi. Leksik analiz processinde wol teksti leksemalarga aji'rati'w ushi'n «probel aji'rati'wchisini» isletedi. Probel aji'rati'wshi'si' - probel belgisi(' '), '\t'-

tabulyatsiya belgisi, '\n'-keyingi qatarg'a o'ti'w belgisi, basqa aji'rati'wshi'lar ha'm tu'sinikler(kommentariylar) kiredi.

Programma teksti tu'sinikli boli'wi' ushi'n tu'sinikler isletiledi. Tu'sinikler kompilyator ta'repinen «o'tkerilip» jiberiledi ha'm wolar programma a'mel yetiwine hesh qanday ta'sir yetpeydi.

C++ tilinde tu'sinikler yeki ko'riniste jazi'li'wi' mu'mkin.

Birinshisinde /* den baslani'p, */ belgalari menen tawsi'lg'an barli'q belgiler izbe-izligi tu'sindirme yesaplanadi', yekinshisi «qatarli' tu'sindirme» dep ataladi' ha'm wol // belgilerden baslang'an ha'm qatar aqi'ri'i'na shekem jazi'lg'an belgiler izbe-izligi boladi'. Tu'sindirmenin' birinshi ko'riniside jazi'lg'an tu'sinikler bir neshe qatar boli'wi' ha'm wolardan keyin C++ operatorlari' dawam yetiwi mu'mkin.

Mi'sal.

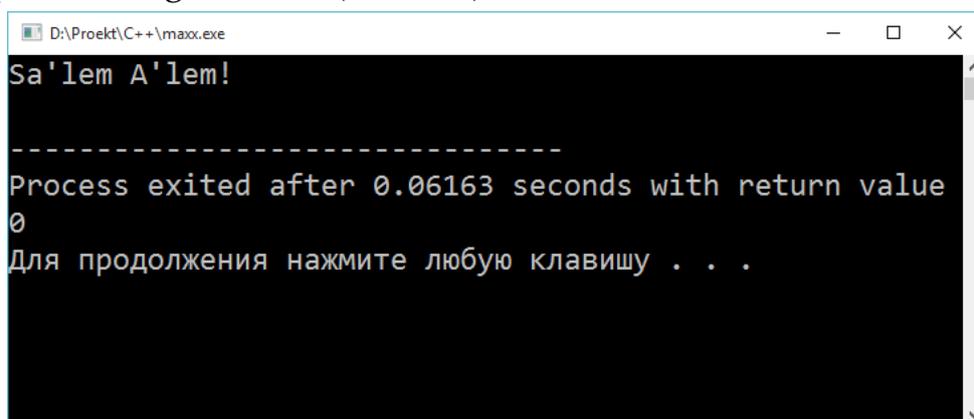
```
int main()
{
    // bul qatar tu'sindirme yesaplanadi'
    int a=0; //int d;
    int c;
    /* int b=15 */
    /*-tu'sindirme baslani'wi'
    a=c;
    tu'sindirme tugashi */
    return 0;
}
```

Programmada d, b wo'zgeriwshiler dag'azalari' inabatqa ali'nbaydi' ha'm a=c a'meli atqari'lmaydi'.

To'mende C++ tilindegi a'piwayi' programma teksti keltirilgen.

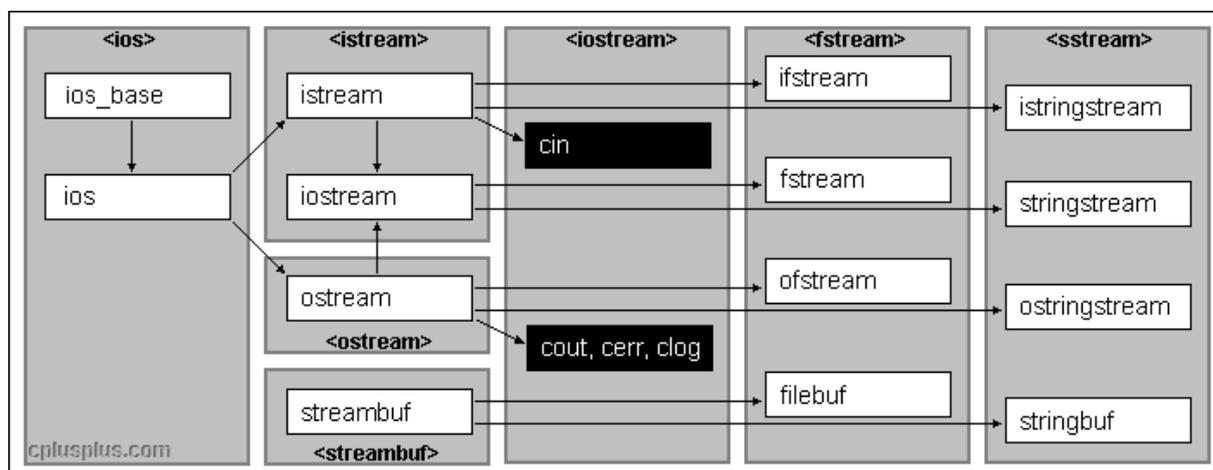
```
#include <iostream> // bas bet fayldi' qosi'w
using namespace std; //standart atlar ken'isligi
int main() // bas funksiya suwretlewi
{ // blok baslani'wi'
    cout<<"Sa'lem A'lem!\n"; //baspadan shi'g'ari'w
    return 0; // funksiya qaytaratug'i'n ma'nis
}
```

Programma atqari'li'wi' na'tiyjesinde konsol ekrang'a "Sa'lem A'lem!" qatari' baspadan shi'g'ari'ladi' (su'ret-1).



Su'ret-1. Programmanin' natiyje aynasi'

Programmanin' 1-qatari'nda #include ... preprotessor direktivasi boli'p, programma kodi'na ag'i'mli' woqi'w/jazi'w funksiyalari' ha'm woni'n' wo'zgeriwshileri dag'azasi' jaylasqan «iostream.h» bas bet fayli'n qosadi'. Keyingi qatarlarda programmanin' birden-bir, tiykarg'i' funksiyasi' - main() funksiyasi' suwretlewi keltirilgen. Soni' atap ko'rsetiw za'ru'r, C++ programmasida a'l'bette main() funksiyasi' boli'wi' sha'rt ha'm programma sol funksiyani' wori'nlaw menen wo'z jumi'si'n baslaydi'.



Su'ret-1. Kiritiw/shig'ari'w kutipxanasi ierarxiyasi. <http://www.cplusplus.com/> saytinan aling'an

Programma denesinde konsol rejiminde belgiler izbe-izligini ag'i'mg'a shi'g'ari'w a'meli qollani'lg'an. Mag'li'wmatlardi' standart ag'i'mg'a (ekrang'a) shi'g'ari'w ushi'n to'mendegi format isletilingen:

```
cout<<an'latpa;
```

Bul jerde “an’latpa” si’pati’nda wo’zgeriwshi yamasa sintaksisi tuwri’ jazi’lg’an ha’m qanday da ma’nis qabi’l yetiwshi til an’latpasi’ keliwi mu’mkin (keyinirek, mu’yesh qawi’s ishine ali’ng’an qaraqalpaq tilinde qatar asti’n til qurami’na kirmeytugi’n tu’sinik dep qabi’llaw kerek).

Mi’sal:

```
int wozg=324;
cout<<wozg; // pu’tin san baspadan shi’g’ari’ladi’
```

Berilgenlerdi standart ag’i’mdan(a’detde klaviaturadan) woqi’w to’mendegi formatda a’melge asi’ri’ladi’:

```
cin>> <o’zgaruvchi>;
```

Bul jerde <o’zgaruvchi> ma’nis qabi’l yetiwshi wo’zgeriwshinin’ ati’.

Mi’sal:

```
int Jas;
cout <<”Jasi’ni’zdi kiritin’ ”;
cin>>Jas;
```

Pu’tin tu’rdegi Jas wo’zgeriwshisi kiritilgen ma’nisti wo’zlestiredi. Kiritilgen ma’nisti wo’zgeriwshi tu’rine sa’ykes keliwin tekseriw juwapkerligi programma du’ziwshisinin’ waziypasi’na juklenedi.

Bir waqtti’n’ wo’zinde probel (‘ ’) jardeminde bir neshe ha’m har-xil ma’nislerdi’ ag’i’mdan kirgiziw mu’mkin. Ma’nis kirgiziw <Enter> tuymesin basi’w menen tawsi’ladi’. Yeger kiritilgen ma’nisler sani’ wo’zgeriwshiler sani’nan ko’p bolsa, «arti’qsha» ma’nisler bufer yadta saqlani’p qaladi’.

```
#include <iostream.h>
int main()
{
    int x, y;
    float z;
    cin>>x>>y>>z;
    cout <<“Kiritilgen ma’nisler \n”;
    cout<<x<<'\t'<<y<<'\t'<<z;
    return 0;
}
```

Wo’zgeriwshilerge ma’nis kirgiziw ushi’n klaviatura arqali’

10 20 3.14 <enter>

ha'reketi a'melge asi'ri'ladi'. Soni' atap ko'rsetiw za'ru'r, ag'i'mg'a ma'nis kirgiziwde probel aji'rati'wshi' yesaplanadi'. Haqi'yqi'y sanni'n' pu'tin ha'm bo'lshek bo'limleri ' . ' belgisi menen ajrati'ladi'

C++ tili alfaviti ha'm leksemalar

C++ tili alfaviti ha'm leksemalarina to'mendegiler kiredi:

- u'lken ha'm kishi lotin alfaviti ha'ripleri;
- nomerler-0, 1, 2, 3, 4, 5, 6, 7, 8, 9;
- arnawli' belgiler: " {} | []() +/- % \ ; ' : ? <=> _ ! & ~ # ^ . *

Alfavit belgilerinen tildin' leksemalari qa'liplestiredi:

- identifikatorlar;
- gilt so'zler (xizmetshi yamasa rezervlengen);
- turaqli'lar;
- a'meller bellenishlari;
- aji'rati'wshi'lar.

Identifikatorlar ha'm gilt so'zler: Programmalasti'ri'w tilinin' za'ru'rli tayansh tu'siniklerinen biri identifikator tu'sinigi boli'p tabi'ladi'.

Identifikator-u'lken ha'm kishi lotin ha'ripleri, nomerler ha'm tag si'zi'q('_') belgilerinen sho'lkemlesken ha'm nomerden baslanmaytugi'n belgiler izbe-izligi tu'siniledi. Identifikatorlarda ha'riplerdin' registrlari (u'lken yamasa kishiligi) yesapqa ali'nadi'. Mi'sali', RUN, run, Run - bul har-xil identifikatorlar boli'p tabi'ladi'. Identifikator uzi'nli'g'i'na shegara qoyi'lmag'an, lekin wolar kompilyator ta'repinen tek basi'ndag'i' 32 belgisi menen pari'q yetedi.

Identifikatorlar gilt so'zler, wo'zgeriwshiler, funksiyalar, ni'shanlar ha'm basqa obyektleri atawda isletiledi.

C++ tilinin' gilt so'zlerine to'mendegiler kiredi:

asm, auto, break, case, catch, char, class, const, continue, default, delete, do, double, else, enum, explicit, extern, float, for, friend, goto, if, inline, int, long, mutable, new, operator, private, protected, public, register, return, short, signed, sizeof, static, struct,

swith, template, this, throw, try, typedef, typename, union, unsigned, virtual, void, volatile, while.

Juqari'da keltirilgen identifikatorlardi basqa maqsette isletiw mu'mkin yemes.

Protsessor registrlarini belgilew ushi'n to'mendegi so'zler isletiledi:

`_AH, _AL, _AX, _EAX, _BN, _BL, _BX, _EVX, _CL, _CN, _CX, _ESX, _DN, _DL, _DX, _EDX, _CS, _ESR, EBP, _FS, _GS, _DI, _EDI, _SI, _ESI, _BP, SP, DS, _ES, SS, _FLAGS.`

Bulardan ti'sqari' «_» (yeki tag si'zi'q) belgilerinen baslang'an identifikatorlar kitapxanalar ushi'n rezervlengen. Usi'ni'n' sebebinen '_' ha'm «_» belgilerdi identifikatordin' birinshi belgisi si'pati'nda isletmegen maqul. Identifikator belgiler arasi'nda probel isletiw mu'mkin yemes, za'ru'r bolg'anda woni'n' worni'na '_' isletiw mu'mkin: `cilindr_radiusi`, `shen'ber_diametri`.

Soraw ha'm tapsi'rmalar

1. *Programmanin' birden-bir tiykarg'i' funksiyasi'.*
2. *Mag'li'wmatlardi' standart ag'i'mlari'n shi'g'ari'w ha'm woqi'w.*
3. *Idintifikatorlar.*
4. *Programmatalasti'ri'w tilleri ha'm wolardi'n' kategoriyalari'.*
5. *Programma atqari'li'w processinde wo'zgeriwshi ha'm turaqli'lardan paydalani'w.*
6. *C++ tilinde tu'sinikler.*
7. *C++ tilinde mag'li'wmatlardi' woqi'w.*
8. *C++ tilinde mag'li'wmatlardi' jazi'w.*
9. *C++ tili alfaviti.*
10. *C++ tilinin' gilt so'zleri.*

One of the hardest things about learning a programming language is the fact that no element exists in isolation. Rather, the components of the language work together. It is this interrelatedness that makes it difficult to discuss one aspect of C++ without involving another. To help overcome this problem, this chapter provides a brief overview of several core C++ features, including the general form of a C++ program, some simple control statements, variables, and operators. It does not go into too many details, but rather concentrates on the general concepts common to all C++ programs. Most of the topics presented here are examined more closely in later chapters.

Since learning is best accomplished by doing, it is recommended that you work through the examples using your computer.

Your First C++ Program

Before getting into any theory, let's look at a simple C++ program. We will start by entering, compiling, and running the following program.

```
/* Program #1 - A first C++ program.

   Enter this program, then compile and run it.
*/

#include <iostream>
using namespace std;

// main() is where program execution begins.
int main()
{
    cout << "This is my first C++ program.";

    return 0;
}
```

You will follow these steps.

1. Enter the program.
2. Compile the program.
3. Execute the program.

6-Lekciya: C++ tilinde berilgenler ha'm wolardi'n' tu'rleri

Joba:

1. Turaqli'lar
2. Berilgenler tu'rleri ha'm wo'zgeriwshiler
3. C++ tilinin' tayansh tu'rleri
4. Tu'rlengen turaqli'lar
5. Sanap o'tiliwshi tu'r
6. Tu'rdi basqa tu'rge keltiriw

Tayansh so'zler: Turaqli'lar, wo'zgeriwshiler, tayansh tu'rler, tu'rlengen turaqli'lar, sanap o'tiliwshi tu'rler, tu'rdi basqa tu'rge keltiriw.

Turaqli'lar: Turaqli' (literal) - bul fiksirlengen sandi', qatardi' ha'm belgini an'latiwshi' leksema boli'p tabi'ladi'.

Turaqli'lar bes gruppag'a bo'linedi - pu'tin, haqi'yqi'y(ju'ziwshi noqatli'), sanap o'tiliwshi, belgi (literli) ha'm qatar(string, literli qatar).

Kompilyator turaqli'di' leksema si'pati'nda ani'qlaydi', wog'an yaddan jay ajratadi', ko'rinisi ha'm ma'nisine(tu'rine) qaray saykes gruppalar'ga boladi'.

Pu'tin turaqli'lar. Pu'tin turaqli'lar to'mendegi formatlarda boladi':

- ✓ wonli'q san;
- ✓ segizlik san;
- ✓ won alti'li'q san.

Wonli'q turaqli' 0 nomerinen ayri'qsha nomerden baslani'wshi' nomerler izbe-izligi ha'm 0 yesaplanadi': 0; 123; 7987; 11.

Keri turaqli' - bul belgisiz turaqli' boli'p, wog'an tek belgini wo'zgertiw a'meli qollani'lg'an dep yesaplanadi'.

Segizlik turaqli' 0 nomerinen baslani'wshi' segizlik sanaq sistemasi'(0, 1,..., 7) nomerlerinen sho'lkemlesken nomerler izbe-izligi:

023; 0777; 0.

Won alti'li'q turaqli' Ox yamasa OX belgilerinen baslanatug'i'n won alti'li'q sanaq sistemasi' nomerlerinen ibarat izbe-izlik yesaplanadi':

0x1A; 0X9F2D; 0x23.

Ha'rip belgiler qa'legen registrlarda beriliwi mu'mkin.

Kompilyator sandi'n' ma'nisine qaray wog'an saykes tu'rdi belgileydi. Yeger tilde belgilengen tu'rler programma du'ziwshin qa'naatlandirmasa, wol oshkor tu'rde tu'rdi ko'rsetiwi mu'mkin. Woni'n' ushi'n pu'tin turaqli' nomerleri aqi'ri'na, probelsiz l yamasa L(long) jazi'ladi'. Za'ru'r jag'daylarda bir turaqli' ushi'n bul belgilerdin' yekewin da isletiw mu'mkin:

451u, 012U1, 0xA2L.

Haqi'yqi'y turaqli'lar. Haqi'yqi'y turaqli'lar -ju'ziwshi noqatli' san boli'p, wol yeki xil formatda beriliwi mu'mkin:

-wonli'q fiksirlengen noqatli' formatda. Bul ko'riniste san noqat arqali' aji'rati'lg'an pu'tin ha'm bo'lshek bo'limler ko'rinisinde boladi'. Sanni'n' pu'tin yamasa bo'lshek bo'legi bolmaytug'i'ni' mu'mkin, lekin noqat a'l'bette boli'wi' kerek. Fiksirlengen noqatli' turaqli'larg'a mi'sallar:

24.56; 13.0; 66.; .87;

-eksponensial formada haqi'yqi'y turaqli' 6 bo'lekden ibarat boladi':

- 1) pu'tin bo'legi(o'nli pu'tin san) ;
- 2) o'nli bo'lshek noqat belgisi;
- 3) bo'lshek bo'legi(wonli'q belgisiz turaqli') ;
- 4) eksponenta belgisi 'e' yamasa 'E';
- 5) won da'rejesi ko'rsetkishi(o'nli pu'tin san) ;
- 6) qosi'msha belgisi('F' yamasa f, 'L' yamasa 'l').

Eksponensial formadag'i' turaqli' sanlarg'a mi'sallar:

1e2; 5e+3; .25e4; 31.4e-1.

Belgi turaqli'lar. Belgi turaqli'lar ti'rnaqsha(', '-apostroflar) ishine ali'ng'an bo'lek belgilerden tashkil tabadi' ha'm wol char gilt so'zi menen ani'qlanadi'. Belgi turaqli' ushi'n yadta bir bayt jay ajrati'ladi' ha'm wol jag'dayda pu'tin san ko'rinisindegi belginin' ASCII kodi' jaylasadi'. To'mendegiler belgi turaqli'larg'a mi'sal boladi':

'e', '@', '7', 'z', 'w', '+', 'sh', '*', 'a', 's'

Escape belgileri	Ishki kodi (16 san)	Ati'	Belginin' nomlanishi ha'm wog'an saykes a'mel
\\	0x5C	\	Teris yon si'zi'qdi' baspadan shi'g'ari'w

\'	0x27	'	Apostrofni' baspadan shi'g'ari'w
\"	0x22	"	Ti'rnaqshani' baspadan shi'g'ari'w
\?	0x3 F	?	Soraw belgisi
\a	0x07	be1	Dawi's signali'n beriw
\b	0x08	bs	Kursordi' 1 belgi worni'na keyin basi'p qaytari'w
\f	0x0C	ff	Betti wo'tkeriw
\n	0x0A	lf	Qatardi' wo'tkeriw
\r	0x0D	cr	Kursordi' a'yne qatardi'n' basi'na qaytari'w
\t	0x09	ht	Kursordi na'wbettegi tabulyatsiya jayi'na wo'tkeriw
\v	0x0D	vt	Vertikal tabulyatsiya(to'menge)
\000	000		Segizlik kodi'
\xNN	0xNN		Belgi won alti'li'q kodi' menen berilgen

Ayi'ri'm belgi turaqli'lar '\ ' belgisinen baslanadi', bul belgi birinshiden, grafikali'q ko'rinske iye bolmag'an turaqli'lardi' belgileydi, yekinshiden, arnawli' wazi'ypalar ju'klengen belgiler-apostrof belgisi, soraw belgisin(?), teris yon si'zi'q belgisin(\) ha'm yeki ko'stirnaq belgisin("") baspadan shi'g'ari'w ushi'n isletiledi. Wodan ti'sqari', bul belgi arqali' belgi ko'rinsin yemes, ba'lki oshkor tu'rde woni'n' ASCII kodi' segizlik yamasa won alti'li'q formada jazi'w mu'mkin. Bunday belgiden baslang'an belgiler escape izbe-izlikler dep ataladi'(1.1-keste).

C++ tilinde qosi'msha tu'rde wide harfli turaqli'lar ha'm ko'p belgili turaqli'lar ani'qlang'an.

Wide harfli turaqli'lar tu'ri milliy kodlardi' belgilew ushi'n kiritilgen boli'p, wol shar _t gilt so'zi menen beriledi, ha'mde yadta 2 bayt jay iyeleydi. Bul turaqli' L belgisinen baslanadi':

L '\013\022', L/cc'

Ko'p belgili turaqli' tu'ri int boli'p, wol to'rtew belgiden ibarat boli'wi' mu'mkin:

'abs', '\001\002\003\004'

Qatar turaqli'lar. Yeki ti'rnaqsha(", ") ishine ali'ng'an belgiler izbe-izligi qatar turaqli' dep ataladi':

"bul qatar turaqli' ha'm woni'n' ati' string\n"

Qatar ishinde escape izbe-izligi da isletiliwi mu'mkin, tek bul izbe-izlik apostrofsi'z jazi'ladi'.

Probel menen aji'rati'p jazi'lg'an qatarlar kompilyator ta'repinen birden-bir qatarg'a jalg'anadi' :

```
“qatar-bul belgiler massivi” /* bul qatar keyingi qatarg'a qosi'ladi' */ “, woni'n' tu'ri char[]”;
```

Bul jazi'wg'a

```
“qatar-bul belgiler massivi, woni'n' tu'ri char[]”;
```

jazi'wi' menen ekvivalent yesaplanadi'.

Uzi'n qatardi' bir neshe qatarg'a jazi'w mu'mkin ha'm woni'n' ushi'n qatar aqi'ri'nda '\n' belgisi qo'yi'ladi':

“Kompilyator ha'r bir qatar ushi'n kompyuter yadi'nda \ qatar uzi'nli'g'i'na ten' sandag'i' baytlardag'i' bo'lek \ yad ajratadi' ha'm bir - 0 ma'nisli' bayt qosadi'”;

Joqari'dag'i' ush qatarda jazi'lg'an qatar keltirilgen. Teris yon si'zi'q('\') belgisi keyingi qatarda jazi'lg'an belgiler izbe-izligin joqari'dag'i' qatarg'a qosi'w kerekligin an'latadi'. Yeger qosi'latug'i'n qatar baslani'wi'nda probeller bolsa, wolar da qatar qurami'na kiredi.

Qatar yadta jaylasqanda woni'n' aqi'ri'na '\0'(0 kodli' belgi) qosi'ladi' ha'm bul belgi qatar tawsi'lg'anli'g'i'n an'latadi'. Usi'ni'n' sebepinen qatar uzi'nli'g'i', woni'n' «haqiqiy» ma'nisinen birge ko'p boladi'.

Berilgenler tu'rleri ha'm wo'zgeriwshiler: Programma atqari'li'wi' waqti'nda qanday da berilgenlerdi saqlap turi'w ushi'n wo'zgeriwshiler ha'm turaqli'lardan paydalani'ladi'. Wo'zgeriwshi-programma obykti boli'p, yaddag'i' bir neshe yacheykaldardi iyeleydi ha'm berilgenlerdi saqlaw ushi'n xi'zmet yetedi. Wo'zgeriwshi atqa, wo'lshemge ha'm basqa atributlarg'a-ko'rinis oblasti', a'mel yetiw waqi'ti' ha'm basqa ayri'qshali'qlarg'a iye boladi'. Wo'zgeriwshilerdi isletiw ushi'n wolar a'l'bette dag'aza yetiliwi kerek. Dag'aza na'tiyjesinde wo'zgeriwshi ushi'n yaddan qanday da tarawdi'n' rezervlanadi, tarawdi'n' wo'lshemi bolsa wo'zgeriwshinin' konkret tu'rine baylani'sli' boladi'. Soni' atap ko'rsetiw za'ru'rki, bir tu'rge ha'r qi'yli' apparat platformalarda ha'r qi'yli'sha jay ajrati'li'wi' mu'mkin.

Wo'zgeriwshi dag'azasi' woni'n' tu'rin ani'qlawshi' gilt so'zi menen baslanadi' ha'm '=' belgisi arqali' baslang'i'sh ma'nis beriledi (sha'rt yemes). Bir gilt so'z menen bir neshe wo'zgeriwshilerdi ja'riyalaw mu'mkin. Woni'n' ushi'n wo'zgeriwshiler bir-birinen ', ' belgisi menen ajrati'ladi'. Dag'azalar ';' belgisi menen tawsi'ladi'. Wo'zgeriwshi ati' 255 belgiden oshmasligi kerek.

C++ tilinin' tayansh tu'rleri: C++ tilinin' tayansh tu'rleri, wolardi'n' baytlardag'i' wo'lshemleri ha'm ma'nisleri'ni'n' shegaralari' 1.2-kestede keltirilgen.

1.2-keste. C++ tilinin' tayansh tu'rleri

Tu'r ati'	Baytlardag'i' wo'lshemi	Ma'nis shegarasi'
bool	1	true yamasa false
unsigned short int	2	0...65535
short int	2	-32768...32767
unsigned long int	4	0..42949667295
long int	4	-2147483648...2147483647
int(16 razryadli)	2	-32768... 32767
int(32 razryadli)	4	-2147483648...2147483647
unsigned int (16 razryadli)	2	0...65535
unsigned int (32 razryadli)	4	0...42949667295
unsigned char	1	0...255
char	1	-128...127
float	4	1.2E-38...3.4E38
double	8	2.2E-308...1.8E308
long double (32 razryadli)	10	3.4e-4932...-3.4e4932
void	2 yamasa 4	bos tu'r

Pu'tin san tu'rleri. Pu'tin san ma'nislerdi' qabi'l yetetug'i'n wo'zgeriwshiler int(pu'tin), short(qi'sqa) ha'm long(uzi'n) gilt so'zler menen ani'qlanadi'. Wo'zgeriwshi ma'nisleri' belgili boli'wi' yamasa unsigned gilt so'zi menen belgisiz son si'pati'nda qarali'wi' mu'mkin(1-qosi'mshag'a qarang).

Belgi tu'ri. Belgi tu'rindagi wo'zgeriwshiler char gilt so'zi menen beriledi ha'm wolar wo'zinde belginin' ASCII kodi'n saqlaydi'. Belgi, tu'rindagi ma'nisler sali'sti'rg'anda quramali' bolg'an strukturalar-qatarlar, belgiler massivleri ha'm basqalardi payda yetiwde isletiledi(2-qosi'mshag'a qarang).

Haqi'yqi'y san tu'ri. Haqi'yqi'y sanlar float gilt so'zi menen dag'aza yetiledi. Bul tu'rdegi wo'zgeriwshi ushi'n yadta 4 bayt jay ajrati'ladi' ha'm <ishora><ta'rtip><mantissa> qa'lipinde sandi' saqlaydi'. Yeger bo'lshekli san u'lken(kishi) ma'nislerdi' qabi'l yetetug'i'n bolsa, wol yaddi' 8 yamasa 10 baytda yekilengen ani'qli'q ko'rinishinde saqlanadi' ha'm saykes double ha'm long double gilt so'zleri menen dag'aza yetiledi. Son'g'i' jag'day 32-razryadli platformalar ushi'n wori'nli'(1-qosi'mshag'a qaran').

Logikali'q tu'r. Bul tu'rdegi wo'zgeriwshi bool gilt so'zi menen dag'aza yetiledi. Wol tu'rdegi wo'zgeriwshi 1 bayt jay iyeleydi ha'm 0(false, jalg'an) ma'nisinen ayri'qsha ma'nis(true, ras) qabi'l yetedi. Logikali'q tu'r wo'zgeriwshiler ma'nisler wortasi'ndagi' munasa'betlerdi an'latatug'i'n woy-pikirlerdi ras(true) yamasa jalg'an(false) yekenligin suwretlewinde qollaniladi ha'm wolar qabi'l yetetug'i'n ma'nisler matematikali'q logika ni'zamli'qlari'na tiykarlanadi'.

Matematikali'q logika-pikrlewdin' formasi' ha'm ni'zamli'atlari haqqi'ndagi' pa'n. Woni'n' tiykari'n woy-pikirler yesabi' sko'lkemlestiredi. Woy-pikir - bul qa'legen ga'p boli'p, wog'an sali'sti'rg'anda ras yamasa jalg'an pikrdi xabar beriw mu'mkin. Mi'sali' «3>2», «5-jup san», « Tashkent-Wo'zbekistan paytaxti» ha'm basqa. Lekin «0.000001 kishi san» ga'pi woy-pikir yesaplanmaydi, sebebi «kishi san» tu'sinigi ju'da sali'sti'rmali' esaplani'p, yag'ni'y kishi san degende qanday sandi' tu'siniw kerekligi

ani'q yemes. Sol sebepli joqari'dag'i' ga'pti ras yamasa jalg'an ekenligi haqqi'nda xabar beriw qi'yi'n.

Woy-pikirlerdin' rasligi jag'daylarg'a baylani'sli' tu'rde wo'zgeriwi mi'sali' «bugin-sha'rshembi» ga'pin ras yamasa wo'tirikligi qarali'pati'rg'an ku'nge baylani'sli'. Tap sonday « $x < 0$ » ga'pi x wo'zgeriwshisinin' usi' waqi'ttag'i' ma'nisine saykes tu'rde ras yamasa jalg'an boladi'.

C++ tilinde logikali'q tu'r ati' Angliyalı'q matematikalı'q Jorj Bul hu'rmetine bool so'zi menen ko'rsetilgen. Logikali'q a'meller «Bul algebrasi» dep ataladi'.

Logikali'q woy-pikirler u'stinde ush a'mel ani'qlang'an:

1) biykar - A woy-pikirdi biykari' degende A ras bolg'anda jalg'an ha'm jalg'an bolg'anda ras ma'nis qabi'l yetiwshi woy-pikirge ayti'ladi'. C++ tilinde biykar-'!' belgisi menen beriledi. Mi'sali', A woy-pikir biykari' «!A» ko'rinishinde jazi'ladi';

2) konyuksiya - yeki A ha'm B woy-pikirler konyuksiyasi yamasa logikali'q ko'beymesi «A && B» ko'rinishke iye. Bul woy-pikir tek A ha'm B woy-pikirler ras bolg'andag'ana ras boladi', bolmasa jalg'an boladi'(a'detde «&&» a'meli «ha'm» dep woqi'ladi'). Mi'sali' «bugin aydi'n' 5 ku'ni ha'm bugin sha'rshembi» woy-pikiri aydi'n' 5 ku'ni sha'rshembi bolg'an ku'n ushi'ng'ana ras boladi';

3) dizyunksiya-yeki A ha'm B woy-pikirler dizyunksiyasi yamasa logikali'q ji'yi'ndi'si' «A || B» ko'rinishke jazi'ladi'. Bul woy-pikir ras boli'wi' ushi'n A yamasa B woy-pikirlerden biri ras boli'wi' jetkilikli. A'detde «||» a'meli «yamasa» dep woqi'ladi'.

Joqari'da keltirilgen pikirler tiykari'nda logikali'q a'meller ushi'n rasli'q kestesini ani'qlang'an(1.3-keste).

1.3-keste. Logikali'q a'meller ushi'n rasli'q kestesini

Woy-pikirler		Woy-pikirler u'stinde a'meller		
A	B	!A	A&&B	A B
false	false	true	false	false
false	true	true	false	true
true	false	false	false	true
true	true	false	true	true

Logikali'q tu'r ma'nisleri' u'stinde logikali'q ko'beytiw, qosi'w ha'm biykar a'mellerin qo'llaw arqali' quramali' logikali'q an'latpalardi' quri'w mu'mkin. Mi'sal ushi'n, «x-won' ha'm wol ma'nisi [1..3] sanlar aralig'ina tiyisli yemes» woy-pikirin logikali'q an'latpa ko'rinisi to'mendegishe boladi':

$$(x>0) \ \&\&(y<1 \ || \ y>3)$$

void tu'ri. void tu'rindegi programma obykti hesh qanday ma'nisge iye bolmaydi' ha'm bul tu'rden til sintaksisina sa'ykes keliwin ta'miyinlew ushi'n isletiledi. Mi'sali', C++ tili sintaksisi funksiya ma'nis qaytari'wi'n talap yetedi. Yeger funksiya ma'nis qaytarmaytug'i'n bolsa, wol void gilt so'zi menen dag'aza yetiledi.

Mi'sallar.

```
int a=0, A=1;
float abs=17.5;
double Koren;
bool ok=true;
char LETTER='z';
void menin_funksiyam();/*funksiya qaytaratug'in ma'nis
inabatqa ali'nbaydi' */
```

Tu'rlengen turaqli'lar: Tu'rlengen turaqli'lar tap wo'zgeriwshiler si'yaqli' isletiledi ha'm initsializatsiya yetilgennen(baslang'i'sh ma'nis berilgennen) keyin wolardi'n' ma'nisin wo'zgertirip bolmaydi'. Tu'rlengen turaqli'lar const gilt so'zi menen dag'aza yetiledi, wodan keyin turaqli' tu'ri ha'm a'l'bette initsializatsiya bo'legi boli'wi' kerek.

Mi'sal si'pati'nda tu'rlengen ha'm literal turaqli'lardan paydalang'an halda radius berilgende shen'ber ju'zesin yesaplaytug'i'n programmani keltiremiz.

```
#include <iosream.h>
int main()
{
    const double pi=3.1415;
    const int radius=3;
    double square=0;
    square=pi*radius*radius;
```

```

        cout<<square<<'\n';
        return 0;
    }

```

Programma bas funksiyasi'ni'n' baslani'wi'nda yeki pi ha'm radius turaqli'lari' dag'aza yetilgen. Shen'ber ju'zesin ani'qlawshi' square turaqli' dep dag'aza yetilmegen, sebebi wol programma atqari'li'wi'nda wo'zgeredi. Shen'ber radiusi'n programma islewinde wo'zgertiw qaratilmag'an, usi'ni'n' sebebini wol turaqli' si'pati'nda dag'aza yetilgen.

Sanap o'tiliwshi tu'r: Ko'p mug'dardag'i, logikali'q baylani'sqan turaqli'lardan paydalang'anda sanap o'tiliwshi tu'rden paydalani'lg'ani' maqul. Sanap o'tiliwshi turaqli'lar enum gilt so'zi menen ani'qlanadi'. Mazmuni' boyi'nsha bul turaqli'lar a'piwayi' pu'tin sanlar boli'p tabi'ladi'. Sanap o'tiliwshi turaqli'lar C++ standarti' boyi'nsha pu'tin tu'rdegi turaqli'lar yesaplanadi'. Ha'r bir turaqli'g'a(sang'a) mazmunli at beriledi ha'm bul identifikatordi programmanin' basqa jaylari'nda ataw ushi'n isletiliwi mu'mkin yemes. Sanap o'tiliwshi tu'r tomendegi ko'rinishke iye:

```

enum<Sanap_o'tiletug'in_tu'r_ati'>{<at1=ma'nis1>,
    <at2=ma'nis2>, ...<atn>=<ma'nisn>};

```

Bul jerde, enum-gilt so'z(inglizshe enumerate-nomerlew);

<Sanap_o'tiletug'in_tu'r_ati'> - turaqli'lar diziminin' ati'; <at> - pu'tin ma'nisli' konstantalardin' atlari'; <ma'nis> - sha'rt bolmag'an initsializatsiya ma'nisi(an'latpa).

Mi'sal ushi'n ha'pte ku'nleri menen baylani'sli' ma'sele sheshiwde ha'pte ku'nleri du'y(du'yshembi), shiy(shiyshenbi), shar(sha'rshembi), bys(byshenbe), juma(juma), shembi(shembi), yaksh(yekshembi) turaqli'lari'n isletiw mu'mkin ha'm wolar sanap o'tiliwshi tu'r ja'rdeminde bir qatarda jazi'ladi':

```

enum Hapte{duy, shiy, shar, bys, juma, shembi, yaksh};

```

Sanap o'tiliwshi turaqli'lar to'mendegi wo'zgeshelikke iye: yeger turaqli' ma'nisi ko'rsetilmegen bolsa, wol aldi'ng'i' turaqli' ma'nisinen birge arti'q boladi'. Sha'rtlesiw boyi'nsha birinshi turaqli' ma'nisi 0 boladi'.

Initsializatsiya ja'rdeminde turaqli' ma'nisin wo'zgertiw mu'mkin:

```
enum Hapte{duy=8, shiy, shar=12, bys=13, juma=16, shembi,  
yaksh=20};
```

Bul dag'azada shiy ma'nisi 9, shembi bolsa 17 ge ten' boladi'.

Sanap o'tiliwshi turaqli'lardi'n' atlari' har-xil boli'wi' kerek, lekin wolardi'n' ma'nisleri' birdey boli'wi' mu'mkin:

```
enum{nol=0, taza=0, bir, yeki, jup=2, ush};
```

Turaqli'ni'n' ma'nisi an'latpa ko'riniste beriliwi mu'mkin, tek an'latpadag'i' atlardi'n' ma'nisleri' sol adimg'asha ani'qlang'an boli'wi' kerek:

```
enum {eki=2, tort=eki*2};
```

Turaqli'di' ma'nisleri' kerri san boli'wi' da mu'mkin:

```
enum {minus2 = -2, minus1, nol, bir} ;
```

Tu'r di basqa tu'rge keltiriw: C++ tilinde bir tu'r di basqa tu'rge keltiriwdn' oshkor ha'm oshkormas jollarina iye .

Uluwma alg'anda, tu'r di basqa tu'rge oshkormas keltiriw an'latpada har-xil tu'rdegi wo'zgeriwshiler qatnasqan jag'daylarda a'mel yetedi(aralas tu'rler arifmetikasi). Ayi'ri'm jag'daylarda, atap aytqanda tayansh tu'rler menen baylani'sli' tu'rge keltiriw a'mellerinde aljasi'qlar ju'zege keliwi mu'mkin. Mi'sali', yesaplaw na'tiyjesinin' yaddan waqti'nsha iyelegen jayi' uzi'nli'g'i', woni' wo'zlestiretug'i'n wo'zgeriwshi ushi'n aji'rati'lg'an jay uzi'nli'g'i'nan u'lken bolsa, ma'nisge iye razryadlardi' joyti'w varianti ju'z beredi.

Oshkor tu'rde tu'rge keltiriwde, wo'zgeriwshi aldi'na qawi's ishinde basqa tu'r ati' jazi'ladi':

```
#include <iostream.h>
```

```
int main()
```

```
{
```

```
    int integer_1=54;
```

```
    int integer_2;
```

```
    float floating=15.854;
```

```
    integer_1=(int) floating; // oshkor keltiriw;
```

```
    integer_2=(int) floating // oshkormas keltiriw;
```

```
    cout<<"jan'a integer(oshkor) : "<<integer_1<<"\n";
```

```
    cout<<"jan'a integer(oshkormas) :"<<integer_2<<"\n";
```

```
    return 0;
}
```

Programma na'tiyjesi to'mendegi ko'rinishinde boladi':

Jan'a integer(oshkor) :15

Jan'a integer(oshkormas) :15

Ma'sele. Berilgen belginin' ASCII kodi' baspag'a shig'arilsin. Ma'sele belgi tu'rindagi ma'nisti oshkor tu'rde pu'tin san tu'rine keltirip baspadan shi'g'ari'w arqali' sheshiledi.

Programma teksti:

```
#include <iostream.h>
int main()
{
    unsigned char A;
    cout<<"belgini kiritin:'";
    cin>>A;
    cout<<" ' " <<A<<" ' -belgi ASCII kodi=" <<(int)A<<"\n";
    return 0;
}
```

Programma na'tiyjesi to'mendegi ko'rinishinde boladi':

Belgi kiritin':

A <enter>

'A'-belgi ASCII kodi=65

Soraw ha'm tapsi'rmalar

1. Turaqli'lar.
2. Pu'tin san tu'rleri.
3. Haqi'yqi'y san tu'rleri.
4. Logikali'q tu'r.
5. Logikali'q woy-pikirler u'stinde a'meller.
6. Sanap o'tiliwshi turaqli'lar.
7. C++ tilinde bir tu'rdi basqa tu'rge keltiriw.
8. Qatar turaqli'lar.
9. C++ tilinin' tayansh tu'rleri.

10. *void* tu'ri.

11. *Pu'tin* turaqli'lar.

12. *Haqi'yqi'y* turaqli'lar.

Variables of type **int** can hold integer quantities that do not require fractional components. Variables of this type are often used for controlling loops and conditional statements. Variables of the types **float** and **double** are employed either when a fractional component is required or when your application requires very large or small numbers. The difference between a **float** and a **double** variable is the magnitude of the largest (and smallest) number that each one can hold. As shown in Table 3-1, a **double** in C++ can store a number approximately ten times larger than a **float**.

Type	Typical Bit Width	Typical Range
char	8	-128 to 127
wchar_t	16	0 to 65,535
int (16-bit environments)	16	-32,768 to 32,767
int (32-bit environments)	32	-2,147,483,648 to 2,147,483,647
float	32	3.4E-38 to 3.4E+38
double	64	1.7E-308 to 1.7E+308
bool	N/A	true or false
void	N/A	valueless

Type	Bit Width	Common Range
char	8	-128 to 127
unsigned char	8	0 to 255
signed char	8	-128 to 127
int	16	-32,768 to 32,767
unsigned int	16	0 to 65,535
signed int	16	-32,768 to 32,767
short int	16	same as int
unsigned short int	16	same as unsigned int
signed short int	16	same as short int
long int	32	-2,147,483,648 to 2,147,483,647
unsigned long int	32	0 to 4,294,967,295
signed long int	32	-2,147,483,648 to 2,147,483,647
float	32	3.4E-38 to 3.4E+38
double	64	1.7E-308 to 1.7E+308
long double	80	3.4E-4932 to 1.1E+4932
bool	N/A	true or false
wchar_t	16	0 to 65,535

Declaration of Variables

The general form of a variable declaration statement is shown here:

```
type variable_list;
```

Here, *type* must be a valid C++ data type, and *variable_list* may consist of one or more identifier names separated by commas. Some declarations are shown here, for example:

```
int i, j, k;  
char ch, chr;  
float f, balance;  
double d;
```

In C++, the name of a variable has nothing to do with its type.

Standard C++ states that at least the first 1,024 characters of any identifier name (including variable names) will be significant. This means that if two variable names differ in at least one character within the first 1,024 characters, then the compiler will consider them to be different names.

There are three places where variables will be declared: inside functions, in the definition of function parameters, and outside of all functions. These variables are called *local variables*, *formal parameters*, and *global variables*, respectively. Although we will examine the importance of these three different types of variables in greater detail later in this book, let's take a brief look at them now.

Local Variables

Variables that are declared inside a function are local variables. They can be used only by statements that are inside that function. Local variables are not known to functions outside their own. Consider this example:

```
#include <iostream>  
using namespace std;  
  
void func();  
  
int main()
```

7-Lekciya: C++ tilinde a'meller ha'm ma'nis beriw operatori'

Joba:

1. Arifmetikali'q a'meller. Ma'nis beriw operatori'
2. An'latpa tu'sinigi
3. Inkrement ha'm dekrement a'melleri
4. sizeof a'meli

Tayansh so'zler: *An'latpalar, operatorlar, arifmetikali'q a'meller, ma'nis beriw operatori', til ko'rsetpesi, inkrement, dekrement, sizeof.*

Arifmetikali'q a'meller. Ma'nis beriw operatori': Berilgenlerdi qayta islew ushi'n C++ tilinde a'mellerdin' ju'da' ken' kompleksi ani'qlang'an. A'mel-bul qanday da ha'reket boli'p, wol bir(unar) yamasa yeki(binar) operandlar u'stinde atqari'ladi', yesap na'tiyjesi woni'n' qaytaratug'i'n ma'nisi yesaplanadi'.

Tayansh arifmetikali'q a'mellerga qosi'w(+), ayi'ri'w(-), ko'beytiw(*), boli'w(/) ha'm boli'w qaldi'g'ini ali'w(%) a'mellerin keltiriw mu'mkin.

A'meller qaytaratug'i'n ma'nislerdi' wo'zlestiriw ushi'n ma'nis beriw a'meli(=) ha'm woni'n' ha'r qi'yli' modifikatsiyalari' isletiledi: qosi'w, ma'nis beriw menen(+=) ; ayi'ri'w, ma'nis beriw menen(-=) ; ko'beytiw, ma'nis beriw menen(*=) ; boli'w, ma'nis beriw menen(/=) ; boli'w qaldi'g'ini ali'w, ma'nis beriw menen(%=) ha'm basqalar. Bul jag'daylardin' uli'wma ko'rinisi:

`<o'zgeriwshi> <a'mel>= <an'latpa>;`

To'mendegi programma tekstinde ayi'ri'm a'mellerga mi'sallar keltirilgen.

```
#include <iostream.h>
int main()
{
    int a=0, b=4, c=90;
```

```

char z='\t';
a=b; cout<<a<<z;           //a=4
a=b+c+c+b; cout<<a<<z;    //a=4+90+90+4=188
a=b-2; cout<<a<<z;        //a=2
a=b*3 cout<<a<<z;         //a=4*3=12
a=c/(b+6); cout<<a<<z;    //a=90/(4+6)=9
cout<<a%2<<z;             //9%2=1
a+=b; cout<<a<<z;         //a=a+b=9+4=13
a*=c-50; cout<<a<<z;      //a=a*(c-50)=13*(90-50)=520
a-=38; cout<<a<<z;       //a=a-38=520-38=482
a%=8; cout<<a<<z;        //a=a%8=482%8=2
return 0;
}

```

Programma atqari'li'wi' na'tiyjesinde ekrang'a to'mendegi sanlar qatari' payda boladi':

4 188 2 12 9 1 482 2

An'latpa tu'sinigi: C++ tilinde an'latpa-a'meller, operandlar ha'm punktatsiya belgilerinin' izbe-izligi boli'p, kompilyator ta'repinen berilgenler u'stinde ma'lum bir a'mellerdi wori'nlawg'a ko'rsetpe yesaplanadi'. Ha'r qanday ';' belgi menen tawsi'latug'i'n an'latpag'a til ko'rsetpesi dep ataladi'.

C++ tilindegi an'latpag'a(til ko'rsetpesine) mi'sal:

```

x=3*(y-2.45);
u=summa(a, 9, c);

```

Inkrement ha'm dekrement a'melleri: C++ tili operand ma'nisin birge asi'ri'w ha'm kemeytiwdin' na'tiyjeli qurallari'na iye . Bular inkrement(++ ha'm dekrement unar a'meller boli'p tabi'ladi'.

Operandqa sali'sti'rg'anda bul a'mellerdin' prefiks ha'm postfiks ko'rinishleri boladi'. Prefiks ko'rinishde a'mel til ko'rsetpesi boyi'nsha jumi's atqari'li'wi'nan aldi'n operandqa qollani'ladi'. Postfiks jag'dayda bolsa a'mel til ko'rsetpesi boyi'nsha jumi's wori'nlang'annan keyin operandqa qollani'ladi'.

Prefiks yamasa postfiks a'mel tu'sinigi tek ma'nis beriw menen baylani'sli' an'latpalarda wori'nli':

```
x=u++; // postfiks
index =--i; // prefiks
count++; //unar a'mel, " ++count; " menen ekvivalent
abc--; //unar a'mel, " --abc; " menen ekvivalent
```

Bul jerde wol wo'zgeriwshinin' ma'nisin x wo'zgeriwshisine wo'zlestiriledi ha'm keyin birge asi'ri'ladi', i wo'zgeriwshinin' ma'nisi birge kemeytirilip, index wo'zgeriwshisine wo'zlestiriledi.

sizeof a'meli: Har-xil tu'rdegi wo'zgeriwshiler kompyuter yadi'nda ha'r qi'yli' sandag'i' baytlardi iyeleydi. Bunda, ha'tte bir tu'rdegi wo'zgeriwshiler da qaysi' kompyuterde yamasa qaysi' operatsion sistemada a'mel yetiliwine qaray ha'r qi'yli' wo'lshemdegi yadni' ba'nd yetiwi mu'mkin.

C++ tilinde qa'legen(tayansh ha'm tuwi'ndi') tu'rdegi wo'zgeriwshilerdin' wo'lshemin sizeof a'meli ja'rdeminde ani'qlanadi'. Bul a'meldi turaqli'g'a, tu'rge ha'm wo'zgeriwshige qollani'wi' mu'mkin.

To'mende keltirilgen programmada kompyuterdin' platformasiga saykes tu'rde tayansh tu'rlerinin' wo'lshemleri baspadan shi'g'ari'ladi'.

```
int main()
{
    cout<<"int tu'rinin' wo'lshemi:"<<sizeof(int)<<"\n";
    cout<<"float tu'rinin' wo'lshemi:" <<sizeof(float)<<"\n";
    cout<<"double tu'rinin' wo'lshemi:"<<sizeof(double)<<"\n";
    cout<<"char tu'rinin' wo'lshemi:"<<sizeof(char)<<"\n";
    return 0;
}
```

Soraw ha'm tapsi'rmalar.

1. A'meller.
2. C++ tilinde an'latpalar.
3. Inkrement ha'm dekrement a'melleri.

4. *sizeof* a'meli.
5. Tayansh arifmetikali'q a'meller.
6. Sali'sti'ri'w a'melleri.
7. C++ tilinde du'zilgen arifmetikali'q an'latpalarg'a mi'sallar keltiring.
8. Inkrement a'melinin' qollani'li'wi'na tiyisli mi'sal keltirig.
9. Dekrement a'melinin' qollani'li'wi'na tiyisli mi'sal keltirig.
10. *sizeof* a'melinin' qollani'li'wi'.

Variable Initializations

You can assign a value to a variable at the same time that it is declared by placing an equal sign and the value after the variable name. The general form of initialization is:

```
type variable_name = value;
```

Some examples are:

```
char ch = 'a';  
int first = 0;  
float balance = 123.23F;
```

Although variables are frequently initialized by constants, you can initialize a variable by using any expression valid at the time of the initialization. As you will see, initialization plays an important role when you are working with objects.

Global variables are initialized only at the start of the program. Local variables are initialized each time the function in which they are declared is entered. All global variables are initialized to zero if no other initializer is specified. Local variables that are not initialized will have unknown values before the first assignment is made to them.

Here is a simple example of variable initialization. This program uses the **total()** function to compute the summation of the value that it is passed. In other words, **total()** sums the digits from 1 to the value. For example, the summation of 3 is $1 + 2 + 3$, or 6. In the process, **total()** displays a running total. Notice the use of the **sum** variable in **total()**.

Operators

C++ is rich in built-in operators. An *operator* is a symbol that tells the compiler to perform specific mathematical or logical manipulations. C++ has three general classes of operators: *arithmetic*, *relational* and *logical*, and *bitwise*. In addition, C++ has some special operators for particular tasks. This chapter will examine the arithmetic, relational, and logical operators, reserving the more advanced bitwise operators for later.

Arithmetic Operators

Table 3-4 lists the arithmetic operators allowed in C++. The operators $+$, $-$, $*$, and $/$ all work the same way in C++ as they do in any other computer language (or algebra, for that matter). These can be applied to any built-in data type allowed by C++. When $/$ is applied to an integer or a character, any remainder will be truncated; for example, $10/3$ will equal 3 in integer division.

Operator	Action
-	subtraction, also unary minus
+	addition
*	multiplication
/	division
%	modulus
--	decrement
++	increment

8-Lekciya: Razryadli', logikali'q ha'mde ta'kirarlaw a'melleri

Joba:

1. Razryadli' logikali'q a'meller
2. Shepke ha'm won'g'a ji'lji'ti'w a'melleri
3. Sali'sti'ri'w a'melleri
4. «U'tir» a'meli
5. A'mellerdin' u'stemshilikleri ha'm atqari'li'w bag'darlari'

Tayansh so'zler: *Razryadli logikali'q a'meller, bayraqlar, shepke ji'lji'ti'w, won'g'a ji'lji'ti'w, sali'sti'ri'w, u'tir, a'mellerdin' u'stemshilikleri, atqari'li'w bag'dari'.*

Razryadli' logikali'q a'meller: Programma du'ziw ta'jiriybesi soni' ko'rsetedi, a'detde qoyi'lg'an ma'seleni sheshiwde qandayda bir jag'day ju'z bergen yamasa joq yekenligin an'lati'w ushi'n 0 ha'm 1 ma'nis qabi'l yetiwshi bayraqlardan paydalani'ladi'. Bul maqsette bir yamasa wodan arti'q baytli wo'zgeriwshilerden paydalani'w mu'mkin. Mi'sali', bool tu'rindegi wo'zgeriwshin sol maqsette isletsa boladi'. Basqa ta'repden, bayraq si'pati'nda bayttin' razryadlari'nan paydalani'w da mu'mkin. Sebebi razryadlar tek yeki ma'nisti - 0 ha'm 1 sanlari'n qabi'l yetedi. Bir baytda 8 razryad bolg'ani' ushi'n wol jag'dayda 8 dana bayraqti' kodlaw mu'mkinshiligine iye.

Shama menen woylayi'q, qorg'aw sistemasi'na 5 dana bo'lme jalg'ang'an ha'm sistema taxtasi'nda 5 dana shi'rasha(indikator) bo'lmeler jag'dayi'n an'latadi': bo'lme qorg'aw sistemasi' qadag'alawi'nda yekenligin saykes indikatorin' jani'p turi'wi'(razryadti'n' 1 ma'nisi) ha'm xanani' sistemag'a jalg'anbag'anli'g'i'n indikator o'shkenligi(razryadni'n' 0 ma'nisi) an'latadi'. Sistema jag'dayi'n an'lati'w ushi'n bir bayt jetkilikli boladi' ha'm woni'n' kishi razryadi'nan baslap besewin sol maqsette isletiw mu'mkin:

7 6 5 4 3 2 1 0

			ind	ind	ind	ind	ind
			5	4	3	2	1

Mi'sali', bayttin' to'mendegi varianti 1, 4 ha'm 5 bo'lmeler qorg'aw sistemasi'na jalg'ang'anli'g'i'n an'latadi':

7	6	5	4	3	2	1	0
x	x	x	1	1	0	0	1

To'mendegi kestede C++ tilinde bayt razryadlari' u'stinde logikali'q a'meller kompleksi keltirilgen.

3.1-keste. Bayt razryaddari u'stinde logikali'q a'meller

A'meller	Mazmuni'
&	Logikali'q HA'M(ko'beytiw)
	Logikali'q yamasa(qosi'w)
^	Biykar yetiwshi YAMASA
~	Logikali'q BIYKARLAW(inversiya)

Razryadli' logikali'q a'mellerdin' wori'nlaw na'tiyjelerin keste ko'rinishinde ko'rsetiw mu'mkin.

3.2-keste. Razryadli' logikali'q a'mellerdin' wori'nlaw na'tiyjeleri

A	B	C=A&B	C=A B	C=A^B	C=~A
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

Joqari'dag'i' keltirilgen mi'sal ushi'n qorg'aw sistemasi'n an'latiwshi' bir baytli char tu'rindagi wo'zgeriwshin ja'riyalaw mu'mkin:

char k=0;

Bul jerde k wo'zgeriwshisine 0 ma'nis beriw arqali' barli'q bo'lmeler qorg'aw sistemasi'na jalg'anbag'anli'g'i' an'latpalanadi':

7	6	5	4	3	2	1	0
0	0		0	0	0	0	0

Yeger 3-xanani' sistemag'a jalg'aw za'ru'r bolsa

$$k = k \ 0x04_{16};$$

a'melin wori'nlaw kerek, sebebi $0x04_{16}=00000100_2$ ha'm logikali'q YAMASA a'meli na'tiyjesinde k wo'zgeriwshisi bayti to'mendegi ko'riniste boladi':

7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0

Tap sonday jol menen basqa bo'lmelerdi sistemag'a jalg'aw mu'mkin, za'ru'r bolsa birdaniga yekewin(za'ru'r bolsa barli'g'i'n) :

$$k=k \ | \ 0x1F_{16};$$

Logikali'q ko'beytiw arqali' bo'lmelerdi qorg'aw sistemasi'nan shi'g'ari'w mu'mkin:

$$k=k \ \& \ 0xFD_{16}; \ //0xFD_{16}=11111101_2$$

Tap sol na'tiyjeni '~' a'meliden paydalang'an halda da ali'w mu'mkin. Yekinshi bo'lme sistemag'a jalg'ang'anli'g'i' bildiruvchi bayt ma'nisi- 00000010_2 , sonday yeken sol jag'dayni' biykar yetken halda logikali'q ko'beytiwdi wori'nlaw kerek.

$$k=k \ \& \ (\sim 0x02);$$

Ha'm aqi'r-aqi'betde, yeger 3-bo'lme indikatorini, woni' qanday ma'nisda boli'wi'nan qaramastan keru jag'dayg'a wo'tkeriwdi «biykar yetiwshi YAMASA» a'meli ja'rdeminde wori'nlaw mu'mkin:

$$k=k \ \wedge \ 0x04; \ //0x04_{16}=00000100_2$$

Razryadli' logikali'q a'mellerdi ma'nis beriw operatori' birge atqari'li'wi'ni'n' to'mendegi ko'rinisleri bar:

&=-razryadli' HA'M ma'nis beriw menen;

|=-razryadli' YAMASA ma'nis beriw menen;

^=-razryadli' istisno yetiwshi YAMASA ma'nis beriw menen.

Shepke ha'm won'g'a ji'lji'ti'w a'melleri: Baytdagi bi'ytlar ma'nisin shepke yamasa won'g'a ji'lji'ti'w ushi'n, saykes tu'rde << ha'm >> a'melleri qollani'ladi'. A'melden keyingi san bi'ytlar neshe wori'n shepke yamasa o'nga ji'lji'ti'w kerekligin an'latadi'.

Mi'sali':

```
unsigned char A=12;      //A=000011002=0x0C16
A=A<<2;                 //A=001100002=0x3016=4810
A=A>>3;                 //A=000001102=0x0616=610
```

Razryadlardi' n ta shepke(o'nga) ji'lji'ti'w sandi' 2ⁿ sani'na ko'beytiw(boli'w) a'meli menen ekvivalent boli'p ha'm sali'sti'rg'anda tez atqari'ladi'. Soni' itibarg'a ali'w kerek, operand belgili san bolsa, wol jag'dayda shepke ji'lji'ti'wda yen' chapdagi belgi razryadi' ta'kirarlanadi'(belgi saqlani'p qaladi') ha'm kerri sanlar u'stinde bul a'mel wori'nlang'anda matematika noqati'-na'zerden qa'te na'tiyjeler ju'zege keledi:

```

unsigned char B=-120; // V=100010002=0x8816
V = V<<2; // V=001000002=0x2016=3210
V = -120; // V=100010002=0x8816
V = V>>3; // V=111100012=0xF116=-1510

```

Usi'ni'n' sebebini, bul razryadli' ji'lji'ti'w a'melleri belgisiz(unsigned) tu'rdegi ma'nisler u'stinde wori'nlang'ani' maqul.

Sali'sti'ri'w a'melleri: C++ tilinde ma'nislerdi' sali'sti'ri'w ushi'n sali'sti'ri'w a'melleri ani'qlang'an(3.3-keste). Sali'sti'ri'w a'meli binar a'mel boli'p, to'mendegi ko'rinishke iye:

<operand1> <sali'sti'ri'w a'meli> <operand2>

Sali'sti'ri'w a'mellerinin' na'tiyjesi-sali'sti'ri'w wori'nli' bolsa, true(ras), bolmasa false(jalg'an) ma'nis boladi'. Yeger sali'sti'ri'wda arifmetikali'q an'latpa qatnashsa, woni'n' ma'nisi 0 ma'nisinen ayri'qsha jag'daylar ushi'n 1 dep yesaplanadi'.

3.3-keste. Sali'sti'ri'w a'melleri ha'm wolardi'n' qollani'wi'

A'meller	Qollani'wi'	Mazmuni'(o'qiliwi)
<	a<b	"a kishi b"
<=	a<=b	"a kishi yamasa ten' b"
>	a>b	"a u'lken b"
>=	a>=b	"a u'lken yamasa ten'"
==	a==b	"a ten' b"
!=	a!=b	"a ten' yemas b"

«U'tir» a'meli: Programmatalasti'ri'wda bir neshe an'latpalardi' kompilyator ta'repinen pu'tin bir an'latpa dep qabi'llawi' ushi'n «u'tir» a'meli qollani'ladi'. Bul a'meldi qo'llaw arqali' programma jazi'wda ma'lum bir na'tiyjelilikke yerisiw mu'mkin. A'detde «u'tir»

a'meli `if` ha'm `for` operatorlari'nda ken' qollani'ladi'. Mi'sali', `if` operatori' qo'yidagi ko'riniste boli'wi' mu'mkin:

```
if(i=Callfunc(), i<7)...
```

Bul jerde, aldi'n `Callfunc()` funksiyasi' shaqi'ri'ladi' ha'm woni'n' na'tiyjesi `i` wo'zgeriwshisine wo'zlestiriledi, keyin `i` ma'nisi 7 menen sali'wti'ri'ladi'.

A'mellerdin' u'stemshilikleri ha'm atqari'li'w bag'darlari': Arifmetikadag'i'day C++ tilinde da a'meller ma'lum bir ta'rtip ha'm jo'neliste atqari'ladi'. Yekenin ayti'w kerek, matematikali'q an'latpalarda birdey u'stinlikdegi(prioritetdagi) a'meller u'shrasa(mi'sali', qosi'w ha'm ayi'ri'w), wolar shep ta'repten won'g'a atqari'ladi'. Bul ta'rtip C++ tilindegi da wori'nli', biraq ayi'ri'm jag'daylarda a'mel won' ta'repten shepke wori'nlawg'a bolatug'i'n(ma'nis beriw a'melinde).

An'latpalar ma'nisin yesaplawda a'meller u'stinligi yesapqa ali'nadi', Birinshi na'wbette yen' joqari' u'stinlikka iye bolg'an a'mel atqari'ladi'.

Operator	Suwretlewi	U'stinligi	Jo'nelis
::	Ko'rinish oblasti'na ruxsat beriw	16	=>
[]	Massiv indeksi	16	=>
()	Funksiyani' shaqi'ri'w	16	=>
->	Struktura yamasa klass elementin tan'law	16	=>
++	Postfiks inkrement	15	<=
--	Postfiks dekrement	15	<=
++	Prefiks inkrement	14	<=
--	Prefiks dekrement	14	<=
sizeof	Wo'lshemdi ali'w	14	<=
(<tur>)	Tu'rge akslantirish	14	
~	Bitli logikali'q INKOR	14	<=
!	Logikali'q biykar	14	<=
-	Unar minus	14	<=
+	Unar plyus	14	<=
&	Adresdi ali'w	14	<=
*	Quralli' murajat	14	<=

new	Dinamikali'q obyektни jarati'w	14	<=
delete	Dinamikali'q obyektни joq yetiw	14	<=
sasting	Tu'rge keltiriw	14	
*	Ko'beytiw	13	=>
/	Boli'w	13	=>
%	Boli'w qaldi'g'i	13	=>
+	Qosi'w	12	=>
-	Ayi'ri'w	12	=>
>>	Razryad boyi'nsha won'g'a ji'lji'ti'w	11	=>
<<	Razryad boyi'nsha shepke ji'lji'ti'w	11	=>
<	Kishi	10	=>
<=	Kishi yamasa ten'	10	=>
>	U'lken	10	=>
>=	U'lken yamasa ten'	10	=>
==	Ten'	9	=>
!=	Ten' yemes	9	=>
&	Razryadli' HA'M	8	=>
^	Razryadli' istisno yetiwshi YAMASA	7	=>
	Razryadli' YAMASA	6	=>
&&	Logikali'q HA'M	5	=>
	Logikali'q yamasa	4	=>
?:	Sha'rt a'meli	3	<=
=	Ma'nis beriw	2	<=
*=	Ko'beytiw ma'nis beriw a'meli menen	2	<=
/=	Boli'w ma'nis beriw a'meli menen	2	<=
%=	Modulli boli'w ma'nis beriw a'meli menen	2	<=
+=	Qosi'w ma'nis beriw a'meli menen	2	<=
-=	Ayi'ri'w ma'nis beriw a'meli menen	2	<=
<<=	Shepke ji'lji'ti'w ma'nis beriw a'meli menen	2	<=
>>=	won'g'a ji'lji'ti'w ma'nis beriw a'meli menen	2	<=
&=	Razryadli' ha'm ma'nis beriw menen	2	<=
^=	Razryadli' istisno yetiwshi yamasa ma'nis beriw menen	2	<=
=	Razryadli' yamasa ma'nis beriw menen	2	<=
throw	Istisno jag'dayni' ju'zege keltiriw	2	<=
,	u'tir	1	<=

C++ tili programma du'ziwshisine a'mellerdin' atqari'li'w rejimin wo'zgertiw mu'mkinshiligin beredi. Tap matematikadag'i' a'mellerdi qawi'slar ja'rdeminde gruppalarg'a jiynaw mu'mkin. Qawi's isletiwge shegara joq.

To'mendegi programmada qawi's ja'rdeminde a'mellerdi wori'nlaw rejimin wo'zgertiw ko'rsetilgen.

```
#include <iostream.h>
int main()
{
    int x=0, y=0;
    int a=3, b=34, c=82;
    x=a*b+c;
    y=(a*(b+c) ) ;
    cout<<"x="<<x<<"n\";
    cout<<"y="<<y<<"n\";
}
```

Programmada a'meller u'stinligine ko're x ma'nisin yesaplawda aldi'n a wo'zgeriwshi b wo'zgeriwshige ko'paytiriladiva wog'an c wo'zgeriwshi ma'nisine qosi'ladi'.

Na'wbettegi ko'rsetpeni wori'nlawda bolsa birinshi na'wbette ishki qawi's ishindegi an'latpa $-(b+c)$ ma'nisi yesaplanadi', keyin bul ma'nis a ko'paytirilib, wol wo'zgeriwshisine wo'zlestiriledi.

Programma atqari'li'wi' na'tiyjesinde ekrang'a

x=184

u=348

qatarlari' baspadan shi'g'ari'ladi'.

Soraw ha'm tapsi'rmalar

1. C++ tilinde bayt razryadlari' u'stinde atqari'latug'i'n a'meller.
2. "<<" a'melin ani'qlama berin'.
3. ">>" a'melin ani'qlama berin'.
4. Sali'sti'ri'w a'melleri.
5. "u'tir" a'melinin' qollani'li'wi'.

6. Razryadli logikali'q a'mellerdi ma'nis beriw operatori' menen birge atqari'li'wi'.
7. Yesaplawda a'mellerdin' u'stinligi.
8. A'mellerdin' atqari'li'w bag'darlari'.

Increment and Decrement

C++ has two operators not found in some other computer languages. These are the increment and decrement operators, ++ and --. These operators were mentioned in passing in Chapter 2, when the **for** loop was introduced. The ++ operator adds 1 to its operand, and -- subtracts 1. Therefore,

```
x = x+1;
```

is the same as

```
++x;
```

and

```
x = x-1;
```

is the same as

```
--x;
```

Both the increment and decrement operators can either precede (prefix) or follow (postfix) the operand. For example:

```
x = x+1;
```

can be written as

```
++x; // prefix form
```

or as

```
x++; // postfix form
```

In the foregoing example, there is no difference whether the increment is applied as a prefix or a postfix. However, when an increment or decrement is used as part of a larger expression, there is an important difference. When an increment or decrement operator precedes its operand, C++ will perform the corresponding operation prior to obtaining the operand's value for use by the rest of the expression. If the operator follows its operand, then C++ will obtain the operand's value before incrementing or decrementing it. Consider the following:

```
x = 10;  
y = ++x;
```

The precedence of the arithmetic operators is shown here:

highest	++ --
	- (unary minus)
	* / %
lowest	+ -

The operands for a relational operator can be of nearly any type, as long as they can be compared. The operands to the logical operators must produce a true or false result. Because any non-zero value is true and zero is false, the logical operators can be used with any expression that evaluates to a zero or non-zero result.

Relational Operators	
Operator	Meaning
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to
==	equal to
!=	not equal to
Logical Operators	
Operator	Meaning
&&	AND
	OR
!	NOT

The logical operators are used to support the basic logical operations AND, OR, and NOT, according to the following truth table. The table uses 1 for true and 0 for false.

p	q	p AND q	p OR q	NOT p
0	0	0	0	1
0	1	0	1	1
1	1	1	1	0
1	0	0	1	0

3

Although C++ does not contain a built-in exclusive-OR (XOR) logical operator, it is easy to construct one. The XOR operation uses this truth table:

p	q	XOR
0	0	0
0	1	1
1	0	1
1	1	0

In words, the XOR operation produces a true result when one, and only one, operand is true. The following function uses the `&&` and `||` operators to construct an XOR operation. The result is returned by the function.

9-Lekciya: Sha'rt ha'm tan'law operatorlari'

Joba:

1. Operator tu'sinigi
2. Sha'rt operatorlari'
3. **if** operatori'
4. **if-else** operatori'
5. **?: sha'rt a'meli**
6. **switch** operatori'

Tayansh so'zler: *Programma atqari'li'wi'n basqari'w, si'zi'qli' operator, basqari'w operatori', an'latpa operatori', dag'aza operatori', sha'rt operatori', if operatori', blok if-else operatori', ?: sha'rt a'meli, switch operatori'.*

Operator tu'sinigi: Programmatalasti'ri'w tili operatorlari' sheshilipati'rg'an ma'sele algoritmin a'melge asi'ri'w ushi'n isletiledi. Operatorlar si'zi'qli' ha'm basqari'w operatorlari'na bo'linedi. Ko'pshilik jag'daylarda operatorlar noqatli' u'tir(';') belgisi menen tamamlanadi' ha'm wol kompilyator ta'repinen bo'lek operator dep qabi'l yetiledi(for operatori'ni'n' qawi's ishinde turg'an an'latpalari' bunnan ayri'qsha). Bunday operator an'latpa operatori' dep ataladi'. Ma'nis beriw a'melleri topari', atap aytqanda, ma'nis beriw operatorlari' an'latpa operatorlari' yesaplanadi':

```
i++;  
--j;  
k+=i;
```

Programma du'ziw a'meliyati'nda bos operator ';' isletiledi. Biraq, bul operator hesh narse atqarmasa da, yesaplaw an'latpalari'n til qurallari'na sa'ykes keliwin ta'miyinleydi. Ayi'ri'm jag'daylarda ju'zege kelgen quramali' jag'daylardan shi'g'i'p ketiw imkaniyati'n beredi.

Wo'zgeriwshilerdi ja'riyalaw da operator yesaplanadi' ha'm wolarg'a dag'aza operatori' dep ataladi'.

Sha'rt operatorlari': Aldi'ng'i' leksiyada mi'sal menende keltirilgen

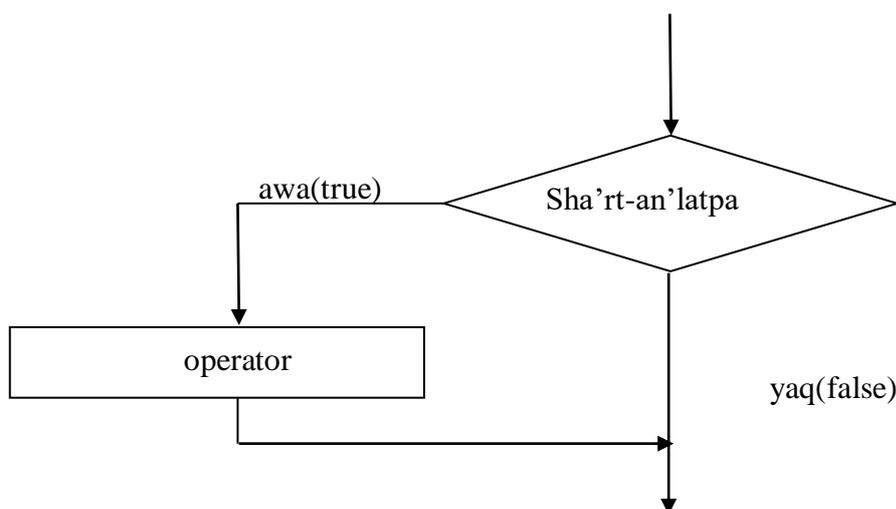
programmalarda a'meller jazi'li'w rejiminde izbe-iz ha'm tek bir ret atqari'latug'in jag'daylar, yag'ni'y si'ziqli' algoritmlar keltirilgen yedi. A'melde bolsa kemnen-kem ma'seleler sheshiliwi mu'mkin. Ko'pshilik ma'seleler ju'zege keletug'in ha'r qi'yli' jag'daylarg'a baylani'sli' tu'rde saykes qarar qabi'llawdi'(sheshimdi) talap yetedi. C++ tili programmanin' bo'lek bo'leklerinin' atqari'li'w rejimin basqari'wg'a mu'mkinshilik jarati'wshi' qurallardi'n' jetkiliklishe u'lken kompleksine iye. Mi'sali', programma atqari'li'wi'ni'n' qandayda-bir qa'deminde qanday da sha'rtti tekseriw na'tiyjesine ko're basqari'wdi' programmanin' wol yamasa bul bolegine uzati'w mu'mkin(tarmaqlani'wshi' algoritm). Tarmaqlanishni a'melge asi'ri'w ushi'n sha'rtli operatoridan paydalani'ladi'.

if operatori: if operatori' qanday da sha'rtti rasli'qqa tekseriwi na'tiyjesine ko're programmada tarmaqlaniwdi a'melge asi'radi':

```
if(<shart> )
    <operator>;
```

Bul jerde <shart> ha'r qanday an'latpa boli'wi' mu'mkin a'detde wol sali'sti'ri'w a'meli boladi'.

Yeger sha'rt 1 ma'nisinen ayri'qsha yamasa ras(true) bolsa, <operator> atqari'ladi', bolmasa, yag'ni'y sha'rt 0 yamasa jalg'an(false) bolsa, hesh qanday a'mel atqari'lmaydi' ha'm basqari'w if operatori'nan keyingi operatorg'a wo'tedi. Usi' jag'day 4.1-suwretde ko'rsetilgen.



Su'wret-1. if() sha'rt operatori'ni'n' blok sxemasi'

C++ tilinin' operatorlardi' blok ko'riniside sho'lkemlestiriwge mu'mkinshilik beredi. Blok - '{' ha'm '}' belgi arali'g'ina ali'ng'an operatorlar izbe-izligi boli'p, wol kompilyator ta'repinen pu'tin bir operator dep qabi'l yetiledi. Blok ishinde dag'aza operatorlari' da boli'wi' mu'mkin ha'm wolar da dag'aza yetilgen wo'zgeriwshiler tek sol blok ishinde ko'rinedi(a'mel yetedi), bloktan si'rtda ko'rinbeydi. Bloktan keyin ';' belgisi qoyi'lmasligi mu'mkin, lekin blok ishindegi ha'r bir an'latpa ';' belgisi menen juwmaqlani'wi' sha'rt.

To'mende keltirilgen programmada if operatori'nan paydalani'w ko'rsetilgen.

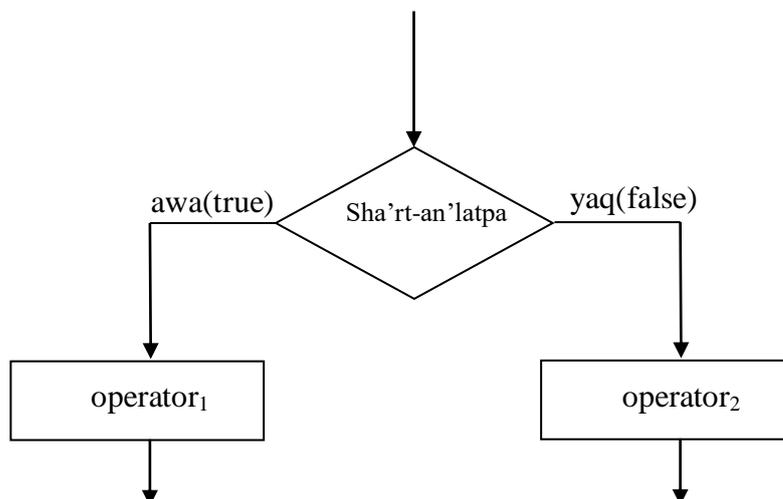
```
#include <iostream.h>
int main()
{
    int b;
    cin>>b;
    if(b>0)
    {
        //b>0 sha'rt wori'nlang'an jag'day
        ...
        cout<<"b-won' san";
        ...
    }
    if(b<0)
        cout<<"b-keri san";//b<0 sha'rt wori'nlang'an jag'day
    return 0;
}
```

Programma atqari'li'wi' processinde pu'tin tu'rdegi b wo'zgeriwshi dag'aza yetiledi ha'm woni'n' ma'nisi klaviaturadan woqi'ladi'. Keyin b ma'nisin 0 sani'nan u'lkenligi tekseriledi, yeger sha'rt atqari'lsa(true), wol halda '{' ha'm '}' belgiler ishindegi operatorlar atqari'ladi' ha'm ekrang'a "b-won' san" xabari' shi'g'adi'. Yeger sha'rt atqari'lmasa, bul operatorlar sheklep wo'tiledi. Na'wbettegi sha'rt operatori' b wo'zgeriwshi ma'nisi kerilikke tekseredi, yeger sha'rt atqari'lsa, birden-bir cout ko'rsetpesi atqari'ladi' ha'm ekrang'a "b-keri san" xabari' shi'g'adi'.

if-else operatori: Sha'rt operatori'ni'n' if-else ko'rinisi to'mendegishe:

```
if(<shart-an'latpa>
    <operator1>;
else
    <operator2>;
```

Bul jerde <sha'rt-an'latpa> 0 ma'nisinen ayri'qsha yamasa true bolsa <operator1>, bolmasa <operator2> atqari'ladi'. if-else sha'rt operator mazmuni'na ko're algoritmdin' tarmaqlani'wshi' bloki'n an'latadi': <sha'rt-an'latpa>-sha'rt bloki'(romb) ha'm <operator1> blokni'n' «awa» shaqasi, <operator2> bolsa blokni'n' «yaq» shaqasina saykes keliwshi a'meller bloklari' dep qaraw mu'mkin.



Su'wret-2. if(), else sha'rt operatori'ni'n' blok sxemasi'

Mi'sal si'pati'nda diskriminantni yesaplaw usili' ja'rdeminde $ax^2+bx+c=0$ ko'rinisindegi kvadrat ten'leme korenlerini tabi'w ma'selesin ko'reylik:

```
#include <iostream.h>
#include <math.h>
int main()
{
    float a, b, c;
    float D, x1, x2;
    cout<<"ax^2+bx+c=0; ten'leme korenlerin tabi'w.";
    cout<<"\n a-koiffitsiyentin kiritin': ";
```

```

cin>>a;
cout<<"\n b-koeffitsientin kiritin' : ";
cin>>b;
cout<<"\n c-koeffitsientin kiritin' : ";
cin>>c;
D=b*b-4*a*c;
if(D<0)
{
    cout<<"ten'leme haqi'yqi'y korenge iye yemes!";
    return 0;
}
if(D==0)
{
    cout<<"ten'leme birden-bir korenge iye: ";
    x1=-b/(2*a) ;
    cout<<"\nx="<<x1;
    return 0;
}
else
{
    cout<<"ten'leme bir korenge iye: ";
    x1=(-b+sqrt(D) ) /(2*a) ;
    x1=(-b-sqrt(D) ) /(2*a) ;
    cout<<"\nx1="<<x1;
    cout<<"\nx1="<<x1;
}
return 0;
}

```

Programma wori'nlang'anda, birinshi na'wbette ten'leme koeffitsientlari a , b , c wo'zgeriwshiler ma'nisleri' kiritiledi, keyin diskriminanti D wo'zgeriwshi ma'nisi yesaplanadi'. Keyin D ma'nisinin keriyekenligi tekseriledi. Yeger sha'rt wori'nli' bolsa, pu'tin operator si'pati'nda keliwshi '{' ha'm '}' belgileri arasi'ndag'i' operatorlar atqari'ladi' ha'm ekrang'a "Ten'leme haqi'yqi'y korenlerge iye yemes" xabari' shi'g'adi' ha'm programma wo'z jumi'si'n toqtatadi'(return 0; operatori'n wori'nlaw arqali'). Diskriminant nolden kishi bo'lmasa, na'wbettegi sha'rt operatori'

woni' nolge ten'ligin tekseredi. Yeger sha'rt wori'nli' bolsa, keyingi qatarlardag'i' operatorlar bloki' atqari'ladi' - ekrang'a " Ten'leme bir koreng'e iye:" xabari', ha'mde x1 wo'zgeriwshi ma'nisi baspadan shi'g'ari'ladi' ha'm programma bulmanda wo'z jumi'si'n toqtatadi, bolmasa, yag'ni'y D ma'nisti nolden u'lken varianti ushi'n else gilt so'zinen keyingi operatorlar bloki atqari'ladi' ha'm ekrang'a "Ten'leme yeki koreng'e iye:" xabari', ha'mde x1 ha'm x2 wo'zgeriwshiler ma'nisleri' baspadan shi'g'ari'ladi'. Usi'ni'n' menen sha'rt operatori'nan shi'g'i'ladi' ha'm tiykarg'i' funksiyani'n' return ko'rsetpesin wori'nlaw arqali' programma wo'z jumi'si'n toqtatadi'.

Wo'z gezeginde <operator1> ha'm <operator2> da sha'rtli operator boli'wi' mu'mkin. An'latpadag'i' ha'r bir else gilt so'zi, aldag'i' bawi'rlas if gilt so'zine tiyisli yesaplanadi'(tap ashi'latug'i'n ha'm yopiluvchi qawi'slar si'yaqli'). Buni' inabatqa almaw mazmunan aljasi'qlarg'a ali'p keliwi mu'mkin.

Mi'sali':

```

if(x==1)
    if(y==1)
        cout<<"x=1 ha'm y=1";
    else
        cout<<"x<>1";

```

Bul mi'salda «x<>1» xabari' x ma'nisin 1 ga ten' ha'm wol ma'nisin 1 ga ten' bolmag'an halda da baspadan shi'g'ari'ladi'. To'mendegi variantda usi' mazmunan qa'telik du'zetilgen:

```

if(x==1)
{
    if(y==1)
        cout<<"x=1 ha'm y=1";
}
else
    cout<<"x<>1";

```

Yekinshi mi'salda ush pu'tin sanni'n' maksimal ma'nisin tabatug'i'n programma bo'legini keltiriwimiz mu'mkin:

```

...
int x, y, z, max;

```

```

cin>>x>>y>>z;
if(x>y)
    max=x;
else
    max=y;

if(max<z)
    max=z;

cout<<max<<endl;

```

...

Sha'rt operatori'nda ja'riyalaw operatorlari'n isletiw qadag'an yetiledi, lekin wodag'i bloklarda wo'zgeriwshilerdi ja'riyalaw mu'mkin ha'm bul wo'zgeriwshiler tek blok ishinde a'mel yetedi. To'mendegi mi'salda bul jag'day menen baylani'sli' qa'telik ko'rsetilgen:

```

if(j>0)
{
    int i;
    i=2*j;
}
else
    i=-j;//qa'te, sebebi i blokdan si'rtida ko'rinbeydi

```

Ma'sele. Berilgen to'rt xanali' belgisiz sanni'n' basi'ndag'i' yeki nomerinin' ji'yi'ndi'si' qalg'an nomerler ji'yi'ndi'si'na ten' yamasa joq yekenligi aniqlansin(nomerler ji'yi'ndi'si' degende wolarg'a saykes san ma'nisleri'ni'n' ji'yi'ndi'si' tu'siniledi). Sanni'n' nomerlerin aji'rati'p ali'w ushi'n pu'tin sanlar arifmetikasi a'mellerinen paydalani'ladi':

```

#include <iostream.h>
int main()
{
    unsigned int n, a3, a2, a1, a0;
    cout<<"\n n-ma`nisin kiritin:'";
    cin>>n;

```

```

if(n<1000 || n>9999)
{
    cout<<"kiritilgen san 4 xanali' yemes!";
    return 0;
}
a3=n/1000;
a2=n%1000/100;
a1=n%100/10;
a0=n%10;
if(a3+a2==a1+a0)
    cout<<"a3+a2=a1+a0";
else
    cout<<"a3+a2<>a1+a0";
return 0;
}

```

Programma belgisiz pu'tin san kirgiziwdi usi'ni's yetedi. Yeger kiritilgen san 4 xanali' bo'lmasa($n < 1000$ yamasa $n > 9999$), bul haqqi'nda xabar beriledi ha'm programma wo'z jumi'si'n toqtatadi'. Bolmasa n sani'ni'n' nomerleri aji'rati'p ali'nadi', ha'mde basi'ndag'i' yeki nomerdin' ji'yi'ndi'si(a_3+a_2) qalg'an yeki nomerler ji'yi'ndi'si(a_1+a_0) menen sali'wtiriladi ha'm wolardi'n' ten' yamasa joq yekenligine qaray saykes juwap baspadan shi'g'ari'ladi'.

?: sha'rt a'meli: Yeger tekserilip ati'rg'an sha'rt sali'sti'rg'anda a'piwayi' bolsa, sha'rt a'melinin' $\langle\langle?:\rangle\rangle$ ko'rnishini isletiw mu'mkin:

```
<shart an'latpa> ? <an'latpa1> : <an'latpa2>;
```

Sha'rt a'meli if sha'rt operatori'na uqsaw halda isleydi: yeger $\langle\text{shart an'latpa}\rangle = 0$ ma'nisinen ayri'qsha yamasa true bolsa, $\langle\text{an'latpa1}\rangle$, bolmasa $\langle\text{an'latpa2}\rangle$ atqari'ladi'. A'detde an'latpalar ma'nisleri' qandayda-bir wo'zgeriwshige wo'zlestiriledi.

Mi'sal jol menenda yeki pu'tin son maksimumin ani'qlawdi ko'reyik.

```

#include <iostream.h>
int main()
{
    int a, b, c;
    cout<<"a ha'm b sanlar maksimumin ani'qlaw programmasi.";
}

```

```

    cout<<"\n a-ma'nisti kiritin': ";
    cin>>a;
    cout<<"\n b-ma'nisti kiritin': ";
    cin>>b;
    c=a>b?a:b;
    cout<<"\n Sanlar maksimumi: "<<c;
    return 0;
}

```

Programmada g'i' sha'rt operatori' ma'nis beriw operatori'ni'n' qurami'na kirgen boli'p, a wo'zgeriwshinin' ma'nisi b wo'zgeriwshinin' ma'nisinen u'lkenligi tekseriledi, yeger sha'rt ras bolsa c wo'zgeriwshige a wo'zgeriwshi ma'nisin, bolmasa b wo'zgeriwshinin' ma'nisin ha'm c wo'zgeriwshinin' ma'nisi baspadan shi'g'ari'ladi'.

?: a'melinin' ma'nis qaytari'w wo'zgesheligiden paydalang'an halda, woni' tikkeley cout ko'rsetpesine jazi'w arqali' da qoyi'lg'an ma'seleni sheshiw mu'mkin:

```

#include <iostream.h>
int main()
{
    int a, b;
    cout<<"a ha'm b sanlar maksimumini tabi'w programmasi.";
    cout<<"\n a-ma'nisti kiritin': ";
    cin>>a;
    cout<<"\n b-ma'nisti kiritin': ";
    cin>>b;
    c=a>b?a:b;
    cout<<"\n Sanlar maksimumi: "<<(a>b) ?a:b;
    return 0;
}

```

switch operatori': Sha'rt operatorinin' ja'ne bir ko'rinishi switch tarmaqlani'w operatori' boli'p, woni'n' sintaksisi to'mendeneshe:

```

switch(<an'latpa>)
{
    case <o'zgermes an'latpa1>:<operatorlar topari'1>; break;
    case <o'zgermes an'latpa2>:<operatorlar topari'2>; break;
    ...
}

```

```

    case <o'zgermes an'latpan>:<operatorlar topari'n>; break;
    default: <operatorlar topari'n+1>;
}

```

Bul operator to'mendegishe a'mel yetedi: birinshi na'wbette <an'latpa> ma'nisi yesaplanadi, keyin bul ma'nis case gilt so'zi menen aji'rati'lg'an <o'zgarmas an'latpai> menen sali'wti'ri'ladi'. Yeger wolar ustma-u'st tushsa, sol qatardag'i ' : ' belgisinen baslap, toki' break gilt so'zine shekem bolg'an <operatorlar topari_i> atqari'ladi' ha'm basqari'w tarmaqlani'wshi' operatoridan keyingi jaylasqan operatorg'a wo'tedi. Yeger <an'latpa> qandayda-bir da <o'zgermes an'latpa_i> menen saykes kelmasa, operatoridin' default bo'legidegi <operatorlar topari'n+1> atqari'ladi'. Soni' atap ko'rsetiw za'ru'r, operatorida default gilt so'zi tek bir ret du's keliwi mu'mkin.

Mi'sal ushi'n, kiriw ag'i'mi'nan "Process dawam etilsinbe?" sorawi'na paydalani'wshi' ta'repinen juwap ali'nadi'. Yeger unamli' juwap olinsa, ekrang'a "Process dawam yetedi!" xabari' baspadan shi'g'ari'ladi' ha'm programma wo'z jumi'si'n tarmaqlani'wshi' operatoridan keyingi operatorlardi' wori'nlaw menen dawam etiriladi, bolmasa "Process tamamlandi!" juwapi' beriledi ha'm programma wo'z jumi'si'n toqtatadi'. Bunda, paydalani'wshi'ni'n 'y' yamasa 'Y' juwaplari' processti dawam yettiriwdi an'latadi', basqa belgiler bolsa tamamlawdi' an'latadi'.

```

#include <iostream.h>
int main()
{
    char juwap=' ';
    cout<<"Process dawam etsinmi?('y', 'Y') : ";
    cin>>juwap;
    switch(juwap)
    {
        case 'y':
        case 'Y':
            cout<<" Process dawam yetedi!\n";
            break;
        default:

```

```

        cout<<" Process tamamlandi!\n";
        return 0;
    }
    ... // process
    return 0;
}

```

Uluwma alg'anda, tarmaqlani'wshi' operatorlarda break ha'm default gilt isletiw sha'rt yemes. Lekin bul halda operator'di'n' mazmuni' mu'mkin. Mi'sali', default bo'legi bolmag'an halda, yeger <an'latpa> qandayda-bir <o'zgarmas an'latpai> menen ustma-u'st tushmasa, operator hesh qanday a'mel atqarmasta'n basqari'w tarmaqtani'wshi' operator'dan keyingi operator'ga wo'tedi. Yeger break bo'lmasa, <an'latpa> qandayda-bir <o'zgarmas an'latpai> menen ustma-u'st tu'sken halda, wog'an saykes keliwshi operatorlar topari'n atqaradi' ha'm «toqtatpastan» keyingi qatardag'i' operatorlar topari'n wori'nlawg'a wo'tip ketadi. Mi'sali', joqari'dag'i' mi'salda break operatori' bo'lmasa ha'm processti dawam yettiriwni tastiylaytug'i'n('Y') juwap bolg'an ta'g'dirde ekrang'a

```

    Process dawam yetedi!

```

```

    Process tugadi!

```

Xabarlarini shi'g'adi' ha'm programma wo'z jumi'si'n toqtatadi'(return operatorini'n' atqari'li'wi' na'tiyjesinde).

Tarmaqlani'wshi' operator sanap o'tiliwshi tu'rdegi turaqli'lar menen birge isletilingende na'tiyje beredi. To'mendegi programmada ren'ler gammasi'n kategoriyalaw ma'selesi sheshilgen.

```

#include <iostream.h>
int main()
{
    enum Renler {qizil, toq_sari, sari, jasil, kok,
    binapshe_guli};
    Renler ren;
    //...
    switch(ren')
    {
        case qizil:
        case toq_sariq:

```

```

    case sari;
        cout<<"I'ssi' gamma saylanadi'. \n";
        break;
    case jasil:
    case kok:
    case binapshe_guli:
        cout<<"Suwi'q gamma tan'landi'.\n";
        break;
    default:
        cout<<"Woq jay bunday ren'ge iye yemes. \n";
    }
    return 0;
}

```

Programma atqari'li'wi'nda basqari'w tarmaqlani'wshi' operatorg'a kelip, ren' ma'nisi qi'zi'l yamasa toq_sari yamasa sari' bolsa, "I'ssi' gamma tan'landi'" xabari', yeger ren' ma'nisi jasi'l yamasa ko'k yamasa bina'pshe gu'li bolsa, ekrang'a "Suwi'q gamma tan'landi'" xabari', yeger ren' ma'nisi sanap wo'tilgen ma'nislerdan ayri'qsha bolsa, ekrang'a "Woq jay bunday ran'ge iye yemes" xabari' baspadan shi'g'ari'ladi' ha'm programma wo'z jumi'si'n toqtatadi'.

switch operatori'nda dag'aza operatorlari' da du's keliwi mu'mkin. Lekin switch operatori' atqari'li'wi'nda «sekirip wo'tiw» jag'daylari' boli'wi' yesabi'na blok ishindegi ayi'ri'm dag'azalar atqari'lmasligi ha'm buni'n' a'qibetinde programma jumi'si'nda qa'telik ju'z beriwi mu'mkin:

```

//...
int k=0, n=0;
cin>>n;
switch(n)
{
    int=10;//qa'te, bul operator hesh qashan atqari'lmaydi'
    case 1:
    int j=20;//yeger n=2 bolsa, bul dag'aza atqari'lmaydi'
    case 2:
    k+=i+j;//qa'te, sebebi i, j wo'zgeriwshiler belgisiz
}
cout<<k;
//...

```

Ma'sele. To'mende, sanap o'tiliwshi tu'rler ha'm sol tu'rdegi wo'zgeriwshiler dag'aza yetilgen:

```
enum Birlik{detsimetr,kilometr,metr,millimetr,santimetr};
float y;
Birlik r;
```

Birlikde berilgen x wo'zgeriwshisinin' ma'nis metrlerde baspadan shig'arilsin.

```
#include <iostream.h>
int main()
{
    enum Birlik{detsimetr,kilometr,metr,millimetr, santimetr};
    float x, y;
    Birlik r;
    cout<<"uzunlikni kiritin': x=";
    cin>>x;
    cout<<"uzi'nli'q birlikleri\n";
    cout<<"0-detsimetr\n";
    cout<<"1-kilometr\n";
    cout<<"2-metr\n";
    cout<<"3-millimetr\n";
    cout<<"4-santi'metr\n";
    cout<<"uzunlikni birligini saylan'; r=";
    cin>>r;
    switch(r)
    {
        case detsimetr:
            y=x/10;
            break;
        case kilometr:
            y=x*1000;
            break;
        case metr:
            y=x;
            break;
```

```

    case millimetr:
        y=x/1000;
        break;
    case santimetr:
        y=x/100;
        break;
    default:
        cout<<"uzunlik birligi naduri's kirgizildi!";
        return 0;
}
cout<<y<<"metr";
return 0;
}

```

Soraw ha'm tapsi'rmalar

1. Operatorlardi'n' tu'rleri.
2. *if* operatori'ni'n' uli'wma ko'rinisi.
3. *if-else* operatori'.
4. *switch* operatori'ni'n' wazi'ypasi'.
5. *switch* operatori'ni'n' qollani'li'wi'.
6. Sha'rt operatorlari'.
7. ?: operatori'.
8. *if* operatoridin' *if-else* operatori'nan ayi'rmashi'li'g'i'.
9. Dag'aza operatorlari'.
10. An'latpa operatorlari'.

The if Statement

Chapter 2 introduced the **if** statement. Now it is time to examine it in detail. The complete form of the **if** statement is

```
if(expression) statement;  
else statement;
```

where the targets of the **if** and **else** are single statements. The **else** clause is optional. The targets of both the **if** and **else** can be blocks of statements. The general form of the **if** using blocks of statements is

```
if(expression)  
{  
    statement sequence  
}  
else  
{  
    statement sequence  
}
```

If the conditional expression is true, the target of the **if** will be executed; otherwise, if it exists, the target of the **else** will be executed. At no time will both of them be executed. The conditional expression controlling the **if** may be any type of valid C++ expression that produces a true or false result.

The Conditional Expression

Sometimes newcomers to C++ are confused by the fact that any valid C++ expression can be used to control the **if**. That is, the type of expression need not be restricted to only those involving the relational and logical operators or to operands of type **bool**. All that is required is that the controlling expression evaluate to either a true or false result. As you should recall from the previous chapter, a value of zero is automatically converted into **false**, and all non-zero values are converted to **true**. Thus, any expression that results in a zero or non-zero value can be used to control the **if**. For example, the following program reads two integers from the keyboard and displays the quotient.

The conditional expressions are evaluated from the top downward. As soon as a true condition is found, the statement associated with it is executed, and the rest of the ladder is bypassed. If none of the conditions is true, then the final **else** statement will be executed. The final **else** often acts as a default condition; that is, if all other conditional tests fail, then the last **else** statement is performed. If there is no final **else** and all other conditions are false, then no action will take place.

The **?** is called a *ternary operator* because it requires three operands. It takes the general form

```
Exp1 ? Exp2 : Exp3;
```

where *Exp1*, *Exp2*, and *Exp3* are expressions. Notice the use and placement of the colon.

The value of a **?** expression is determined like this: *Exp1* is evaluated. If it is true, then *Exp2* is evaluated and becomes the value of the entire **?** expression. If *Exp1* is false,

The switch Statement

Before looking at C++'s other loop constructs, let's examine its other selection statement: the **switch**. Although a series of nested **if** statements can perform multiway tests, for many situations, a more efficient approach can be used. C++ has a built-in multiple-branch decision statement called **switch**. It works like this: the value of an expression is successively tested against a list of integer or character constants. When a match is found, the statement sequence associated with that match is executed. The general form of the **switch** statement is

```
switch(expression) {
    case constant1:
        statement sequence
        break;
    case constant2:
        statement sequence
        break;
    case constant3:
        statement sequence
        break;
    .
    .
    .
    default:
        statement sequence
}
```

The **switch** expression must evaluate to either a character or an integer value. (Floating-point expressions, for example, are not allowed.) Frequently, the expression controlling the **switch** is simply a variable.

The **default** statement sequence is performed if no matches are found. The **default** is optional; if it is not present, no action takes place if all matches fail. When a match is found, the statements associated with that **case** are executed until the **break** is encountered or, in the case of the **default** or the last **case**, the end of the **switch** is reached.

There are four important things to know about the **switch** statement:

- ◆ The **switch** differs from the **if** in that **switch** can test only for equality (i.e., for matches between the **switch** expression and the **case** constants), whereas the **if** conditional expression can be of any type.
- ◆ No two **case** constants in the same **switch** can have identical values. Of course, a **switch** statement enclosed by an outer **switch** may have **case** constants in common.
- ◆ A **switch** statement is usually more efficient than nested **ifs**.
- ◆ The statement sequences associated with each **case** are *not* blocks. However, the entire **switch** statement *does* define a block. The importance of this will become apparent as you learn more about C++.

Standard C++ specifies that a **switch** can have at least 16,384 **case** statements. In practice, you will want to limit the number of **case** statements to a much smaller total, for reasons of efficiency.

10-Lekciya: Ta'kirlaw operatorlari'

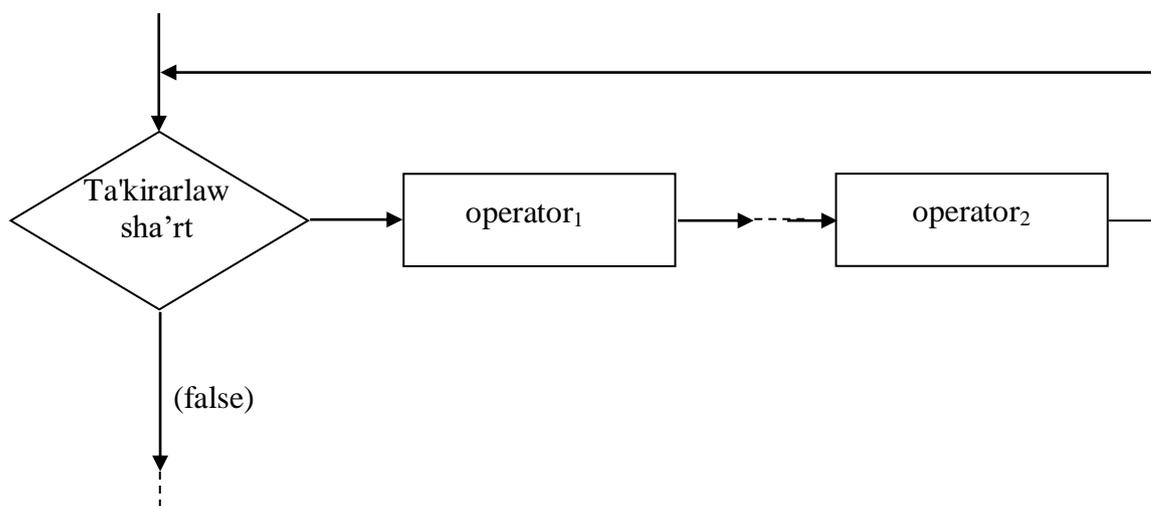
Joba:

1. **for** ta'kirlaw operatori'
2. **while** ta'kirlaw operatori'
3. **do-while** ta'kirlaw operatori'

Tayansh so'zler: *Ta'kirlaw operatorlari'*, *for ta'kirlaw operatori'*, *while ta'kirlaw operatori'*.

Programma atqari'li'wi'n basqari'wdi'n' basqa bir ku'shli mexanzmlarinan biri-ta'kirlaw operatorlari' yesaplanadi'.

Ta'kirlaw operatori' «takrorlaw sha'rti» dep atali'wshi an'latpani'n' ras ma'nisinde programmanin' ma'lum bir bo'legidegi operatorlardi'(ta'kirlaw denesi) ko'p ret ta'kirlaw tu'rde(itarativ process).



1-su'wret. Ta'kirlaw operatori'ni'n' blok sxemasi'

Ta'kirlaw wo'zinin' kiriw ha'm shi'g'i'w noqatlari'na iye, lekin shi'g'i'w noqati'ni'n' bolmaytug'i'ni' mu'mkin. Bul halda ta'kirlawg'a sheksiz ta'kirlaw dep ataladi'. Sheksiz ta'kirlaw ushi'n ta'kirlawni dawam yettiriw sha'rti mudami ras boladi'.

Ta'kirarlaw sha'rtini tekseriw ta'kirarlaw denesindegi operatorlardi' wori'nlawdan aldi'n tekseriliwi mu'mkin(for, while ta'kirarlawlari') yamasa ta'kirarlaw denesindegi operatorlari' bir ret wori'nlang'annan keyin tekseriliwi mu'mkin(do-while).

Ta'kirarlaw operatorlari' ishpe-ish jaylasqan boli'wi' mu'mkin.

for ta'kirarlaw operatori': for ta'kirarlaw operatori'ni'n' sintaksisi to'mendegi ko'rinske iye:

```
for(<an'latpa1>; <an'latpa2>; <an'latpa3>)  
    <operator yamasa blok>;
```

Bul operator wo'z jumi'si'n <an'latpa1> an'latpasi'n wori'nlawdan baslaydi'. Keyin ta'kirarlaw qa'demleri baslanadi'. Ha'r bir qa'demde <an'latpa2> atqari'ladi', yeger na'tiyje 0 ma'nisinen ayri'qsha yamasa true bolsa, ta'kirarlaw denesi - <operator yamasa blok> atqari'ladi' ha'm aqi'ri'nda <an'latpa3> atqari'ladi'. Yeger <an'latpa2> ma'nisi 0(false) bolsa, ta'kirarlaw procesi toqtaydi' ha'm basqari'w ta'kirarlaw operatori'nan keyingi operatorg'a wo'tedi. Soni' dizimnen wo'tkeriw kerek, <an'latpa2> an'latpasi' u'tir menen aji'rati'lg'an bir neshe an'latpalar birlespesinen ibarat boli'wi' mu'mkin, bul halda son'g'i' an'latpa ma'nisi ta'kirarlaw sha'rti yesaplanadi'. Ta'kirarlaw denesi si'pati'nda bir operator, atap aytqanda bos operator boli'wi' yamasa operatorlar bloki' keliwi mu'mkin.

Mi'sal ushi'n 10 dan 20 ge shekem bolg'an pu'tin sanlar ji'yi'ndi'si'n yesaplaw ma'selesin ko'reylik.

```
#include <iostream.h>  
int main()  
{  
    int Summa=0;  
    for(int i=10; i<=20; i++)  
        Summa += i;  
    cout<<"Summa = " << Summa;  
    return 0;  
}
```

Programmadagi ta'kirarlaw operatori' wo'z jumi'si'n, i ta'kirarlaw parametrine(ta'kirarlaw sanag'ishina) baslang'i'sh ma'nis - 10 sani'n

beriwden baslaydi' ha'm ha'r bir ta'kirarlaw qa'deminen(itaratsiyadan) keyin woni'n' ma'nisi birge artadi(qawi's ishindegi u'shinshi operator atqari'li'wi'n yesabi'na). Ha'r bir ta'kirarlaw qa'deminde ta'kirarlaw denesindegi operator atqari'ladi', yag'ni'y Summa wo'zgeriwshisine 1 ma'nisi qosi'ladi'. Ta'kirarlaw sanag'ishi i ma'nisi 21 bolg'anda " $i \leq 20$ " ta'kirarlaw sha'rti(0-ma'nisi) boladi' ha'm ta'kirarlaw tawsi'ladi'. Na'tiyjede basqari'w ta'kirarlaw operatori'nan keyingi cout operatori'na wo'tedi ha'm ekrang'a ji'yi'ndi' baspadan shi'g'ari'ladi'.

Joqari'da keltirilgen mi'salg'a qaray ta'kirarlaw operatorlari'ni'n' qawi's ishindegi an'latpalari'na tu'sindirime beriw mu'mkin:

<an'latpa1> - ta'kirarlaw sanag'ishi' wazi'ypasi'n wori'nlawshi' wo'zgeriwshisine baslang'i'sh ma'nis beriwge xi'zmet yetedi ha'm wol ta'kirarlaw procesi basi'nda tek bir ret yesaplanadi'. An'latpada wo'zgeriwshi dag'azasi' ushrasiwi mu'mkin ha'm bul wo'zgeriwshi ta'kirarlaw operatori' denesinde a'mel yetedi ha'm ta'kirarlaw operatori'nan si'rtida «ko'rinbeydi»(C++ Builder kopilyatori ushi'n) ;

<an'latpa2> - ta'kirarlawdi' wori'nlaw yamasa wori'nlanbawdi ani'qlap beretug'i'n logikali'q an'latpa, yeger sha'rt ras bolsa, ta'kirarlaw dawam yetedi, bolmasa joq. Yeger bul an'latpa bos bolsa, sha'rt natiyjesi ras dep yesaplanadi';

<an'latpa3>-a'detde ta'kirarlaw sanag'ishnin' ma'nisin asi'ri'w(kemeytiw) ushi'n xi'zmet yetedi yamasa wol jag'dayda ta'kirarlaw sha'rtine ta'sir basqa a'meller boli'wi' mu'mkin.

Ta'kirarlaw operatorida qawi's ishindegi an'latpalar bolmaytug'i'ni' mu'mkin, lekin sintaksis ';' bolmaytug'i'ni'na ruxsat bermiydi. Usi'ni'n' sebebini a'piwayi' ko'rinistegi ta'kirarlaw operatori' to'mendegishe boladi':

```
for(;;)
    cout<<"Sheksiz ta'kirarlaw ...";
```

Yeger ta'kirarlaw processinde bir neshe wo'zgeriwshilerdin' ma'nisi sinxron tu'rde wo'zgeriwi kerek bolsa, <an'latpa1> ha'm <an'latpa3>

an'latpalari'nda za'ru'r operatorlardi' ', ' menen jazi'w arqali' bug'an yerisiw mu'mkin:

```
for(int i=10, j=2; i<=20; i++, j=i+10)
{
    ...
}
```

Ta'kirarlaw operatori'ni'n' ha'r bir qa'deminde j ha'm i wo'zgeriwshi ma'nisleri' saykes tu'rde wo'zgerip baradi'.

for operatori'nda ta'kirarlaw denesi bolmaytug'i'ni' da mu'mkin. Mi'sali', programma atqari'li'wi'n ma'lum bir mu'ddetke «toqtap» turi'w za'ru'r bolsa, bug'an ta'kirarlawdi' hesh qanday qosi'msha jumi'slardi' atqarmasta'n a'mel yetiwi arqali' yerisiw mu'mkin:

```
#include <iostream.h>
int main()
{
    int delay;
    ...
    for(delay=5000; delay>0; delay--) ;// bos operator
    ...
    return 0;
}
```

Joqari'da keltirilgen 10 dan 20 ge shekem bolg'an sanlar ji'yi'ndi'si'n bos deneli(bos operatorli') ta'kirarlaw operatori' arqali' yesaplaw mu'mkin:

```
...
for(int i=10; i<=20; summa+=i++) ;
...
```

Ta'kirarlaw operatori' denesi si'pati'nda operatorlar bloki' isletiw in faktorialdi yesaplaw mi'sali'nda ko'rsetiw mu'mkin:

```
#include <iostream.h>
int main()
{
    int a;
    unsigned long fact=1;
    cout<<"Pu'tin san kiritin': ";
```

```

cin>>a;
if((a>=0) &&(a<3) )
{
    for(int i=1; i<=a; i++)
        fact*=i;
    cout<<a<<"!=" <<fact<<"\n";
}
return 0;
}

```

Programma paydalani'wshi' ta'repinen 0 den 33 ge shekem aralig'indag'i' san kiritilgende a'mel yetedi, sebebi 34! Ma'nisi unsigned long ushi'n aji'rati'lg'an razryadlarg'a jayg'aspaydi.

Ma'sele. Ta'kirarlaw operatori'ni'n' ishpe-ish jaylasi'wi'na mi'sal si'pati'nda nomerleri bir-birine wo'z-ara ten' bolmag'an u'sh xanali' natural sanlardi' wo'siw rejiminde baspadan shi'g'ari'w ma'selesin ko'riwimiz mu'mkin:

```

#include <iostream.h>
int main()
{
    unsigned char a2, a1, a0;// u'sh xanali' san nomerleri
    for(a2='1';a2<='9';a2++) //sandin' 2-wori'ndag'i' nomeri
    for(a1='0';a1<='9' ;a1++) //sandin' 1-wori'ndag'i' nomeri
        for(a0='0';a0<='9';a0++)//sandin' 0-wori'ndag'i' nomeri
            // nomerlerdi wo'z-ara ten' emasligini tekseriw
            if(a0!a1 && a1!=a2 && a0!=a2) //o'zaro ten' yemes
                cout<<a2<<a1<<a0<<"\n" ;
    return 0;
}

```

Programmada u'sh xanali' sandin' har bir nomeri ta'kirarlaw operatorlari' ja'rdeminde payda yetiledi. Birinshi, si'rtqi' ta'kirarlaw operatori' menen 2-xanadagi nomer(a2 ta'kirarlaw parametri) payda yetiledi. Yekinshi, ishki ta'kirarlaw operatori'nda(a1 ta'kirarlaw parametri) san ko'rinishinin' 1-reti kelgendedegi nomer ha'm aqi'r-aqi'betde, wog'an sali'sti'rg'anda ishki bolg'an a0 parametrli ta'kirarlaw operatori'nda 0-xanadag'i' nomerler payda yetiledi. Ha'r bir si'rtqi' ta'kirarlawdi'n' bir qa'demine ishki ta'kirarlaw operatori'ni'n' toli'q atqari'li'wi' tuwri' keledi.

while ta'kirarlaw operatori':

while takrarlaw operatori', operator yamasa blokti ta'kirarlaw sha'rti jalg'an(false yamasa 0) bolaman degenge shekem ta'kirar atqaradi'. Wol to'mendegi sintaksisga iye:

```
while(<an'latpa>)  
    <operator yamasa blok>;
```

Yeger <an'latpa> ras ma'nisli' turaqli' an'latpa bolsa, ta'kirarlaw sheksiz boladi'. Tap sonday, <an'latpa> ta'kirarlaw baslani'wi'nda ras boli'p, woni'n' ma'nisine ta'kirarlaw denesindegi yesaplaw ta'sir etmasa, yag'ni'y woni'n' ma'nisi turaqli'a, ta'kirarlaw sheksiz boladi'.

while ta'kirarlaw sha'rtini aldi'nan tekseriwshi ta'kirarlaw operatori' yesaplanadi'. Yeger ta'kirarlaw basi'nda <an'latpa> jalg'an bolsa, while operatori' qurami'ndag'i' <operator yamasa blok> bo'legi atqari'lmasta'n sheklep wo'tiledi.

Ayi'ri'm jag'daylarda <an'latpa> ma'nis beriw operatori' ko'rinishinde keliwi mu'mkin. Bunda ma'nis beriw a'meli atqari'ladi' ha'm na'tiyje 0 sali'wti'ri'ladi'. Na'tiyje nolden ayri'qsha bolsa, ta'kirarlaw dawam ettiriledi.

Yeger ras an'latpani'n' ma'nisi nolden ayri'qsha turaqli' bolsa, sheksiz ta'kirarlaw ju'z beredi. Mi'sali':

```
while(1) ; // sheksiz ta'kirarlaw
```

Tap for operatorina u'qsas, ', ' ja'rdeminde <an'latpa> da bir neshe a'meller sinxron tu'rde wori'nlaw mu'mkin. Mi'sali', san ha'm woni'n' kvadratlari'n baspadan shi'g'aratug'in programmada usi' jag'day ko'rsetilgen:

```
#include <iostream.h>  
int main()  
{  
    int n, n2;  
    cout<<'sonni kiritin'(1..10) :_';  
    cin>>n;  
    n++;  
    while(n--, n2=n*n, n>0)
```

```

        cout<<'n='<<n<<' n^2='<<n2<<endl;
    return 0;
}

```

Programmada ta'kirarlaw operatori atqari'li'wi'nda n sani' 1 ge shekem azayi'p baradi'. Ha'r bir qa'demde n ha'm woni'n' kvadrati' baspadan shi'g'ari'ladi'. Sog'an itibar beriw kerek, sha'rt an'latpasi'nda operatorlardi' jazi'li'w izbe-izliginin' a'hmiyeti bar, sebebi, yen' son'g'i' operator ta'kirarlaw sha'rti si'pati'nda qaraladi ha'm n ma'nisi 0 bolg'anda ta'kirarlaw tawsi'ladi'.

Keyingi programmada berilgen wonli'q sanni'n' yekilik ko'rinishin baspadan shi'g'ari'w ma'selesin sheshiwde sheshiwde while operatori'n qo'llaw ko'rsetilgen.

```

#include <iostream.h>
int main()
{
    int sanagish=4;
    short son10, process=1;
    while(process) // sheksiz ta'kirarlaw
    {
        cout<<"o'nlik san kiritin'(0..15): ";
        cin>>son10;
        cout<<"/n"<<son10<<"soninin' yekilik ko'rinishi:";
        while(sanagish)
        {
            if(son10&8) //son10&00001000
                cout<<'1';
            else
                cout<<'0';
            son10=son10<<1//razryadlardi wo'rin shepke
            ji'lji'ti'w
            sanagish--;
        }
        cout<<"\n";
        cout<<"processti toqtati'w(0), dawam yettiriw(1) :";
        cin>>process;
        sanagish=4;
    }
}

```

```

    return 0;
}

```

Programmada ishpe-ish jaylasqan ta'kirarlaw operatorlari' isletilingen. Birinshisi, sanni'n' yekilik ko'rinisin baspadan shi'g'ari'w procesin dawam yettiriw sha'rti boyi'nsha a'mel yetedi. Ishki jaylasqan, yekinshi ta'kirarlaw operatori'ndag'i' a'meller ha'r qanday, 0 den 15 ge shekem bolg'an sanlar to'rtew razryadli' yekilik san ko'rinisinde boli'wi'na tiykarlang'an. Wol jag'dayda kiritilgen sanni'n' ishki, yekilik ko'rinisinde u'shinshi razryadi'nda 0 yamasa 1 turg'anli'g'i' ani'qlanadi('son10& 8"). Sha'rt na'tiyjesi na'tiyje 1(ras) bolsa, ekrang'a '1', bolmasa '0' belgisi baspadan shi'g'ari'ladi'.

Keyingi qa'demde san razryadlari' shepke birge ji'lji'tiladi' ha'm tag'i' u'shinshi razryaddag'i' nomer baspadan shi'g'ari'ladi'. Ta'kirarlaw sanagish ma'nisi 0 bolaman degenge shekem dawam yetedi(to'rt ret) ha'm basqari'w ishki ta'kirarlaw operatori'nan shi'g'adi'.

while ta'kirarlaw operatori' ja'rdeminde na'tiyjeli programma kodi' jaziwga bir mi'sal bul-yeki natural sanlardi'n' yen' u'lken uli'wma bo'liwshisi(EKUB) Evklid algoritmi menen tabi'w ma'selesin ko'riwimiz mu'mkin:

```

int main()
{
    int a, b;
    cout<<'A ha'm B natural sanlar EKUBini tabi'w.\n';
    cout<<'A ha'm B natural sanlardi' kiritin:''
    cin>>a>>b;
    while(a!=b)
        a>b? a-=b:b-=a;
    cout<<"bul sanlar";
    cin<<" bul sanlar EKUBi="<<a;
    return 0;
}

```

Pu'tin tu'rdegi a ha'm b ma'nisleri' ag'i'mdan o'qillip balg'annan keyin wolar ma'nisleri' toki' wo'z-ara ten' bo'lmaguncha ta'kirarlaw procesi ju'z beredi. Ta'kirarlawdi'n' ha'r bir qa'deminde a ha'm b sanlari'ni'n' u'lkeninen-kishisi ayriladi ha'm wolar ten'ligi tekseriledi. Ta'kirarlawdan

keyingi ko'rsetpe jardeminde a wo'zgeriwshisinin' ma'nisi na'tiyje si'pati'nda baspadan shi'g'ari'ladi'.

do-while ta'kirlaw operatori: do-while ta'kirlaw operatori' while operatori'nan ayri'qsha tu'rde aldi'n operator yamasa blokdi' atqaradi', keyin ta'kirlaw sha'rtini tekseredi. Bul operator ta'kirlaw denesin keminde bir ret atqari'li'wi'n ta'miyinleydi. do-while ta'kirlaw operatori' to'mendegi sintaksisga iye:

```
do
    <operator yamasa blok>;
while(<an'latpa>) ;
```

Bunday ta'kirlaw operatori'ni'n' ken' qollani'latug'i'n jag'daylari' bul ta'kirlawdi' baslamasdan turi'p, ta'kirlaw sha'rtini tekseriwdin' i'laji' bolmag'an jag'daylar yesaplanadi'. Mi'sali', qandayda-bir processni dawam yettiriw yamasa toqtati'w haqqi'ndag'i' sorawg'a juwap ali'w ha'm woni' tekseriw za'ru'r bo'lsin. Ko'rinip turi'pti', wolda, processni baslamasdan aldi'n bul sorawdi' beriwdin' ma'nisi joq. Hesh bolmag'anda ta'kirlaw procesinin' bir qa'demi a'melge asi'ri'lg'an boli'wi' kerek:

```
#include <iostream.h>
int main()
{
    char juwap;
    do
    {
        ...// programma denesi
        cout<<' ' processni toqtap qali'wi'sh(N) :_'';
        cin>>javob;
    }while(juwap !=N)
    return 0;
}
```

Programma toki' " processti toqtap qali'wi'sh(N) :_"so'roviga(N) juwabi' kiritilgenge shekem dawam yetedi.

Bul operator da sheksiz ta'kirlani'wi' mu'mkin:

```
do;
while(1) ;
```

Ma'sele. Ha'r qanday 7 den u'lken pu'tin sandag'i' pul mug'dari'n 3 ha'm 5 sumli'qlarda beriw mu'mkinligi tasti'yi'qlansin. Qoyi'lg'an ma'sele $p=3n+5m$ ten'lemesi qanatlantiratug'i'n m, n sanlar jupli'qlari'n tabi'w ma'selesi boli'p tabi'ladi'(p-pul mug'dari'). Bul sha'rtting atqari'li'wi'n m ha'm n wo'zgeriwshilerinin' mu'mkin bolg'an ma'nisleri'ni'n' barli'q kombinatsiyalari'nda tekseriw za'ru'r boladi'.

```
#include <iostream.h>
```

```
int main()
```

```
{
```

```
    unsigned int pul; //pu1- kirgizetug'i'n pul mug'dari'
    unsigned n3, m5; //n-3 sumli'qlar, m-5 sumli'qlar sani'
    bool qate=false; //pu1 ma`nisin kiritilgendegi qa'telik
    do
```

```
{
```

```
    if(qa'te)
```

```
        cout<<'kiritilgan pul ma`nisi 7 den kishi !';
```

```
        qate=true ; //keyingi ta'kirarlaw qa'te yesaplanadi'
```

```
        cout<<'\\npul ma`nisin kiritin'(>7) :';
```

```
        cin>>pul;
```

```
    }while(pul<=7); //toki' 7 sani'nan u'lken san
```

```
kiritilgenshe
```

```
    n3=0 ; //qanday da 3 sumli'q joq
```

```
    do
```

```
{
```

```
        m5=0; // qandayda-bir da 5 sumli'q joq
```

```
        do
```

```
{
```

```
            if(3*n3+5*m5==pul)
```

```
                cout<<n3<<'ta 3 sumli'q + ' <<m5<<' ta 5
                sumli'q\\n';
```

```
                m5++ //5so'mliklar birge asi'ri'ladi'
```

```
        }while(3*n3+5*m5<=pul) ;
```

```
    n3++; //3 sumli'qlar birge asi'ri'ladi'
```

```
    }while(3*n3<=pul) ;
```

```
    return 0;
}
```

Programma pul ma'nisin kirgiziwdi so'raydi(pul wo'zgeriwshige). Yeger pul ma'nisi 7 sani'nan kishi bolsa, bul haqqi'nda xabar beriledi ha'm ta'kirar tu'rde ma'nis kirgiziw talap yetiledi. Pul ma'nisi 7 den u'lken bolg'anda, 3 ha'm 5 sumli'qlardi'n' mu'mkin bolg'an toli'q kombinatsiyasi'n a'melge asi'ri'w ushi'n ishpe-ish ta'kirarlawlar a'melge asi'ri'ladi'. Si'rtqi' ta'kirarlaw n^3 (3 sumli'qlar mug'dari') boyi'nsha, ishki ta'kirarlaw bolsa m^5 (5 sumli'qlar mug'dari') boyi'nsha, toki' bul mug'dardag'i' pul ma'nisi pul ma'nisinen asi'p ketmaguncha dawam yetedi. Ishki ta'kirarlawda m^5 wo'zgeriwshisnin' ha'r bir ma'nisinde « $3*n^3+5*m^5==pul$ » sha'rti tekseriledi, yeger wol wori'nli' bolsa, sheshim varianti' si'pati'nda n^3 ha'm m^5 wo'zgeriwshiler ma'nisleri' baspadan shi'g'ari'ladi'.

Pul ma'nisi 30 sum kiritilgende, ekrang'a 0 ta 3 sumli'q +6 ta 5 sumli'q baspadan shi'g'ari'ladi'.

5 dana 3 sumli'q +6 ta 5 sumli'q

10 ta 3 sumli'q +0 ta 5 sumli'q

sheshim variantlari' baspadan shi'g'ari'ladi'.

Soraw ha'm tapsi'rmalar

1. Ta'kirarlaw operatorlari' haqqi'nda tu'sinik.
2. For ta'kirarlaw operatori'.
3. Sheksiz ta'kirarlaw operatori'.
4. Sha'rtli ta'kirarlaw operatorlari'.
5. while operatori'n ani'qlama berin'.
6. Sheksiz ta'kirarlawg'a mi'sal keltiring.
7. do-while operatori'.
8. Ishpe-ish jaylasqan cikl operatorlari'na mi'sal keltiring.
9. break operatori'.
10. Qanday jag'daylarda break operatori'nan paydalani'ladi'?

The for Loop

You were introduced to a simple form of the **for** loop in Chapter 2. You might be surprised just how powerful and flexible the **for** loop is. Let's begin by reviewing the basics, starting with the most traditional forms of the **for** loop.

The general form of the **for** loop for repeating a single statement is

```
for(initialization; expression; increment) statement;
```

For repeating a block, the general form is

```
for(initialization; expression; increment)
{
    statement sequence
}
```

The *initialization* is usually an assignment statement that sets the initial value of the *loop control variable*, which acts as the counter that controls the loop. The *expression* is a conditional expression that determines whether or not the loop will repeat. The *increment* defines the amount by which the loop control variable will change each time the loop is repeated. Notice that these three major sections of the loop must be separated by semicolons. The **for** loop will continue to execute as long as the conditional expression tests true. Once the condition becomes false, the loop will exit, and program execution will resume on the statement following the **for** block.

The following program uses a **for** loop to print the square roots of the numbers between 1 and 99. Notice that in this example, the loop control variable is called **num**.

```
#include <iostream>
#include <cmath>
using namespace std;

int main()
{
    int num;
    double sq_root;

    for(num=1; num < 100; num++) {
        sq_root = sqrt((double) num);
        cout << num << " " << sq_root << '\n';
    }

    return 0;
}
```

The Infinite Loop

You can create an *infinite loop* (a loop that never terminates) by using this **for** construct:

```
for(;;)
{

    //...
}
```

This loop will run forever. Although there are some programming tasks, such as operating system command processors, that require an infinite loop, most "infinite loops" are really just loops with special termination requirements. Near the end of this chapter, you will see how to halt a loop of this type. (Hint: it's done using the **break** statement.)

The while Loop

Another loop is the **while**. The general form of the **while** loop is

```
while(expression) statement;
```

where *statement* may be a single statement or a block of statements. The *expression* defines the condition that controls the loop, and it may be any valid expression. The statement is performed while the condition is true. When the condition becomes false, program control passes to the line immediately following the loop.

The next program illustrates the **while** in a short but sometimes fascinating program. Virtually all computers support an extended character set beyond that defined by ASCII. The extended characters, if they exist, often include special characters such as foreign language symbols and scientific notations. The ASCII characters use values that are less than 128. The extended character set begins at 128 and continues to 255. This program prints all characters between 32 (which is a space) and 255. When you run this program, you will most likely see some very interesting characters.

The do-while Loop

Unlike the **for** and the **while** loops, in which the condition is tested at the top of the loop, the **do-while** loop checks its condition at the bottom of the loop. This means that a **do-while** loop will always execute at least once. The general form of the **do-while** loop is

```
do {  
    statements;  
} while(expression);
```

Although the braces are not necessary when only one statement is present, they are often used to improve readability of the **do-while** construct, thus preventing confusion with the **while**. The **do-while** loop executes as long as the conditional expression is true.

The following program loops until the number 100 is entered.

```
#include <iostream>  
using namespace std;  
  
int main()  
{  
    int num;  
  
    do {  
        cout << "Enter a number (100 to stop): ";  
        cin >> num;  
    } while(num != 100);  
  
    return 0;  
}
```

11-Lekciya: break, continue ha'mde goto operatorlari'

Joba:

1. break operatori'
2. continue operatori'
3. goto operatori'

Tayansh so'zler: *do-while ta'kirarlaw operatori', break operatori', continue operatori', goto operatori', ni'shanlar.*

break operatori': Ta'kirarlaw operatorlari'nin' atqari'li'wi'nda sonday jag'daylar ju'zege keliwi mu'mkin, wol jag'dayda qanday da bir qa'demde, ta'kirarlawdi' juwmag'i'na yetkazmasdan ta'kirarlawdan shi'g'i'w za'ru'rli'gi boli'wi' mu'mkin. Basqasha aytqanda ta'kirarlawdi' «uziw» kerek boli'wi' mu'mkin. Bunda break operatori'nan paydalani'ladi'. break operatori'n ta'kirarlaw operatori' denesinin' qa'legen(za'ru'r) jaylari'na qoyi'w arqali' sol wori'nlardan ta'kirarlawdan shi'g'i'wdi' a'melge asi'ri'w mu'mkin. Itibar beretug'i'n bolsaq switch-case operatori'nda da break operatori'n qo'llaw arqali' yerisilgen.

Ishpe-ish jaylasqan ta'kirarlaw ha'm switch operatorlari'nda break operatori' tek wo'zi jaylasqan blokdan shi'g'i'w mu'mkinshiligin beredi.

To'mendegi programmada yeki ishpe-ish jaylasqan ta'kirarlaw operatori'nan paydalang'an halda paydalani'wshi' ta'repinen kiritilgen qanday da sandi' 3 ha'm 7 sanlari'na sali'sti'rg'anda qanday arali'qqa tu'siwi ani'qlanadi'. Si'rtqi' ta'kirarlawda "san kiritin'(0-toqtap qali'w) :_" sorawi beriledi ha'm juwap juwap_san wo'zgeriwshisine woqi'ladi'. Yeger san nolden ayri'qsha bolsa, ishki ta'kirarlaw operatori'nda bul sannin' qanday da tu'siwi ani'qlani'p, sol haqqi'nda xabar beriledi ha'm ishki operatoridan shi'g'i'ladi'. Si'rtqi' ta'kirarlawdag'i sorawg'a juwap joli menende 0 kiritilse, programma wo'z jumi'si'n toqtatadi'.

```
#include <iostream.h>
```

```

int main()
{
    int juwap_san=0;
    do
    {
        while(juwap_san)
        {
            if(juwap_san<3)
            {
                cout<<"3 kishi !";
                break;
            }
            if(3<=juwap_san && juwap_san<=7)
            {
                cout<<"3 ha'm 7 araliqda!";
                break;
            }
            if(juwap_san>7)
            {
                cout<<"7 den u'lken !";
                break;
            }

        }
        cout<<"\nSan kiritin'(0-toqtap qali'w) :_";
        cin>>juwap_san;
    }
    while(juwap_san !=0) ;
    return 0;
}

```

A'meliyatda break operatoridan sheksiz ta'kirarlawdan shi'g'i'wda paydalani'ladi'.

```

for(; ;){
    // 1-sha'rt
    if(...)
    {
        ...
        break ;
    }
}

```

```

    }
    // 2- sha'rt
    if(...)
    {
        ...
        break;
    }
    ...
}

```

Bul mi'salda sheksiz for ta'kirarlawi'nan 1 yamasa 2-sha'rt wori'nlang'anda shi'g'i'ladi',

Ma'sele. Belgisiz pu'tin sanlar izbe-izligi 0 ma'nisi menen tawsi'ladi'. Bul jerde 0 izbe-izlik ag'zasi' yesaplanbaydi. Izbe-izlikti kamaymaydigan halda ta'rtilengen yamasa ta'rtilengengebengligin ani'qlansin.

```

#include <iostream>
using namespace std;
int main()
{
    unsigned int Ai_1=0, Ai;
    cout<<"sanlar izbe-izligin kiriting";
    cout<<"(0-tugash belgii) :\n";
    cin>>Ai; // izbe-izliktin' birinshi ag'zasi'
    while(Ai)
    {
        Ai_1=Ai;
        cin>>Ai; //na'wbettegi ag'za
        if(Ai_1>Ai)
            break;
    }
    if(Ai_1)
    {
        cout<<"izbe-izlik kamaymaydigan halda
ta'rtilengen";
        if(!Ai)
            cout<<" emas! ";
        else
            cout<<" ! ";
    }
}

```

```

else
    cout<<" izbe-izlik bos! ";
return 0;
}

```

Programma jumi'sqa tu'skende, basi'nda izbe-izliktin' birinshi ag'zasi' bo'lek woqi'p ali'nadi'(Ai wo'zgeriwshisine). Keyin Ai ma'nisi nolga ten' bo'lmag'ang'a shekem ta'kirarlaw operatori' a'mel yetedi. Ta'kirarlaw denesinde Ai ma'nisi aldi'ng'i' ma'nis si'pati'nda Ai_1 wo'zgeriwshisinde eslep qalinadi ha'm na'wbettegi ag'za Ai wo'zgeriwshisine woqi'ladi'. Yeger aldi'ng'i' ag'za na'wbettegi ag'zadan u'lken bolsa, break operatori' ja'rdeminde ta'kirarlaw procesi uziledi ha'm basqari'w ta'kirarlawdan keyingi sha'rt operatori'na wo'tedi. Bul jerdegi sha'rt operatorlari' mazmuni' to'mendegishe yeger Ai_1 nolden ayri'qsha bolsa, izbe-izliktin' keminde bir ag'zasi' kiritilgen boladi' (izbe-izlik bar) ha'm son'g'i' kiritilgen ag'za tekseriledi. Wo'z gezeginde yeger Ai nolden ayri'qsha bolsa, bul jag'day ag'zalar wortasi'nda kamaymaslik sha'rti atqari'lmag'anli'g'i' sebepli ag'zalardi kirgiziw procesi u'zilgenligin an'latadi' ha'm bul haqqi'nda xabar baspadan shi'g'ari'ladi'. Bolmasa izbe-izlikti kamaymay **digan** halda ta'r'tiplengen boladi'.

continue operatori': continue operatori' tap break operatorina u'qsas ta'kirarlaw operatori denesinin' wori'nlawin toqtatadi', lekin ta'kirarlawdan shi'g'i'p ketpesden keyingi qa'demine «sekirip» wo'tiwin tayi'nlaydi'.

continue operatori'n qollani'wi'na mi'sal si'pati'nda 2 ha'm 50 sanlar arali'g'indag'i' api'wayi' sanlardi' tabatug'i'n programma tekstin keltiremiz.

```

#include <iostream>
using namespace std;
int main()
{
    bool bolinedi=false;
    for(int i=2; i<50; i++)
    {
        for(int j=2; j<i/2; j++)
        {
            if(i%j)

```

```

        continue;
    else
    {
        bolinedi=true;
        break;
    }
}
// break wori'nlang'anda basqari'w wo'tetug'i'n jay
if(!bolinedi)
    cout <<i<<" ";
bulinadi=false;
}
return 0;
}

```

Keltirilgen programmada qoyil'g'an ma'sele ishpe-ish jaylasqan yeki ta'kirarlaw operatorlari' ja'rdeminde sheshilgen. Birinshi ta'kirarlaw operatori' 2 den 50 ge shekem sanlardi' payda yetiwge xi'zmet yetedi. Ishki ta'kirarlaw bolsa ha'r bir payda bolg'an sandi' 2 sani'nan baslap sol sanni'n' yari'mi'i'na shekem bolg'an sanlarg'a boli'p, qaldi'g'in tekseredi, yeger qaldi'q 0 sani'nan parqli bolsa, na'wbettegi sang'a boli'w dawam yetedi, bolmasa `bolinedi` wo'zgeriwshisine `true` ma'nis berip, ishki ta'kirarlaw uziledi(san wo'zinin' yari'mi'i'na shekem bolg'an qanday da sang'a bo'liner yeken, wol api'wayi' yemes ha'm keyingi sanlarg'a boli'p tekseriwge ha'jet joq). Ishki ta'kirarlawdan shi'qqannan keyin `bolinedi` ma'nisi `false` bolsa(!`bolinedi`), san api'wayi' boladi' ha'm wol baspadan shi'g'ari'ladi'.

goto operatori' ha'm ni'shanlar: Ni'shan - bul dawami'nda yeki noqat(':') qoyil'g'an identifikator. Ni'shan menen qanday da operator belgilenedi ha'm keyinirek, programmanin' basqa bir bo'legiden wog'an sha'rtsiz wo'tiw a'melge asi'ri'ladi'. Ni'shan menen ha'r qanday operator belgileniwi mu'mkin, atap aytqanda dag'aza operatori' ha'm bos operatori' da. Ni'shan tek funksiyalar ishinde a'mel yetedi.

Ni'shang'a sha'rtsiz wo'tiw `goto` operatori' ja'rdeminde atqari'ladi'. `goto` operatori' arqali' tek woni'n' wo'zi jaylasqan funksiya ishindegi

operatorlarg'a wo'tiw mu'mkin. goto operatori'ni'n' sintaksisi to'mendegishe:

```
goto <nishan>;
```

Ayi'ri'm jag'daylarda, goto operatori'ni'n' «sekirip wo'tiwi» yesabi'na qa'teler ju'zege keliwi mu'mkin. Mi'sali',

```
int i=0;
i++;
if(i)
    goto m;
```

```
int j;
m: j+=i;
```

operatorlaridin' atqari'li'wi' qa'telikke ali'p keledi, sebebi j wo'zgeriwshisnin' dag'aza yetiliwi qali'p ketedi.

Sha'rtsiz wo'tiw operatori' programmani du'ziwdegi ku'shli ha'm usi'ni'n' menen birge qa'wipli qurallardan biri yesaplanadi'. Ku'shliligi sonda, woni'n' ja'rdeminde algoritmdin' «basi berk» jaylari'nan shi'g'i'p ketiw mu'mkin. Yekinshi ta'repden, bloklardi'n' ishine wo'tiw, mi'sali' ta'kirarlaw operatorlari'n ishine «sekirip» kiriw kutilmegen jag'daylardi' ju'zege keltiriwi mu'mkin. Usi'ni'n' sebepinen, mu'mkinshiligi bari'nsha goto operatori'nan paydalanbaw kerek, isletilingen ta'g'dirde da tomendegi qag'i'ydag'a a'mel yetiw za'ru'r - blok ishine, if...else ha'm switch operatorlari' ishine, ha'mde ta'kirarlaw operatorlari' denesine si'rtdan kiriw mu'mkin yemes.

Sonday bolsada, ni'shan ja'rdeminde programmanin' qa'legen jayi'na wo'tiw mu'mkin sonda da, baslang'i'sh ma'nis beriw dag'azalari'nan sekrep wo'tiw qadag'an yetiledi, lekin bloklardan sekrep wo'tiw mu'mkin.

Mi'sali':

```
...
goto B; //qa'te
float x=0.0;
goto B; //tuwri'
{
```

```

    int n=10;
    x=n*x+x;
}
B: cout<<"x="<<x;
...

```

Atap aytqanda, ni'shan ja'rdeminde ishki blokdan si'rtqi' blokka ha'm si'rtqi' blokdan ishki blokka wo'tiwge C++ tili ruxsat beredi:

```

{
    ...
    goto ABC:
    ...
    {
        int i=15;
        ...
        ABC:
        ...
        goto XYZ;
        int y=10;
        ...
        goto KLM;
        ...
    }
    ...
    int k=0;
    ...
    KLM:
...}

```

Lekin, joqari'da keltirilgen mi'saldag'i' barli'q wo'tiwler mazmunan qa'te yesaplanadi'.

To'mendegi programmada yeki natural sanlar EKUBin tabi'w ma'selesidegi ta'kirarlaw procesin ni'shan ha'm goto operatori' jardeminde a'melge asi'ri'w ko'rsetilgen:

```

#include <iostream>
using namespace std;
int main()
{
    int a, b;

```

```

cout<<"A ha'm B natural sanlar EKUBini tabi'w.\n";
cout<<"A ha'm B natural sanlardi' kiritin': ";
cin>>a>>b;
nishan:
if(a==b)
{
    cout<<"Bul sanlar EKUBi="<<a;
    return 0;
}
a>b?a-=b:b-=a;
goto nishan;
}

```

Programmada ni'shan menen belgilengen operatorlarda a ha'm b sanlardi' tengligi tekseriledi. Yeger wolar ten' bolsa, qa'legen birewi, mi'sali' a sani' EKUB boladi' ha'm funksiyadan shi'g'i'ladi'. Bolmasa, bul sanlardi'n' u'lkeniden kishisi' ayriladi ha'm goto arqali' wolardi'n' tengligi tekseriledi. Ta'kirarlaw procesi a ha'm b sanlar wo'z-ara ten' bolaman degenge shekem dawam yetedi.

Soni' atap ko'rsetiw za'ru'r, bul ma'seleni ta'kirarlaw operatorlari' ja'rdeminde wori'nlaw na'tiyjeli yesaplanadi'.

Soraw ha'm tapsi'rmalar

1. *break* operatori'ni'n' qollani'li'wi'.
2. *continue* operatori'.
3. Ni'shan ne ha'm wol qanday dag'aza yetiledi?
4. Sha'rtsiz wo'tiw operatori'.
5. *continue* operatori'ni'n' wazi'ypasi'.
6. *break* ha'm *continue* operatorlari'ni'n' ayi'rmashi'li'g'i'.
7. *break* operatori'ni'n' qollani'li'wi'na tiyisli mi'sal keltiring.
8. Ni'shan ne?
9. Ni'shanlar dag'azasi'na tiyisli mi'sallar.
10. *continue* operatori'ni'n' atqari'li'wi'na tiyisli mi'sal.

Using continue

It is possible to force an early iteration of a loop, bypassing the loop's normal control structure. This is accomplished by using **continue**. The **continue** statement forces the next iteration of the loop to take place, skipping any code between itself and the conditional expression that controls the loop. For example, the following program prints the even numbers between 0 and 100:

```
#include <iostream>
using namespace std;

int main()
{
    int x;

    for(x=0; x<=100; x++) {
        if(x%2) continue;
        cout << x << ' ';
    }

    return 0;
}
```

Only even numbers are printed, because an odd one will cause the loop to iterate early, bypassing the **cout** statement.

In **while** and **do-while** loops, a **continue** statement will cause control to go directly to the conditional expression and then continue the looping process. In the case of the **for**, the increment part of the loop is performed, next the conditional expression is executed, and then the loop continues.

Using `break` to Exit Loops

It is possible to force an immediate exit from a loop, bypassing the loop's conditional test, by using the **`break`** statement. When the **`break`** statement is encountered inside a loop, the loop is immediately terminated, and program control resumes at the next statement following the loop. Here is a simple example:

```
#include <iostream>
using namespace std;

int main()
{
    int t;

    // Loops from 0 to 9, not to 100!
    for(t=0; t<100; t++) {
        if(t==10) break;
        cout << t << ' ';
    }

    return 0;
}
```

This program will print the numbers 0 through 9 on the screen before ending. It will not go to 100, because the **`break`** statement will cause it to terminate early.

The **`break`** statement is commonly used in loops in which a special condition can cause immediate termination. The following fragment contains an example of such a situation, where a keypress can stop the execution of the loop:

```
for(i=0; i<1000; i++) {
    // do something
    if(kbhit()) break;
}
```

A **`break`** will cause an exit from only the innermost loop. Here is an example:

```
#include <iostream>
using namespace std;
```

This program will print the numbers 1 through 9 on the screen 100 times. Each time the **`break`** is encountered, control is passed back to the outer **`for`** loop.

NOTE: A **`break`** used in a **`switch`** statement will affect only that **`switch`**, and not any loop the **`switch`** happens to be in.

Using the goto Statement

The **goto** statement fell out of favor with programmers many years ago because it encouraged the creation of "spaghetti code." However, it is still occasionally—and sometimes effectively—used. This book will not make a judgment regarding its validity as a form of program control. It should be stated, however, that there are no programming situations that require the use of the **goto** statement—it is not an item necessary for making the language complete. Rather, it is a convenience that, if used wisely, can be of benefit in certain programming situations. As such, the **goto** is not used in this book outside of this section. The chief concern most programmers have about the **goto** is its tendency to clutter a program and render it nearly unreadable. However, there are times when the use of the **goto** will actually clarify program flow rather than confuse it.

The **goto** requires a label for operation. A *label* is a valid C++ identifier followed by a colon. Furthermore, the label must be in the same function as the **goto** that uses it. For example, a loop from 1 to 100 could be written using a **goto** and a label, as shown here:

```
x = 1;
loop1:
    x++;
    if(x < 100) goto loop1;
```

One good use for the **goto** is to exit from a deeply nested routine. For example, consider the following code fragment:

```
for(...) {
    for(...) {
        while(...) {
            if(...) goto stop;
            *
            *
            *
        }
    }
}
stop:
    cout << "Error in program.\n";
```

Eliminating the **goto** would force a number of additional tests to be performed. A simple **break** statement would not work here, because it would only cause the program to exit from the innermost loop.

TIP: You should use the **goto** sparingly. But if your code would otherwise be much more difficult to read, or if execution speed of the code is critical, then by all means use the **goto**.

12-Lekciya: Funktsiyalar

Joba:

1. Funktsiya parametrlari ha'm argumentlari
2. Sha'rtlesiw boyi'nsha argumentler
3. Ko'rinis oblasti'. Lokal ha'm global wo'zgeriwshiler

Tayansh so'zler: *Funksiya parametrlari ha'm argumentlari, sha'rtlesiw boyi'nsha argumentler, ko'rinis oblasti'. lokal ha'm global wo'zgeriwshiler.*

Programma ta'miynati'ni jarati'w a'melde quramali' process yesaplanadi'. Programma du'ziwshi programma kompleksini bir pu'tinlikdegi ha'm woni'n' ha'r bir bo'leginin' ishki mazmuni'n ha'm wolardi'n' sezilmes parqlari'n yesapqa ali'wi' kerek boladi'.

Programmatalasti'ri'wg'a sistemali' yondoshuv sonnan ibarat, programma du'ziwshi aldi'na qoyil'g'an ma'sele aldi'nan yeki, ush ha'm wodan arti'q sali'sti'rg'anda kishi ma'sele asti'larg'a bo'linedi. Wo'z gezeginde bul ma'sele asti'lari' ha'm tag'i' kishi ma'sele asti'lari'na bolinwi mu'mkin.

Bul process mayda ma'selelerdi a'piwayi' standart a'meller ja'rdeminde sheshiw mu'mkin bolaman degenge shekem dawam yetedi. Sol jol menen ma'seleni dekompozitsiyalaw a'melge asi'ri'ladi'.

Yekinshi ta'repden, programmatalasti'ri'wda sonday jag'daylar gu'zetilip, wol jag'dayda programmanin' ha'r qi'yli' jaylari'nda mazmunan birdey algoritmlerdi wori'nlawg'a tuwri' keledi. Algoritm din' bul bo'lekleri tiykarg'i' yechilayotgan ma'seleden aji'rati'p ali'ng'an qanday da ma'sele asti'n sheshiwge mo'lsherlengen boli'p, jetkiliklishe wo'zbetinshe ma'nisge(na'tiyjege) iye yesaplanadi'. Mi'sal ushi'n to'mendegi ma'seleni ko'reylik:

Berilgen $a_0, a_1, \dots, a_{30}, b_0, b_1, \dots, b_{30}, c_0, c_1, \dots, c_{30}$ ha'm x, y, z haqi'yqi'y sanlar sheshiw

$$\frac{(a_0 x^{30} + a_1 x^{29} + \dots + a_{30})^2 - (b_0 y^{30} + b_1 y^{29} + \dots + b_{30})}{c_0 (x+z)^{30} + c_1 (x+z)^{29} + \dots + c_{30}}$$

an'latpani'n' ma'nisi yesaplansin.

Bul mi'saldi' sheshiwde bolshektin' surat ha'm bo'limindegi an'latpalar birdey algoritm menen yesaplanadi' ha'm programmada ha'r bir an'latpani'(ma'sele asti') yesaplaw ushi'n bul algoritmda 3 ret jazi'wg'a tuwri' keledi. Ma'seledegi 30-da'rejeli ko'p ag'zali'lardi' yesaplaw algoritmin, mi'sali', Gornor algoritmi bo'lek, bir nusqada jazi'p, wog'an ha'r qi'yli' parametrlar - bir sapar a vektor ha'm x ma'nisin, yekinshi sapar b vektor ha'm y ma'nisin ha'mde c vektor ha'm(x+z) ma'nisleri' menen murajat yetiw tiykarg'i' ma'seleni sheshiw mu'mkin boladi'. Funktsiyalar qollani'wi'ni'n' tag'i' bir sebebini to'mendegi ma'selede ko'riwimiz mu'mkin - berilgen si'ziqli' ten'lemeler sistemasin Gauss, Kramer, Zeydel usi'llari'nin' qandayda -biri menen sheshiw talap yetin'sin. Wol halda tiykarg'i' programmani to'mendegi bo'leklerge boli'w maqsetke muwapi'q bo'lar yedi: ten'leme ko'effitsentlarini kirgiziw bo'legi, sheshiw usi'li'n tan'law bo'legi, Gauss, Kramer, Zeydel usi'llari'n a'melge asi'ri'w ushi'n bo'lek bo'lekler, na'tiyjeni baspadan shi'g'ari'w bo'legi. Ha'r bir bo'lek ushi'n wo'z funksiya boli'p tabi'ladi' kompleksi jarati'p, za'ru'r bolg'anda wolarg'a bas funksiya denesinen murajatdi' a'melge asi'ri'w arqali' bas ma'sele sheshiw na'tiyjeli yesaplanadi'.

Bunday jag'daylarda programmani i'qsham ha'm na'tiyjeli yetiw ushi'n C++ tilinde programma bo'legini bo'lek aji'rati'p ali'p, woni' funksiya ko'rinishinde ani'qlaw mu'mkinshiligi bar.

Funksiya bul - C++ tilinde ma'sele sheshiwdegi gilt elementlerinen biri boli'p tabi'ladi'.

Funksiya parametrlari ha'm argumentleri: Programmada isletiletug'i'n ha'r qanday funksiya dag'aza yetiliwi kerek. A'detde funksiyalar dag'azasi' bas bet fayllarda dag'aza yetiledi ha'm #include direktivasi ja'rdeminde programma tekstine qosi'ladi'. Funksiya dag'azasi'n funksiya prototipi suwretlewleydi(ayi'ri'm jag'daylarda signatura dep ataladi'). Funksiya prototipi to'mendegi ko'rinishde boladi':

```
<qaytariwshi ma'nis tu'ri> <funksiya ati'>(<parametrler
dizimi>);
```

Bul jerde <qaytaruvchi ma'nis turi> - funksiya islewi na'tiyjesinde wol ta'repinen qaytaratug'in ma'nisni'n' tu'ri. Yeger qaytari'latug'in ma'nis tu'ri ko'rsatilmegen bolsa, sha'rtlesiw boyi'nsha funksiya qaytaratug'in ma'nis tu'ri int dep yesaplanadi, <parametrler dizimi> - u'tir menen aji'rati'lg'an funksiya parametrlerinin' tu'ri ha'm atlari' dizimi. Parametrlerdi jazbasa da boladi'. Dizim bos boli'wi' da mu'mkin. Funksiya prototiplariga mi'sallar:

```
int almassin(int, int);
double max(double x, double y);
void func();
void baspadan_shigariw(void);
```

Funksiya prototipi tu'sirip qaldi'riliwi mu'mkin, yeger programma tekstinde funksiya ani'qlani'wi' woni' shaqi'ratug'i'n funksiyalar tekstinen aldi'n jazi'lg'an bolsa. Lekin bul jag'day jaqsi' usi'l yesaplanilmaydi, a'sirese wo'z-ara bir-birine murajat yetiwshi funksiyalardi' ja'riyalawda ma'seleler ju'zege keliwi mu'mkin.

Funksiya ani'qlani'wi' - funksiya basi' ha'm figurali qawi'sqa('{', '}) ali'ng'an qanday da a'meliy mazmunga iye deneden ibarat boladi'. Yeger funksiya qaytaratug'in tu'ri void turinen ayri'qsha bolsa, woni'n' denesinde a'l'bette saykes tu'rdegi parametrge iye return operatori' boli'wi' sha'rt. Funksiya denesinde birden artiq return operatori' boli'wi' mu'mkin. Wolardi'n' qa'legen qandayda - biri'n wori'nlaw arqali' funksiyadan shi'g'i'p ketiledi. Yeger funksiyanin' ma'nisi programmada isletilbeytug'in bolsa, funksiyadan shi'g'i'w ushi'n parametrsiz return operatori' isletiliwi mu'mkin yamasa uluwma return isletilmeydi. Son'g'i' halda funksiyadan shi'g'i'w - son'g'i' jawi'li'wshi' qawi'sg'a jetip kelip ju'z beredi.

Funksiya programmanin' qandayda - bir modulinda birden-bir tu'rde ani'qlani'wi' kerek, woni'n' dag'azasi' bolsa funksiyanin' isletetug'in

modullarda neshe ret jazi'li'wi' mu'mkin. Funksiya ani'qlani'wi'nda bas betdegi barli'q parametrlar atlari' jazi'li'wi' sha'rt.

A'detde programmada funksiya ma'lum bir jumi'sti' a'melge asi'ri'w ushi'n shaqi'ri'ladi'. Funksiyag'a murajat yetkende, wol qoyi'lg'an ma'seleni sheshedi ha'm wo'z jumi'si'n tamamlawi'nda qanday da ma'nisti na'tiyje si'pati'nda qaytaradi'

Funksiyani' shaqi'ri'w ushi'n woni'n' ati' ha'm wodan keyin qawi's ishinde argumentlar dizimi beriledi:

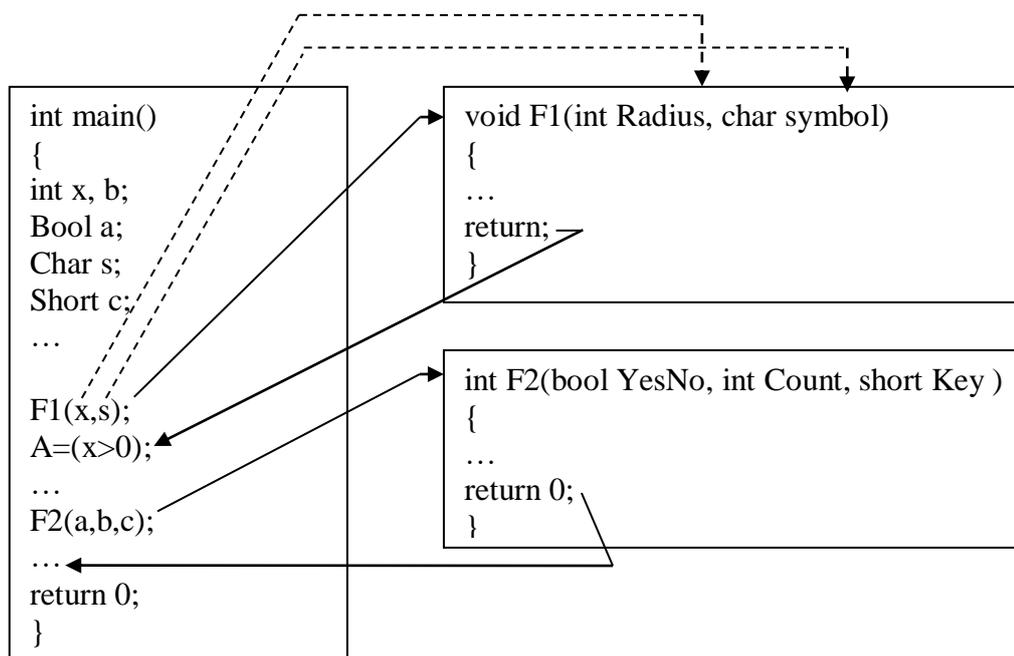
```
<funksiya at'>(<argument1>, <argument2>, ..., <argumentn>);
```

Bul jerde ha'r bir <argument> - funksiya denesine uzati'latug'i'n ha'm keyinirek yesaplaw processinde isletiletug'i'n wo'zgeriwshi, an'latpa yamasa turaqli' boli'p tabi'ladi'. Argumentlar dizimi bos boli'wi' mu'mkin.

Funksiyalar da wo'z denesinde basqa funksiylardi', wo'zin da shaqi'ri'wi' mu'mkin. Wo'z denesinde wo'zin shaqi'ratug'i'n funksiylardi' rekursiv funksiylar dep ataladi'.

Aldi'ng'i' lekciyalarda ayti'p wo'tilgenidey, C++ tilindegi ha'r qanqay programmada a'l'bette main() bas funksiya' boli'wi' kerek. A'yne sol funksiya'ni ju'klegish ta'repinen shaqi'riliwi' menen programma atqari'li'wi' baslanadi'.

To'mendegi suwretde bas funksiya'dan basqa funksiylardi' shaqi'ri'w ha'm wolardan qayti'w sxemasi' ko'rsetilgen.



1-su'wret. Bas funksiyadan basqa funksiyalardi' shaqi'ri'w ha'm qayti'w.

Programma `main()` funksiyasi'n wori'nlawdan baslanadi' ha'm "`f1(x, y);`"- funksiya shaqi'ri'wi'na shekem dawam yetedi ha'm keyinirek basqari'w `f1(x, y)` funksiya denesindegi a'mellerdi wori'nlawg'a wo'tedi. Bunda Radius parametrinin' ma'nisi si'pati'nda funksiya `x` wo'zgeriwshi ma'nisin, Symbol parametri si'pati'nda `y` wo'zgeriwshisinin' ma'nisi isletiledi. Funksiya denesi `return` operatori'na shekem atqari'ladi'. `Return` operatori' basqari'wdi' `main()` funksiyasi' denesindegi `f1()` funksiyasi' shaqi'ri'lg'an operatoridan keyingi operatorg'a wo'tiwdi ta'miyinleydi, yag'ni'y funksiyadan qayti'w ju'z beredi. Sonnan keyin `main()` funksiyasi' operatorlari' atqari'li'wda dawam yetedi ha'm "`f2(a, b, c);`" - funksiya shaqi'ri'wi' arqali' basqari'w `f2()` funksiya denesine wo'tedi ha'm yesaplaw processinde saykes tu'rde `YesNo` si'pati'nda `a` wo'zgeriwshisinin', `cout` si'pati'nda wo'zgeriwshisinin' ha'm key si'pati'nda `c` wo'zgerchuvchisinin' ma'nisleri' isletiledi. Funksiya denesindegi `return` operatori' yamasa son'g'i' operator atqarg'annan keyin avtomatik tu'rde bas funksiyag'a qayti'w a'melge asi'ri'ladi'.

Ko'binese `main()` funksiyasi'ni'n' parametrlar dizimi bos boladi'. Yeger programmani jumi'sqa tu'siriwde, buyri'q qatari'nda ma'lum bir parametrlar uzati'w(beriw) za'ru'r bolsa, `main()` funksiyasi'ni'n' sintaksisi wo'zgeredi:

```
int main(int argc, char* argv[]);
```

Bul jerde `argc`-uzati'latug'i'n parametrlar sani', `argv[]` - bir-birinen punktuatsiya belgileri(ha'm probel) menen aji'rati'lg'an parametrlar dizimin wo'z ishine alg'an massivke ko'rsetkish.

To'mende funksiyalardi' ja'riyalaw, shaqi'ri'w ha'm ani'qlawg'a mi'sallar keltirilgen:

```
// funksiyalar dag'azasi'  
int menin_funksiyam(int Number, float Point) ;  
char Belgini_oqiw() ;  
void bitni_ornatiw(short Num) ;  
void Amel_joq(int, char) ;_
```

```

// funksiyalardi' shaqi'ri'w
Result = menin_funksiyam(Varb1, 3.14) ;
Symb = Belgini_oqiw() ;
bitni_ornatiw(3) ;
Amel_joq(2, Smb1) ;
// funksiyalardi' ani'qlaw
int menin_funksiyam(int Number, float Point) ;
{
    int x;
    ...
    return x;
}
char Belgini_oqiw() ;
{
    char Symbo1;
    cin>>Symbo1;
    return Symbo1;
};
void bitni_ornatiw(short number)
{
    global_bit=global_bit|number;
};
void Amel_joq(int x, char ch)
{
};

```

Funksiyani'n' programmadag'i' worni'n ha'm de tu'sinikli boli'wi' ushi'n san kvadrati'n yesaplaw ma'selesinde funksiyadan paydalani'wdi' ko'reylik.

Funksiya prototipin fayl.h fayli'nda jaylasti'rami'z:

```

long Son_Kvadrati(int) ;

```

Tiykarg'i' programmag'a usi' fayldi qosi'w arqali' Son_Kvadrati() funksiya dag'azasi' programma tekstine kiritiledi:

```

#include <iostream.h>

```

```

#include "fayl.h"

```

```

int main()

```

```

{

```

```

    int Wozgeriwshi=5;
    cout<<Son_kvadrati(Wozgeriwshi);
    return 0;
}

```

Son_kvadrati funksiyasin wo'z aldina "fayl.h" ati' menen saqlap ali'nadi'.

```

long Son_kvadrati(int x)
{
    return x*x;
}

```

Tap sol ma'seleni .h fayli'nan paydalanmag'an halda, funksiya dag'azasi'n programma tekstine jazi'w arqali' da sheshiw mu'mkin:

```

#include <iostream.h>
long Son_kvadrati(int);
int main()
{
    int Wozgeriwshi=5;
    cout<<Son_kvadrati(Wozgeriwshi) ;
    return 0;
}
long Son_kvadrati(int x)
{
    return x*x;
}

```

Programma islewinde wo'zgeriw bolmaydi' ha'm na'tiyje si'pati'nda ekrang'a 25 sani'n baspadan shi'g'aradi'.

Ma'sele. Yeki api'wayi' san «egizek» dep ataladi', yeger bir-birinen ayi'rmashi'li'g'i' 2 ten' bolsa (mi'sali', 41 ha'm 43 sanlari'). Berilgen natural n ushi'n $[n..2n]$ kesmadagi barli'q «egizek» sanlar jupli'qlari' baspag'a shig'arilsin. Ma'seleni sheshiw ushi'n berilgen k sani'n api'wayi' san yamasa joq yekenligi ani'qlawshi' logikali'q ma'nis qaytaratug'i'n funksiya du'ziw za'ru'r boladi'. Funksiyada k sani' $2..k/2$ ge shekem sanlarg'a bo'linedi, yeger k bul sanlardi'n' qandayda-biri'ne da bo'lindisa, wol api'wayi' san yesaplanadi' ha'm funksiya true ma'nisin qaytaradi'. Bas

funksiyada, berilgen n ushi'n $[n, 2n]$ aralig'indag'i' $(n, n+2)$, $(n+1, n+3)$, .., $(2n-2, 2n)$ san jupli'qlari'n api'wayi' sanlar yekenligi tekseriledi ha'm sha'rt qa'naatlanti'rg'an jupli'qlar baspadan shi'g'ari'ladi'.

Programma teksti:

```
#include <iostream>
using namespace std;
bool TubSan(unsigned long k) ;
int main()
{
    unsigned long n, i;
    unsigned char yegiz=0;
    cout<<"n->";
    cin>>n;
    cout<<'['<<n<<".."<<2*n<<']';
    cout<<"aralig'indag'i' yegiz api'wayi' sanlar:\n";
    for(i=n; i<=2*n-2; i++)
        if(TubSan(i) && TubSan(i+2) )
            {
                cout<<"{"<<i<<", "<<i+2<<"}; ";
            };
    return 0;
}
bool TubSan(unsigned long k)
{
    unsigned long m;
    for(m=2; m<=k/2; m++)
        if(k%m==0)
            return false;
    return true;
}
```

Natural n sani' ushi'n 100 kiritilse, programma to'mendegi sanlar jupli'qlari'n baspadan shi'g'aradi':

```
[100..200] aralig'indag'i' yegiz api'wayi' sanlar:
{101, 103}; {107, 109}; {137, 139}; {149, 151}; {179, 181};
{191, 193}; {197, 199};
```

Ko'rinis oblasti'. Lokal ha'm global wo'zgeriwshiler: Wo'zgeriwshiler funksiya denesinde yamasa wodan si'rtda dag'aza yetiliwi mu'mkin. Funksiya ishinde dag'aza yetilgen wo'zgeriwshilerga lokal wo'zgeruvchlar dep ataladi'. Bunday wo'zgeriwshiler yaddag'i' programma stekida joylasadi' ha'm tek wo'zi dag'aza yetilgen funksiya denesinde a'mel yetedi. Basqari'w tiykarg'i' funksiyag'a qayti'wi' menen lokal wo'zgeriwshiler ushi'n aji'rati'lg'an yad bosatiladi (wo'shiriledi).

Ha'r bir wo'zgeriwshi wo'zinin' a'mel yetiw oblasti' ha'm jasaw waqi'ti' qa'siyetleri menen xarakterlenedi.

Wo'zgeriwshi a'mel yetiw oblasti' degende wo'zgeriwshinin' isletiw mu'mkin bolg'an programma oblasti' (bo'legi) tu'siniledi. Bul tu'sinik menen wo'zgeriwshinin' ko'rinis oblasti' aji'ralmas baylani'sqan. Wo'zgeriwshi a'mel yetiw denesinen shi'qqanda ko'rinbey qaladi'. Yekinshi ta'repden, wo'zgeriwshi a'mel yetiw denesinde boli'wi', lekin ko'rinbesligi mu'mkin. Bunda ko'rinis oblasti'na ruxsat beriw a'meli «::» ja'rdeminde ko'rinbes wo'zgeriwshige murajat yetiw mu'mkin boladi'.

Wo'zgeriwshinin' jasaw waqi'ti' dep, wol bar bolg'an programma bo'leginin' atqari'li'wi'na ketken waqi't intervali'na ayti'ladi'.

Lokal wo'zgeriwshiler wo'zleri dag'aza yetilgen funksiya yamasa blok shegarasi'nda ko'rinis oblasti'na iye. Blokdag'i' ishki bloklarda tap sol atdag'i' wo'zgeriwshi dag'aza yetilgen bolsa, ishki bloklarda bul lokal wo'zgeriwshi da a'mel yetpey qaladi'. Lokal wo'zgeriwshi jasaw waqi'ti'-blok yamasa funksiyani' wori'nlaw waqi'ti' menen ani'qlanadi'. Bul jag'day soni' an'latadi', ha'r qi'yli' funksiyalarda bir-birine uluwma baylani'sli' bolmag'an birdey atdag'i' lokal wo'zgeriwshilerdi isletiw mu'mkin.

To'mendegi programmada main() ha'm sum() funksiyalari'nda birdey atdag'i' wo'zgeriwshilerdi isletiw ko'rsetilgen. Programmada yeki sannin' ji'yi'ndi'si' yesaplanadi' ha'm baspadan shi'g'ari'ladi':

```
#include <iostream>
using namespace std;
// funksiya prototipi
int sum(int a, int b);
int main()
{
```

```

        // lokal wo'zgeriwshiler
        int x=2, y=3;
        cout<<sum(x, y) ;
        return 0;
    }
    int sum(int a, int b)
    {
        // lokal wo'zgeriwshi
        int x=a+b;
        return x;
    }

```

Global wo'zgeriwshiler programma tekstinde funksiya ani'qlani'wi'nan si'rtida dag'aza yetiledi ha'm dag'aza yetilgen jayi'nan baslap programma aqi'ri'i'na shekem a'mel yetedi:

```

#include <iostream>
using namespace std;
int f1() ;
int f2() ;
int main()
{
    cout<<f1() <<" "<<f2() <<endl;
    return 0;
}
int f1()
{
    return x;// kompilyatsiya qa'tesi ju'z beredi
}
int x=10; //global wo'zgeriwshi dag'azasi'
int f2()
{
    return x*x;
}

```

Joqari'da keltirilgen programmada kompilyatsiya qa'tesi ju'z beredi, sebebi f1() funksiya ushi'n x wo'zgeriwshisi belgisiz yesaplanadi'.

Programma tekstinde global wo'zgeriwshilerdi wolar dag'azasi'nan keyin jazi'lg'an qa'legen funksiyadan isletiw mu'mkin. Usi'ni'n' sebebini, global wo'zgeriwshiler programma tekstinin' basi'nda jazi'ladi'. Funksiya ishinen

global wo'zgeriwshige murajat yetiw ushi'n funksiyada woni'n' ati' menen saykes tu'setug'i'n lokal wo'zgeriwshiler bolmaytug'i'ni' kerek. Yeger global wo'zgeriwshi dag'azasi'nda wog'an baslang'i'sh ma'nisi berilmagan bolsa, wolardi'n' ma'nisi 0 yesaplanadi'. Global wo'zgeriwshinin' a'mel yetiw oblasti' woni'n' ko'rinish oblasti' menen ustma-u'st tu'sedi.

Soni' atap ko'rsetiw za'ru'r, programma du'ziwshiler mu'mkinshiligi bari'nsha global wo'zgeriwshilerdi isletmaslikka ha'reket yetediler, sebebi bunday wo'zgeriwshiler ma'nisin programmanin' qa'legen jayi'ndan wo'zgeritiw mu'mkinshiligi bar ekenligi sebepli programma islewinde mazmunan qa'teler ju'zege keliwi mu'mkin. Bul pikrimizdi tastiyoqlaytug'i'n programmani ko'reylik.

```
#include <iostream>
using namespace std;
//global wo'zgeriwshi dag'azasi'
int test=100;
void Baspadan_shigariw(void);
int main()
{
    //lokal wo'zgeriwshi dag'azasi'
    int test=10;
    //global wo'zgeriwshi baspadan shi'g'ari'w funksiyasi'n
    shaqi'ri'w
    Baspadan_shigariw() ;
    cout<<"Lokal wo'zgeriwshi: "<<test<<"\n";
    return 0;
}
void Baspadan_shigariw(void)
{
    cout<<"global wo'zgeriwshi: "<<test<<"\n";
}
```

Programma basi'nda test global wo'zgeriwshisi 100 ma'nisi menen dag'aza yetiledi. Keyinirek, main() funksiyasi'nda test ati' menen lokal wo'zgeriwshisi 10 ma'nisi menen dag'aza yetiledi. Programmada, Baspadan_shigariw() funksiyasi'na murajat yetilgeninde, tiykarg'i' funksiya denesinen waqti'nsha shi'g'i'ladi' ha'm na'tiyjede main() funksiyasi'nda dag'aza yetilgen barli'q lokal wo'zgeriwshilerga murajat yetiw mu'mkin bo'lmay qaladi'. Usi'ni'n' sebebinen Baspadan_shigariw() funksiyasi'nda global test wo'zgeriwshisinin' ma'nisin

baspadan shi'g'ari'ladi'. Tiykarg'i' programmaga qaytilgandan keyin, main() funksiyasi'ndag'i' lokal test wo'zgeriwshisi global test wo'zgeriwshisin «jasi'ri'di'» ha'm lokal test wo'zgeriwshin ma'nisi baspadan shi'g'ari'ladi'. Programma islewi na'tiyjesinde ekrang'a to'mendegi na'tiyjeler baspadan shi'g'ari'ladi':

Global wo'zgeriwshi: 100

Lokal wo'zgeriwshi: 10

Soraw ha'm tapsi'rmalar

1. *Funksiya prototipi ne?*
2. *Funksiya parametrleri.*
3. *Funksiya argumentleri.*
4. *void tu'ri.*
5. *Statikali'q wo'zgeriwshilerge tu'sindirme berin'.*
6. *Lokal ha'm global wo'zgeriwshiler, wolar wortasi'ndag'i' parq.*
7. *Rekursiv funksiyalar.*
8. *Wo'zgeriwshinin' a'mel yetiw oblasti'.*
9. *Wo'zgeriwshinin' jasaw waqi'ti'.*
10. *Wo'zgeriwshinin' ko'rinis oblasti'.*

Local Variables

As you know, variables that are declared inside a function are called *local variables*. However, C++ supports a more subtle concept of the local variable than you have previously seen. In C++, variables can be localized to a block. That is, a variable can be declared inside any block of code and is then local to it. (Remember, a block begins with an opening curly brace and ends with a closing curly brace.) In reality, variables local to a function are simply a special case of the more general concept.

A local variable can be used only by statements located within the block in which it is declared. Stated another way, a local variable is not known outside its own code block. Thus, statements outside a block cannot access an object defined within the block.

One of the most important things to understand about local variables is that they exist only while the block of code in which they are declared is executing. This means that a local variable is created upon entry into its block and destroyed upon exit. Because a local variable is destroyed upon exit from its block, its value is lost.

Formal Parameters

As you know, if a function uses arguments, then it must declare variables that will accept the values of those arguments. These variables are called the *formal parameters* of the function. Aside from receiving the arguments when a function is called, formal parameters behave like any other local variables inside the function. The scope of a parameter is local to its function.

You must make sure that the formal parameters you declare are of the same type as the arguments you will pass to the function. Also, even though these variables perform the special task of receiving the values of the arguments, they can be used like any other local variable. For example, you can assign a new value to a parameter within the function.

Global variables are, in many ways, the opposite of local variables. They are known throughout the entire program, can be used by any piece of code, and maintain their values during the entire execution of the program. Therefore, their scope extends to the entire program. You can create global variables by declaring them outside of any function. Because they are global, they can be accessed by any expression, regardless of the function in which the expression is located.

When a global and a local variable share the same name, the local variable has precedence. Put differently, a local variable will hide a global variable of the same name. Thus, even though global variables can be accessed by any code in your program, this will happen only when no local variable's name overrides the global variable.

Function Prototypes

Until this point, prototypes have been used without explanation in the sample programs. Now it is time to explain them formally. In C++, all functions must be declared before they are used. Typically, this is accomplished by use of a function prototype. Prototypes specify three things about a function:

- ◆ Its return type
- ◆ The type of its parameters
- ◆ The number of its parameters

The general form of a function prototype is as follows. It is the same as a function definition, except that no body is present.

```
type func-name(type parm_name1, type parm_name2,...,  
               type parm_nameN);
```

The use of parameter names in a prototype is optional. However, their use does let the compiler identify any type mismatches by name when an error occurs, so it is a good idea to include them.

13-Lekciya: C++ tili funksiyalari'ndag'i' mu'mkinshilikler

Joba:

1. :: a'meli
2. Yad klasslari'
3. Atlar ken'isligi
4. Jaylasti'ri'latug'i'n (inline) funksiyalar

Tayansh so'zler: :: *a'meli, xotira klasslari', avtomat wo'zgeriwshiler, registr wo'zgeriwshiler, si'rtqi' wo'zgeriwshiler, statik wo'zgeriwshiler, volatile klasi' wo'zgeriwshileri, atlar ken'isligi, jaylasti'ri'latug'i'n(inline) funksiyalar.*

::a'meli: Joqari'da ayti'p o'tilgen si'yaqli', lokal wo'zgeriwshi dag'azasi' tap sol atdag'i' global wo'zgeriwshini «jasi'rad'i» ha'm bul wori'ndan global wo'zgeriwshige murajat yetiw mu'mkinshiligi bo'lmay qaladi'. C++ tilinde bunday jag'daylarda da global wo'zgeriwshige murajat yetiw mu'mkinshiligi saqlani'p qali'ng'an. Woni'n' ushi'n «ko'riniw oblasti'na ruxsat beriw» a'melinen paydalani'w mu'mkin ha'm wo'zgeriwshi aldi'na yeki noqat ("::") qoyi'w za'ru'r boladi'. Mi'sal si'pati'nda to'mendegi programani keltiremiz:

```
#include <iostream>
using namespace std;
// global wo'zgeriwshi dag'azasi'
int uzg=5;
int main()
{
    //lokal wo'zgeriwshi dag'azasi'
    int uzg=70;
    //lokal wo'zgeriwshin baspadan shi'g'ari'w
    cout<<uzg<<"\n";
    //global wo'zgeriwshin baspadan shi'g'ari'w
    cout<<::uzg<<"\n";
    return 0;
}
```

Programma islewi na'tiyjesinde ekrang'a aldi'n 70 ha'm keyin 5 sanlari' baspadan shi'g'ari'ladi'.

Yad klasslari': Wo'zgeriwshilerdin' ko'rinishi oblasti' ha'm a'mel yetiw waqti'n aniqlawshi' wo'zgeriwshi modifikatorlari bar (5.1-keste).

5.1-keste. Wo'zgeriwshi modifikatorlari

Modifikator	Qollani'li'wi'	A'mel yetiw oblasti'	Jasaw da'wiri
auto	lokal	blok	waqti'nsha
register	lokal	blok	waqti'nsha
extern	global	blok	waqti'nsha
static	lokal	blok	daimiy
	global	fayl	daimiy
volatile	global	fayl	daimiy

Avtomat wo'zgeriwshiler. auto modifikatori lokal wo'zgeriwshiler dag'azasi'nda isletiledi. A'detde lokal wo'zgeriwshiler dag'azasi'nda bul modifikator shartlesiw boyi'nsha qollani'ladi' ha'm usi'ni'n' sebebini a'melde woni' jazi'wmaydi'.

```
#include <iostream>
using namespace std;
int main()
{
    auto int x=2;    //int x=2; menen ekvivalent
    cout<<x;
    return 0;
}
```

auto modifikatori blok ishinde dag'aza yetilgen lokal wo'zgeriwshilerga qollani'ladi'. Bul wo'zgeriwshiler blokdan shi'g'i'wi' menen avtomatik tu'rde joq boli'p ketadi.

Registr wo'zgeriwshiler. Register modifikatori kompilyatorga, ko'rsetilgen wo'zgeriwshin protsessor registrlariga jaylasti'ri'wg'a ha'reket yetiwdi tayi'nlaydi'. Yeger bul ha'reket na'tiyje bermasa wo'zgeriwshi auto tu'rindagi lokal wo'zgeriwshi si'pati'nda a'mel yetedi.

Wo'zgeriwshilerdi registrlarda jaylasti'ri'w programma kodi'n wori'nlaw tezligi boyi'nsha optimallasti'radi', sebebi protsessor yaddag'i' berilgenlerge sali'sti'rg'anda registrdagi ma'nisler menen talay tez isleydi. Lekin registrar sani' sheklengenligi ushi'n daimiy wo'zgeriwshilerdide registrlarda jaylasti'ri'w di'n' i'laji' bolmaydi'.

```
#include <iostream.h>
int main()
{
    register int Reg;
    ...
    return 0;
}
```

register modifikatori tek lokal wo'zgeriwshilerine sali'sti'rg'anda qollani'ladi', global wo'zgeriwshilerga qo'llaw kompilyatsiya qa'tesinin'ge ali'p keledi.

Si'rtqi' wo'zgeriwshiler. Yeger programma bir neshe moduldan ibarat bolsa, wolar qanday da wo'zgeriwshi arqali' wo'z-ara ma'nis almasi'wlari' mu'mkin (fayllar arasi'nda). Woni'n' ushi'n wo'zgeriwshi qandayda-bir modulda global ta'rzde dag'aza yetiledi ha'm wol basqa faylda (modulda) ko'riniwi ushi'n wonda extern modifikatori menen dag'aza yetiliwi kerek boladi'. extern modifikatori wo'zgeriwshinin' basqa faylda dag'aza yetilgenligin an'latadi'. Si'rtqi' wo'zgeriwshiler isletilingen programmani ko'reylik.

```
//Bas bet.h fayli'
void Bayraq_Almassin(void);
//modul_1.cpp fayli'nda
bool bayraq;
void Bayraq_Almassin(void)
{
    Bayraq=!Bayraq;
}
// ma'sele.cpp fayli'nda
#include <iostream.h>
#include <bet.h>
#include <modul_1.cpp>
extern bool Bayraq;
```

```

int main()
{
    Bayraq_Almassin() ;
    if(Bayraq)
        cout<<"Bayraq TRUE"<<endl;
    else
        cout<<"Bayraq FALSE"<<endl;
    return 0;
}

```

Aldi'n bas bet.h fayli'nda Bayraq_Almassin() funksiya basi' dag'aza yetiledi, keyin modul_1.cpp fayli'nda si'rtqi' wo'zgeriwshi dag'aza yetiledi ha'm Bayraq_Almassin() funksiyasi'ni'n' denesi ani'qlanadi' ha'm aqi'r-aqi'betde, ma'sele.cpp fayli'nda Bayraq wo'zgeriwshisi si'rtqi' dep dag'aza yetiledi.

Statikali'q wo'zgeriwshiler. Statikali'q wo'zgeriwshiler `static` modifikatori menen dag'aza yetiledi ha'm wo'z ayri'qshali'qi'na ko're global wo'zgeriwshilerga qusaydi'. Yeger bul tu'rdegi wo'zgeriwshi global bolsa, woni'n' a'mel yetiw oblasti' dag'aza yetilgen wori'ndan tap fayl aqi'ri'i'na shekem boladi'. Yeger statikali'q wo'zgeriwshi funksiya yamasa blok ishinde dag'aza yetiletug'i'n' bolsa, wol funksiya yamasa blokka birinshi kiritilgan initsializatsiya yetiledi. Wo'zgeriwshinin' bul ma'nisi funksiya keyingi shaqi'rilganida yamasa blokka qayta kiritilgan saqlani'p qaladi' ha'm bul ma'nisti wo'zgertiw mu'mkin. Statikali'q wo'zgeriwshilerdi si'rtqi' dep ja'riyalay otirip bolmaydi'.

Yeger statik wo'zgeriwshi initsializatsiya yetilmegen bolsa, woni'n' birinshi murajatdagi ma'nisi 0 yesaplanadi'.

Mi'sal si'pati'nda qandayda-bir funksiyani' neshe ma'rte shaqi'rilganligini ani'qlaw ma'selesin ko'reylik:

```

#include <iostream>
using namespace std;
int Sanagich(void) ;
int main()
{
    int na'tiyje;
    for(int i=0; i<10; i++)

```

```

        Na'tiyje=sanagich() ;
        cout<<natija;
        return 0;
    }
    int Sanagich(void)
    {
        static short sanagich=0;
        ...
        Sanagich++;
        return sanagich;
    }

```

Bul jerde tiykarg'i' funksiyadan counter statik wo'zgeriwshige iye sanagich() funksiyasi' 30 ret shaqi'ri'ladi'. Funksiya birinshi ret shaqi'rilganda sanagich statikali'q wo'zgeriwshige 0 ma'nisin qabi'l yetedi ha'm woni'n' ma'nisi birge artqan halda funksiya ma'nisi si'pati'nda qaytari'ladi'. Statikali'q wo'zgeriwshiler ma'nisleri'n funksiyani bir shaqi'rilishidan yekinshisine saqlani'p qali'ni'wi sebepli, keyingi ha'r bir murajatlarda sanagich ma'nisi birge arti'p baradi'.

Ma'sele. Berilgen pu'tin sannin' barli'q api'wayi' bo'liwshileri ani'qlansin. Ma'seleni sheshiw algoritmi to'mendegi ta'kirarlani'wshi' processdan ibarat boladi': berilgen san api'wayi' sang'a (birinshi qa'demde 2 ge) bo'linedi. Yeger qaldi'q 0 bolsa, api'wayi' san baspadan shi'g'ari'ladi' ha'm bo'liniwshi si'pati'nda bo'lindi ali'nadi', bolmasa na'wbettegi api'wayi' san ali'nadi'. Ta'kirarlaw na'wbettegi api'wayi' san bo'liniwshine ten' bolaman degenge shekem dawam yetedi.

Programma teksti:

```

#include <iostream>
#include <math.h>
using namespace std;
int Navb_tub() ;
int main()
{
    unsigned int n, p;
    cout<<"\n n ma'nisin kiritin': ";
    cin>>n;
    cout<<"\n 1";

```

```

    p=Navb_tub() ;
    while(n>=p)
    {
        if(n%p==0)
        {
            cout<<"*"<<p;
            n=n/p;
        }
        else p=Navb_tub() ;
    }
    return 0;
}
int Navb_tub()
{
    static unsigned int tub=1;
    for(;;)
    {
        tub++;
        short int ha_tub=1;
        for(int i=2; i<=tub/2; i++)
            if(tub%i==0)
                ha_tub=0;
        if(ha_tub)
            return tub;
    }
    return 0;
}

```

Programmada na'wbettegi api'wayi' sandi' payda yetiw funksiya ko'riniside a'melge asi'ri'lg'an. Navb_tub() funksiyasi'ni'n' ha'r shaqi'rilishida son'g'i' api'wayi' sandan keyingi api'wayi' san tabi'ladi'. Son'g'i' api'wayi' sandi' <<eslep>> qali'w ushi'n api'wayi' wo'zgeriwshisi statik yetip ani'qlang'an.

Programma jumi'sqa tu'skenda klaviaturadan n wo'zgeriwshisinin' ma'nisi si'pati'nda 60 sani' kiritilse, ekrang'a to'mendegi ko'beytpe baspadan shi'g'ari'ladi'.

1*2*3*4*5

volatile klasi' wo'zgeriwshilari. Yeager programmada wo'zgeriwshin qandayda-bir si'rtqi' operator yamasa basqa programma menen bo'lew ushi'n isletiw za'ru'r bolatug'i'n bolsa, wol volatile modifikatori menen dag'aza yetiledi. Kompilyator bunday modifikatorli wo'zgeriwshin registrga jaylasti'ri'wg'a ha'reket yetpeydi. Bunday wo'zgeriwshiler dag'azasi'na mi'sal to'mende keltirilgen:

```
volatile short port_1;  
volatile const int Adress=0x00A2;
```

Mi'saldan ko'rinib turi'пти', wolda, volatile modifikatorli turaqli' da dag'aza yetiliwi mu'mkin.

Atlar ken'isligi: Yekenin ayti'w kerek, programmaga qosi'lg'an bas bet fayllari'nda dag'aza yetilgen identifikator ha'm turaqli'lar kompilyator ta'repinen birden-bir global atlar ken'isligiga kiritiledi. Yeager programma ko'p mug'dardag'i' bas bet fayllardi' isletsa ha'm wodag'i' identifikatorlar(funksiya atlari' ha'm wo'zgeriwshiler atlari', klasslar atlari' ha'm hako'zalar) ha'm turaqli'lar atlari' ha'r qi'yli' programma du'ziwshiler ta'repinen wo'zbetinshe tu'rde ani'qlang'an bolsa, birdey atlardi' isletiw menen baylani'sli' mashqalalar ju'zege keliw mu'mkinshiligi' u'lken boladi'. Atlar ken'isligi tu'siniginin' kiritiliwi usi' mashqalani' ma'lum bir ma'niste sheshiwge ja'rдем beredi. Agar programmada jan'a identifikatordi ani'qlaw kerek bolsa ha'm tap shu nomni basqa modullarda yamasa kitapxanalarda isletiw xavfi bolatug'i'n bolsa, bul identifikatorlar ushi'n wo'zinin' jeke atlar ken'isligini ani'qlaw mu'mkin. Bug'an namespace gilt so'zinen paydalani'lg'an halda yeriwiladi:

```
namespace <atlar ken'isligi ati'>  
{  
    //dag'azalar  
}
```

Atlar ken'isligi ishinde dag'aza yetilgen identifikatorlar tek <atlar ken'isligi ati'> ko'rinis ken'isliginde boladi' ha'm ju'zege keliwi mu'mkin bolg'an keliwmovchiliklarding aldi' ali'nadi'.

Mi'sal si'pati'nda to'mendegi atlar ken'isligini jarataylik:

```
namespace Jeke_atlar  
{
```

```

int x, y, z;
void Menin_funksiyam(char belgi) ;
}

```

Kompilyatorga konkret atlar ken'isligidagi atlardi' isletiw kerekligin ko'rsetiw ushi'n ko'rinish oblasti'na ruxsat beriw a'meliden paydalani'w mu'mkin:

```
Jeke_atlar::x=5;
```

Yeger programma tekstinde konkret atlar ken'isligiga sali'sti'rg'anda ko'p murajat yetiletug'i'n bolsa woni'n' namespace operatorin isletiw arqali jazi'wdi' a'piwayi'lasti'ri'w mu'mkin:

```
using namespace <atlar ken'isligi ati'>;
```

Mi'sali,

```
using namespace Jeke_atlar;
```

ko'rsetpesi kompilyatorga, bunnan keyin toki' na'wbettegi using uchramaguncha Jeke_atlar ken'isligidagi atlar isletiliwi kerekligin an'latadi':

```
x=0; y=z=10;
```

```
Menin'_funksiyam('A') ;
```

Programma ha'm wog'an qosi'lg'an bas bet fayllari' ta'repinen ani'qlanatug'i'n atlar ken'isligi std dep ataladi'. Standart ken'islikke wo'tiw kerek bolsa

```
using namespace std;
```

ko'rsetpesi beriledi.

Yeger qandayda-bir atlar ken'isligidagi bo'lek bir atqa murajat yetiw za'ru'r bolsa, using operatorininin' basqa formasi'nda paydalani'ladi'.

Mi'sal ushi'n

```
using namespace std;
```

```
using namespace Jeke_atlar::x;
```

ko'rsetpesi x identifikatorini Jeke_atlar ken'isliginden isletiw kerekligin an'latadi'.

Soni' atap ko'rsetiw za'ru'r, using namespace operatori standart atlar ken'isligi ko'rinish ken'isligin jasi'radi' ha'm wodagi' atqa murajat yetiw ushi'n ko'rinish oblasti'na ruxsat beriw a'melinen std:: paydalani'w za'ru'r boladi'.

Atlar ken'isligi funksiya ishinde dag'aza yetiliwi mu'mkin yemes, lekin wolar basqa atlar ken'isligi ishinde dag'aza yetiliwi mu'mkin. Ishpe- ish jaylasqan atlar ken'isligindeki identifikatorg'a murajat yetiw ushi'n woni' qanti'p alg'an barli'q atlar ken'isligi atlar izbe-iz tu'rde ko'rsetiliwi kerek. Mi'sal ushi'n, to'mendegi ko'riniste atlar ken'isligi dag'aza yetilgen bo'lsin:

```
namespace joqari
{
    ...
    namespace worta
    {
        ...
        namespace Ishki
        {
            int ishki_n;
        }
    }
}
```

ishki_n wo'zgeriwshisine murajat to'mendegi ko'riniste boladi':

```
Joqari::Worta::Ishki::ishki_n=0;
```

Atlar ken'isliginde funksiyaning ja'riyalawda atlar ken'isliginde tek funksiya prototipinin ja'riyalaw ha'm funksiya denesini basqa wori'nda ja'riyalaw maqul variant yesaplanadi'. Bul jag'dayni'n' ko'rinisine mi'sal:

```
namespace Atlar_ken'isligi
{
    char c;
    int I;
    void Funksiya(char Bayraq) ;
}

...
void Atlar_ken'isligi::Funksiya(char Bayraq)
// funksiya denesi
}
```

Uluwma alg'anda, wo'z ati'na iye bolmag'an atlar ken'isligini ja'riyalaw mu'mkin. Bul halda namespace gilt so'zinen keyin hesh narse jazi'lmaydi'. Mi'sal ushi'n

```
namespace
{
    char c_nomsiz;
    int i_atsiz;
}
```

ko'rinishidagi atlar ken'isligi elemenlerine murajat hesh bir prefiks isletbesden a'melge asi'ri'ladi'. Atsiz atlar ken'isligi tek wo'zi dag'aza yetilgen fayl shegarasi'nda a'mel yetedi.

C++ tili atlar ken'isliginin' psevdonimlarin' ani'qlaw imkaniyati'n beredi. Bul jol arqali' atlar ken'isligini basqa at menen isletiw mu'mkin boladi'. Mi'sali', atlar ken'isligi ati' uzi'n bolg'anda wog'an qi'sqa at menen murajat yetiw:

```
namespace Ju'da_uzi'n_atli_ken'islik
{
    float: y;
}
Ju'da_uzi'n_atli_ken'islik::y=0;
namespace Qi'sqa_at = Ju'da_uzi'n_atli_ken'islik;
Qi'sqa_atm::y = 13.2;
```

Jaylasti'ri'latug'i'n (inline) funksiyalar: Kompilyator islewi na'tiyjesinde ha'r bir funksiya mashina kodi' ko'rinishinde boladi'. Yeger programmada funksiyanin' shaqi'ri'w ko'rsetpesi bolsa, sol wori'nda funksiyanin' adresi boyi'nsha shaqi'ri'wdi'n' mashina kodi' qa'liplesedi. A'detde funksiyanin' shaqi'ri'w protsessor ta'repinen qosi'msha waqi't ha'm yad resurslarini talap yetedi. Usi'ni'n' sebepinen, kompilyatorga, yeger shaqi'ri'latug'i'n funksiya ko'lemi wonsha u'lken bolmag'an jag'daylarda, funksiyanin' shaqi'ri'w kodi' worni'na funksiya denesin wo'zin jaylasti'ri'wg'a ko'rsetpe beriw mu'mkin. Bul jumi's funksiya prototipini inline gilt so'zi menen ja'riyalaw arqali' a'melge asi'ri'ladi'. Na'tiyjede ko'lemi oshgan, lekin sali'sti'rg'anda tez atqari'latug'i'n programma kodi' ju'zege keledi.

Funksiya kodi' jaylasti'ri'latug'i'n programmaga mi'sal,

```
#include <iostream>
#include <math.h>
using namespace std;
inline int Summa(int, int) ;
int main()
{
    int a=2, b=6, c=3;
    string jana_qatar ="\n";
    cout<<Summa(a, b) <<jana_qatar;
    cout<<Summa(a, c) <<jana_qatar;
    cout<<Summa(b, c) <<jana_qatar;
    return 0;
}
int Summa(int x, int y)
{
    return x+y;
}
```

Keltirilgen programma kodi'n payda yetiwde Summa() funksiyasi' shaqi'ri'lg'an jaylarg'a woni'n' denesindegi *buyri'qlar jaylasti'ri'ladi'*.

Soraw ha'm tapsi'rmalar

1. *Atlar ken'isligi qanday dag'aza yetiledi?*
2. *Aftomat wo'zgeriwshiler.*
3. *Registr wo'zgeriwshiler.*
4. *Si'rtqi' wo'zgeriwshiler.*
5. *Statikali'q wo'zgeriwshiler.*
6. *volatile klasi' wo'zgeriwshiler.*
7. *:: a'melin ani'qlama berin'.*
8. *:: a'melidan qanday jag'daylarda paydalani'ladi'?*
9. *Atlar fazasiga mi'sal keltiring?*

10. *Avtomat wo'zgeriwshiler qanday jag'daylarda qo'llaniladi?*
11. *Normal fazasidagi bo'lek bir atqa murajat qanday a'melge asi'ri'ladi'?*
12. *Wo'z ati'na iye bolmag'an atlar fazasini ani'qlama berin'.*

The keyword **static** can be applied to members of a class. When you declare a member of a class as **static**, you are telling the compiler that no matter how many objects of the class are created, there is only one copy of the **static** member. That is,

a **static** member is *shared* by all objects of the class. All **static** data is initialized to zero when the first object is created, if no other initialization is present.

When you declare a **static** data member within a class, you are *not* defining it. Instead, you must provide a global definition for it elsewhere, outside the class. This is done by redeclaring the **static** variable, using the scope resolution operator to identify which class it belongs to. This causes storage to be allocated for the **static** variable.

The creation of the **namespace** keyword was a response to these problems. Because it localizes the visibility of names declared within it, a namespace allows the same name to be used in different contexts, without conflicts arising. Perhaps the most noticeable beneficiary of **namespace** is the C++ standard library. Prior to **namespace**, the entire C++ library was defined within the global namespace (which was, of course, the only namespace). With the addition of **namespace**, the C++ library is now defined within its own namespace, called **std**, which reduces the chance of name collisions. You can also create your own namespaces within your program to localize the visibility of any names that you think may cause conflicts. This is especially important if you are creating class or function libraries.

Namespace Fundamentals

The **namespace** keyword allows you to partition the global namespace by creating a declarative region. In essence, a **namespace** defines a scope. The general form of **namespace** is shown here:

```
namespace name {  
    // declarations  
}
```

Anything defined within a **namespace** statement is within the scope of that namespace.

The following program is an example of a **namespace**. It localizes the names used to implement a simple countdown counter class. In the namespace are defined the **counter** class, which implements the counter, and the variables **upperbound** and **lowerbound**, which contain the upper and lower bounds that apply to all counters.

14-Lekciya: Rekursiv ha'm qayta ju'kleniwshi funksiyalar

Joba:

1. Rekursiv funksiyalar
2. Qayta ju'kleniwshi funksiyalar

Tayansh so'zler: *Rekursiv funksiyalar, qayta ju'kleniwshi funksiyalar, double tu'ri.*

Rekursiv funksiyalar: Joqari'da belgilengenler yetilgen si'yaqli' rekursiya dep funksiya denesinde sol funksiyani'n' wo'zin shaqi'ri'wi'na ayti'ladi'. Rekursiya yeki xil boladi':

- 1) a'piwayi'-yeger funksiya wo'z denesinde wo'zin shaqi'rsa;
- 2) quralli'-yeger birinshi funksiya yekinshi funksiyani' shaqi'rsa, yekinshisi bolsa wo'z gezeginde birinshi funksiyani' shaqi'rsa.

A'detde rekursiya matematikada ken' qollani'ladi'. Sebebi ko'pshilik matematikali'q formulalar rekursiv ani'qlanadi'. Mi'sal si'pati'nda faktorialni hisoblash formulasi'n

$$n! = \begin{cases} 1, & \text{yeger } n=0; \\ n*(n-1)!, & \text{yeger } n>0, \end{cases}$$

ha'm sandin' butun darajasini hisoblashni ko'rishimiz mumkin:

$$x^n = \begin{cases} 1, & \text{yeger } n=0; \\ x*x^{n-1}, & \text{yeger } n>0. \end{cases}$$

Ko'rinib turipdi, navbatdagi qiymatni hisoblash uchun funksiyani'n' «oldingi qiymati» ma'lum bo'lishi kerak. C++ tilida rekursiya matematikadagi rekursiyaga o'xshash. Buni yuqoridagi misollar uchun tuzilgan fuiksiyalarda ko'rish mumkin. Faktorial uchun:

```
long F(int n)
{
    if(!n)
        return 1;
    else
        return n*F(n-1);
}
```

}

Berilgan haqiqiy x sandin' n- darajasini hisoblash funksiyasi:

```
double Putin_Dareje(double x, int n)
{
    if(!n)
        return 1;
    else
        return x*Putin_Dareje(x, n-1);
}
```

Agar faktorial funksiyasiga $n > 0$ qiymat berilsa, quyidagi holat ro'y beradi: shart operatoridin' else shoxidagi qiymati(n qiymati) stekda eslab qolinadi. Noma'lumlardi hisoblash uchun shu funksiyaning o'zi «alding'i» qiymat(n-1 qiymati) bilan bilan chaqiriladi. O'z navbatida, bul qiymat ham eslab qolinadi(stekka joylanadi) ha'm yana funksiya chaqiriladi ha'm hakoza. Funksiya $n=0$ qiymat bilan chaqirilganida if operatorinin' sharti(!n) rost bo'ladi ha'm «return 1;» amali bajarilib, ayni shu chaqirish bo'yicha 1 qiymati qaytariladi, Shundan keyin «teskari» jarayon boshlanadi - stekda saqlangan qiymatlar ketma-ket olinadi ha'm ko'paytiriladi: oxirgi qiymat aniqlangandan keyin(1), u undan oldingi saqlangan qiymatga 1 qiymatiga ko'paytirib $F(1)$ qiymati hisoblanadi, bul qiymat 2 qiymatiga ko'paytirish bilan $F(2)$ topiladi ha'm hakoza. Jarayon $F(n)$ qiymatini hisoblashgacha «ko'tarilib» boradi. Bul jarayonni, $n=4$ uchun faktorial hisoblash sxemasini 5.2-rasmda ko'rish mumkin:

↓	$F(4)=4 \cdot F(3)$	↑	$F(4)=4 \cdot 6$						
↓	$F(3)=3 \cdot F(2)$	↓	$F(3)=3 \cdot F(2)$	↓	$F(3)=3 \cdot F(2)$	↑	$F(3)=3 \cdot 2$		
↓	$F(2)=2 \cdot F(1)$	↓	$F(2)=2 \cdot F(1)$	↑	$F(2)=2 \cdot 1$				
↓	$F(1)=1 \cdot F(0)$	↑	$F(1)=1 \cdot 1$						
↑	$F(0)=1$								

5.2-su'wret. 4! Yesaplaw sxemasi'

Rekursiv funksiyalardi' tuwri' a'mel yetiwi ushi'n rekursiv murajatlardi'n' toqtap qali'w sha'rti boli'wi' kerek. Bolmasa rekursiya toqtamasligi ha'm wo'z gezeginde funksiya jumi'si' tawi'si'lmaslig'i' mu'mkin. Faktorial yesaplawi'nda rekursiv tu'siwlerdin' toqtap qali'w

sha'rti funksiya parametri $n=0$ boli'wi' boli'p tabi'ladi'(sha'rt operatori'ni'n' ras shoxi).

Ha'r bir rekursiv murajat qosi'msha yad talap yetedi-funksiyalardi'n' lokal obyektleri(wo'zgeriwshileri) ushi'n ha'r bir murajatda stekdan tazadan jay ajrati'ladi'. Mi'sali', rekursiv funksiya'ga 100 ret murajat bolsa, ja'mi 100 lokal obyektlarding kompleksi ushi'n jay ajrati'ladi'. Ayi'ri'm jag'daylarda, ju'da' ko'p rekursiya bolg'anda, stek wo'lshemi sheklengenligi sebepli(real rejimde 64Kb wo'lshemne shekem) wol to'lib ketiwi mu'mkin ha'm bul jag'dayda programma wo'z jumi'si'n «Stek to'lib ketti» xabari'menen to'qtadi'.

To'mende, rekursiya menen na'tiyjeli sheshiledigan «Xanoy minari'» ma'selesin ko'reylik.

Ma'sele. Ush A, B, C qazi'q ha'm n-ta har-xil wo'lshemli xalqalar bar. Xalqalardi wo'lshemleri wo'siw rejiminde 1 den n ge shekem ta'rtilengen. Boshda barli'q xalqalar A qazi'qqa 5.3a-suwretdegi si'yaqli' jaylasti'ri'lg'an. A qazi'qdag'i' barli'q xalqalardi B qazi'qqa ja'rdemshi C qazi'qdan paydalang'an halda, to'mendegi qag'i'ydalarg'a a'mel yete oti'ri'p wo'tkeriw talap yetiledi: xalqalardi birden ko'shiriw kerek ha'm u'lken wo'lshemli xalqani kishi wo'lshemli xalqa u'stine qoyi'w mu'mkin yemes.

Su'wret

5.3-su'wret. Xanoy minari' ma'selesin sheshiw procesi

A'meller izbe-izligin baspadan shi'g'aratug'i'n(«Xalqa q den r ga wo'tkerilsin» ko'rinisinde, bunda q ha'm r-5.3-suwretdegi A, B yamasa C xalqalar). Berilgen n ta xalqa ushi'n ma'seleni yechilsin.

Ko'rsetpe : xalqalardi A den B ga tuwri' wo'tkeriwde 5.3b-su'wretlerdegi jag'day ju'zege keledi, yag'ni'y n xalqani A den B wo'tkeriw mi'sali' n-1 halqasi'n A den C ga wo'tkeriw, ha'mde bir xalqani A den B wo'tkeriw ma'selesine keledi. Wodan keyin C qazi'qdag'i' n-1 xalqali A qazi'q ja'rdeminde B qazi'qqa wo'tkeriw ma'selesini ju'zege keledi ha'm hakoza.

```
#include <iostream>
```

```

using namespace std;
void Hanoy(int n, char a='A', char b='B', char c='C')
{
    if(n)
    {
        Hanoy(n-1, a, c, b) ;
        cout<<"Xalqa "<<a<<" den "<<b<<"g'a wo'tkerilsin\n";
        Hanoy(n-1, c, b, a) ;
    }
}
int main()
{
    unsigned int Xalqalar_Sani;
    cout <<"Hanoy minari' ma'selesi "<<endl;
    cout<<"Xalqalar sani'n kiritin': ";
    cin >>Xalqalar_Sani;
    Hanoy(Xalqalar_Sani) ;
    return 0;
}

```

Xalqalar sani' 3 bolg'anda(Xalqalar_Sani'=3) programma ekrang'a halqalardi'

ko'shiriw boyi'nsha a'meller izbe-izligin baspadan shi'g'aradi':

```

Xalqa A den B ga wo'tkerilsin
Xalqa A den C ga wo'tkerilsin
Xalqa B den C ga wo'tkerilsin
Xalqa A den B ga wo'tkerilsin
Xalqa C den A ga wo'tkerilsin
Xalqa C den B ga wo'tkerilsin
Xalqa A den B ga wo'tkerilsin

```

Rekursiya shi'rayli', i'qsham ko'ringeni menen yadni' u'nemlew ha'm yesaplaw waqti'n kemeytiw ko'z qarasi'nan mu'mkinshiligi bari'nsha woni' taticri yesaplaw menen almasti'rilgani maqul. Mi'sali', x haqi'yqi'y sani'ni'n' n-da'rejesin yesaplawdi'n' to'mendegi sheshim varianti' sali'sti'rg'anda kem resurs talap yetedi(n-pu'tin belgisiz san) :

```

double Putin_daraja(double x, int n)
{
    double p=1;

```

```

    for(int i=1; i<=n; i++)
        p*=x;
    return p;
}

```

Lekin sonday ma'seleler bar, wolardi' sheshiwde rekursiya ju'da' na'tiyjeli, ha'tte, birden-bir usul boli'p tabi'ladi'. Atap aytqanda, grammatik analiz ma'selelerinde rekursiya wog'i'ri' won'ayli'q yesapladi.

Qayta ju'kleniwshi funksiyalar: Ayi'ri'm algoritmlar berilgenlerdin' har-xil tu'rdegi ma'nisleri' ushi'n qollani'wi' mu'mkin. Mi'sali', yeki sanni'n' maksimumini tabi'w algoritmida bul sanlar pu'tin yamasa haqi'yqi'y tu'rde boli'wi' mu'mkin. Bunday jag'daylarda bul algoritmlar a'melge asi'ri'lg'an funksiyalar atlari' birdey bolg'ani' maqul. Bir neshe funksiyani' birdey ataw, lekin har-xil tu'rdegi parametrlar menen isletiw funksiyani' qayta ju'klew dep ataladi'.

Kompilyator parametrlar tu'rine ha'm sani'na qaray saykes funksiyani' shaqi'radi'. Bunday a'meldi «hal yetiw a'meli» dep ataladi' ha'm woni'n' maqseti parametrlerge ko're a'yne(sali'sti'rg'anda) tuwri' keletug'i'n funksiyani' shaqi'ri'w boli'p tabi'ladi'. Yeger bunday funksiya tabi'lmasa kompilyator qa'telik haqqi'nda xabar beredi. Funksiyani' ani'qlawda funksiya qaytaratug'i'n ma'nis tu'rinin' a'hmiyeti joq. Mi'sal:

```

#include <iostream>
using namespace std;
int max(int, int) ;
string max(string, string) ;
float max(float, float) ;
int max(int, int, int) ;
int main()
{
    int a;
    int b;
    string c;
    string d;
    int k;
    float x, y;
    cin>>a>>b>>k>>c>>d>>x>>y;
}

```

```

        cout<<max(a, b)<<max(c, d) <<max(a, b, k)<<max(x, y) ;
    }
    int max(int i, int j)
    {
        return(i>j) ?i:j;
    }
    string max(string s1, string s2)
    {
        return(s1>s2) ?s1:s2;
    }
    float max(float x, float y)
    {
        return(x>y) ?x:y;
    }
    int max(int i, int j, int k)
    {
        return (i>j)?(i>k? i:k):((j>k) ?j:k) ;
    }

```

Yeger funksiya shaqi'rilishida argument tu'ri woni'n' prototipidagi tap sol wori'ndag'i' parametr tu'rine saykes kelmasa, kompilyator woni' parametr tu'rine keltirilishga ha'reket yetedi-double ha'm string tu'rlerin int tu'rine, float tu'rin double tu'rine ha'm int tu'rin double tu'rine wo'tkeriwge.

Qayta ju'kleniwshi funksiyalardan paydalani'wda to'mendegi qag'i'ydalarg'a a'mel qi'li'w kerek:

-qayta ju'kleniwshi funksiyalar bir ko'rinis ken'isliginde boli'wi' kerek;

-qayta ju'kleniwshi funksiyalarda sha'rtlesiw boyi'nsha parametrlar isletilse, bunday parametrlar barli'q qayta ju'kleniwshi funksiyalarda da isletiliwi ha'm wolar birdey ma'nisge i'yelew kerek;

-yeger funksiyalar parametrlarinin' tu'ri tek "const" ha'm '&'

belgileri menen pari'q yetetug'i'n bolsa, bul funksiyalar qayta ju'klenbeydi.

Savollar ha'm topshiriqlar

1. *Rekursiya ne?*
2. *Rekursiya tu'rleri ha'm wolardi'n' atlari'.*
3. *Qayta ju'kleniwshi funksiyalardan paydalani'w qag'i'ydalari'.*
4. *Qayta ju'klenbeytug'i'n funksiyalarg'a mi'sallar keltiring.*
5. *Rekursiv funksiyalardi' ani'qlama berin'.*
6. *Rekursiv murajat qanday a'melgem asi'ri'ladi'?*
7. *Rekursiv funksiyalarg'a mi'sal keltirin'.*
8. *Funksiyani' qayta ju'klew tu'sinigin ani'qlama berin'.*
9. *Sheshiw a'meli.*
10. *Qayta ju'kleniwshi funksiyalar haqqi'nda uli'wma tu'sinik berin'.*
11. *Qayta ju'kleniwshi funksiyalarg'a mi'sal keltirin'.*
12. *Qanday funksiyalar qayta ju'klenbeydi?*

Function Overloading and Ambiguity

Before concluding this chapter, we must examine a type of error unique to C++: *ambiguity*. It is possible to create a situation in which the compiler is unable to choose between two (or more) correctly overloaded functions. When this happens, the situation is said to be *ambiguous*. Ambiguous statements are errors, and programs containing ambiguity will not compile.

By far the main cause of ambiguity involves C++'s automatic type conversions. C++ automatically attempts to convert the type of the arguments used to call a function into the type of the parameters defined by the function. Here is an example:

```
int myfunc(double d);  
.  
.  
.  
cout << myfunc('c'); // not an error, conversion applied
```

As the comment indicates, this is not an error, because C++ automatically converts the character **c** into its **double** equivalent. Actually, in C++, very few type conversions of this sort are disallowed. While automatic type conversions are convenient, they are also a prime cause of ambiguity. Consider the following program:

Function Overloading

In this section, you will learn about one of C++'s most exciting features: function overloading. In C++, two or more functions can share the same name, as long as their parameter declarations are different. In this situation, the functions that share the same name are said to be *overloaded*, and the process is referred to as *function overloading*. Function overloading is one way that C++ achieves polymorphism.

Let's begin with a short sample program:

```
// Overload a function three times.
#include <iostream>
using namespace std;

void f(int i);           // integer parameter
void f(int i, int j);   // two integer parameters
void f(double k);       // one double parameter

int main()
{
    f(10);              // call f(int)

    f(10, 20);          // call f(int, int)

    f(12.23);           // call f(double)

    return 0;
}

void f(int i)
{
    cout << "In f(int), i is " << i << '\n';
}

void f(int i, int j)
{
    cout << "In f(int, int), i is " << i;
    cout << ", j is " << j << '\n';
}

void f(double k)
{
    cout << "In f(double), k is " << k << '\n';
}
```

15-Lekciya: Ko'rsetkishler ha'm murajatlar

Joba:

1. Ko'rsetkishler
2. Ko'rsetkishge baslang'i'sh ma'nis beriw
3. Ko'rsetkish u'stinde a'meller
4. Murajatlar
5. Ko'rsetkishler ha'm murajatlar funksiya parametri si'pati'nda.
6. Wo'zgeriwshen parametrli funksiyalar.

Tayansh so'zler: *Ko'rsetkishler, murajatlar, funksiya'ga ko'rsetkish, obyektga ko'rsetkish,, void ko'rsetkish, ko'rsetkishge baslang'i'sh ma'nis beriw, ko'rsetkish u'stinde a'meller, ko'rsetkishler ha'm murajatlar funksiya parametri si'pati'nda, wo'zgeriwshen parametrli funksiyalar.*

Ko'rsetkishler: Programma tekstinde wo'zgeriwshi dag'aza yetilgende, kompilyator wo'zgeriwshige yaddan jay ajratadi'. Basqasha aytqanda, programma kodi' yadqa ju'klengende berilgenler ushi'n, wolar jaylasatug'i'n segmenttin' basi'na sali'sti'rg'anda ji'li'si'wi'n, yag'ni'y sali'sti'rmali' adresin ani'qlaydi' ha'm obyekt kod payda yetiwde wo'zgeriwshi dus kelgen jayg'a woni'n' adresin jaylasti'radi'. Uluwma alg'anda, programmadagi turaqli'lar, wo'zgeriwshiler, funksiyalar ha'm klass obyektlar adreslerin yadni'n' bo'lek jayi'nda saqlaw ha'm wolar u'stinen a'meller wori'nlaw mu'mkin. Ma'nisleri' adres bolg'an wo'zgeriwshilerge ko'rsetkish wo'zgeriwshiler dep ataladi'.

Ko'rsetkish u'sh xil tu'rde boli'wi' mu'mkin:

- qandayda-bir obyektga, atap aytqanda wo'zgeriwshige ko'rsetkish;
- funksiya'ga ko'rsetkish;
- void ko'rsetkish.

Ko'rsetkishnin' bul qa'siyetleri woni'n' qabi'llawi' mu'mkin bolg'an ma'nisleri'nda pari'q yetedi.

Ko'rsetkish a'l'bette qandayda-bir tu'rge baylani'sqan boli'wi' kerek, yag'ni'y wol ko'rsetken adresde qanday da ma'nis jaylani'wi' mu'mkin

ha'm bul ma'nisni'n' yadta qansha jay iyelewi aldi'nan ma'lum boli'wi sha'rt.

Funksiyag'a ko'rsetkish. Funksiyag'a ko'rsetkish programma joylasqan yaddag'i funksiya kodi'ni'n' baslang'i'sh adresin ko'rsetedi yag'ni'y funksiya shaqi'rilganda basqari'w a'yne sol adreske uzati'ladi'. Ko'rsetkish arqali' funksiyaning a'piwayi' yamasa quralli' shaqi'ri'w a'melge asiri'w mu'mkin. Bunda funksiya woni'n' ati' boyi'nsha yemes, ba'lki funksiya ko'rsetiwshi wo'zgeriwshi arqali' shaqi'ri'ladi'. Funksiyaning bas funksiya argumenti si'pati'nda uzati'w da funksiya ko'rsetkishi arqali' atqari'ladi'. Funksiyag'a ko'rsetkishning jazi'li'w sintaksisi to'mendegishe:

```
<tur>(*<at>)(<parametrlar(yamasa tu'rlerdin') dizimi>);
```

Bunda <tur> - funksiya qaytaratug'i'n ma'nis tu'ri; *<at> - ko'rsetkish; wo'zgeriwshinin' ati'; <parametrlar(yamasa tu'rlerdin') dizimi> - funksiya parametrlarining (yamasa wolardi'n' tu'rlerining) dizimi.

Mi'sali:

```
int(*fun)(float, float);
```

Bul jerde pu'tin san tu'rinde ma'nis qaytaradi' fun ati'ndag'i funksiya ko'rsetkish dag'aza yetilgan ha'm wol yeki haqi'yqi'y tu'rdegi parametrlarga iye.

Ma'sele. Berilgan pu'tin $n=100$ ha'm a, b -haqi'yqi'y sanlar ushi'n

$f_1(x)=5\sin(3x)+x$, $f_2(x)=\cos(x)$ ha'm $f_3(x)=x^2+1$ funksiyalar ushi'n $\int_b^a f(x)dx$

integrali'n tuwri' to'rtmuyushlar formulasi' menen taqriban yesaplansin:

$$\int_b^a f(x)dx \approx h[f(x_1)+f(x_2)+\dots+f(x_n)],$$

bul jerde $h = \frac{b-a}{n}$, $x_i = a + ih - h/2$, $i=1..n$.

Programma bas funksiya, integral yesaplaw funksiyasi' ha'm yeki matematikali'q funksiyalar- $f_1(x)$ ha'm $f_3(x)$ ushi'n ani'qlang'an funksiyalardan tashkil tabadi', $f_2(x) = \cos(x)$ funksiyaning adresi math.h bas bet fayli'nan ali'nadi'. Integral yesaplaw funksiyasi'na funksiya ko'rsetkish arqali' integrali' yesaplanatug'i'n funksiya adresi, a ha'm b -

integral shegaralari' ma'nisleri' uzati'ladi'. Arali'qdi' boli'wlar sani'-n global turaqli' yetip dag'aza yetiledi.

```
#include <iostream>
#include <math.h>
using namespace std;
const int n=100;
double f1(double x)
{
    return 5*sin(3*x) +x;
}
double f3(double x)
{
    return x*x+1;
}
double Integral(double(*f)(double), double a, double b)
{
    double x, s=0;
    double h=(b-a) /n;
    x=a-h/2;
    for(int i=1; i<=n; i++)
        s+=f(x+=h);
    s*=h;
    return s;
}
int main()
{
    double a, b;
    int menu;
    while(1)
    {
        cout<<"\nJumi's rejimin saylan':\n";
        cout<<" 1:f1(x) =5*sin(3*x) +x integraldi'
        yesaplaw\n";
        cout<<"2:f2(x) =cos(x) integraldi' yesaplaw\n";
        cout<<"3:f3(x) =x^2+1 integraldi' yesaplaw\n";
        cout<<"0 rogrammadan shi'g'i'w\n";
        do
        {
```

```

        cout<<"Jumi's rejimi-> ";
        cin>>menu;
    }while(menu<0 || menu>3);

    if(!menu) break;
    cout<<"Integral aralig'inin' to'men shegarasi' a=";
    cin>>a;
    cout<<"Integral aralig'inin' joqari' shegarasi'
    b=";
    cin>>b;
    cout<<"Funksiya integrali' S=";
    switch(menu)
    {
        case 1:cout<<Integral(f1, a, b)<<endl;
                break;
        case 2:cout<<Integral(cos, a, b)<<endl;
                break;
        case 3: cout<<Integral(f3, a, b)<<endl;
                break;
    }
}
return 0;
}

```

Programmanin' jumi'si' sheksiz ta'kirarlaw operatori' denesin wori'nlawdan ibarat. Ta'kirarlaw denesinde paydalaniwshi'g'a jumi's rejimin tan'law boyi'nsha menyu usi'ni's yetiledi:

Jumi's rejimin saylan':

```

1:f1(x)=5*sin(3*x) +x integraldi' yesaplaw
2:f2(x)=cos(x) integraldi' yesaplaw
3:f3(x)=x^2+1 integraldi' yesaplaw
0 rogrammadan shi'g'i'w
Jumi's rejimi->

```

Paydalani'wshi' 0 ha'm 3 arali'g'indag'i' pu'tin sandi' kirgiziwi kerek. Yeger kiritilgen san(menu wo'zgeriwshi ma'nisi) 0 bolsa, break operatori' ja'rdeminde ta'kirarlawdan, keyin programmada shi'g'i'ladi'. Yeger menu ma'nisi 1 ha'm 3 arali'g'inda bolsa, integraldi'n' to'men ha'm joqari'

shegaralari'n kirgiziw soraladi', ha'mde integral() funksiyasi' saykes funksiya adresi menen shaqi'ri'ladi' ha'm na'tiyje baspadan shi'g'ari'ladi'.

Sog'an itibar beriw kerek, integral shegaralari'ni'n' ma'nisleri'n tuwri kiritiliwine paydalani'wshi' juwapker.

Obyektga ko'rsetkish. Qandayda bir obyektga ko'rsetkish(atap aytqanda wo'zgeriwshige). Bunday ko'rsetkishde ma'lum tu'rdegi(tayansh yamasa tuwi'ndi'li'q tu'rdegi) berilgenlerdin' yaddag'i' adresi jaylasadi'. Obyektga ko'rsetkish to'mendegishe dag'aza yetiledi:

```
<tu'r> *<at>;
```

Bul jerde <tu'r>-ko'rsetkish ani'qlaytug'i'n adresdegi ma'nisni'n' tu'ri <nom>-obyekt ati'(identifikator). Yeger bir tu'rde bir neshe ko'rsetkishler dag'aza yetiletug'i'n bolsa, ha'r bir ko'rsetkish ushi'n '*' belgisi qoyi'li'wi' sha'rt:

```
int *i, j, *k;  
float x, *y, *z;
```

Bul mi'salda i ha'm k-pu'tin tu'rdegi ko'rsetkishler ha'm j-pu'tin tu'rdegi wo'zgeriwshi, yekinshi operatoroda x-haqi'yqi'y wo'zgeriwshi ha'm y, z-haqi'yqi'y tu'rdegi ko'rsetkishler dag'aza yetilgen.

void ko'rsetkish. Bul ko'rsetkish obyekt tu'ri aldi'nan belgisiz bolg'anda isletiledi. Void ko'rsetkishinin' za'ru'rli abzalli'qlari'nan biri-wog'an ha'r qanday tu'rdegi ko'rsetkish ma'nisin ju'klew mu'mkinligi boli'p tabi'ladi'. Void ko'rsetkish adresindegi ma'nisti isletiwden aldi'n, woni' ani'q bir tu'rge oshkor tu'rde keltiriw kerek boladi'. Void ko'rsetkishni ja'riyalaw to'mendegishe boladi':

```
void *<at>;
```

Ko'rsetkishnin' wo'zi turaqli' yamasa wo'zgeriwshen boli'wi' ha'm turaqli' yamasa wo'zgeriwshiler adresine ko'rsetiwi mu'mkin, mi'sali':

```
int i; // pu'tin wo'zgeriwshi  
const int ci=1; // pu'tin turaqli'  
int *pi; // pu'tin wo'zgeriwshige ko'rsetkish  
const int *pci;//pu'tin turaqli'g'a ko'rsetkish  
int *const cp=&i;//butun wo'zgeriwshige turaqli'  
ko'rsetkish
```

```
const int*const cpc=&ci;//pu'tin turaqli'g'a turaqli'  
ko'rsetkish
```

Mi'sallardan ko'rinib turi'пти', wolda, '*' ha'm ko'rsetkish ati' arasi'nda turg'an const modifikatori tek ko'rsetkishnin' wo'zine tiyisli yesaplanadi' ha'm woni' wo'zgeriw mu'mkin emasligini an'latadi', '*' belgisinen chapda turg'an const bolsa ko'rsetilgen adresdegi ma'nis turaqli' yekenin an'latadi'.

Ko'rsetkishge ma'nisti beriw ushi'n '&'-adresdi ali'w a'meli isletiledi.

Ko'rsetkish wo'zgeriwshilerinin' a'mel yetiw oblasti', jasaw da'wiri ha'm ko'rinis oblasti' uli'wma qag'i'ydalarg'a bo'ysunadi.

Ko'rsetkishge baslang'i'sh ma'nis beriw: Ko'rsetkishler ko'binese dinamikali'q yad(basqasha ati' «uyum» yamasa

«heap») menen baylani'sli' halda isletiledi. Yadni'n' dinamikali'q deyilishiga sebep, bul tarawdag'i' bos yad programma islew processinde, kerekli waqti'nda aji'rati'p ali'nadi' ha'm za'ru'rat qolmaganida qaytari'ladi'(bosatiladi). Keyinirek, bul yad bo'legi programma ta'repinen basqa maqsette tag'i' isletiliwi mu'mkin. Dinamikali'q yadqa tek ko'rsetkishler ja'rdeminde murajat yetiw mu'mkin. Bunday wo'zgeriwshiler dinamikali'q wo'zgeriwshiler dep ataladi' ha'm wolardi' jasaw waqi'ti' jarati'lg'an noqatdan baslap programma aqi'ri'i'na shekem yamasa oshkor tu'rde jog'ati'lg'an(baylani'sqan yad bosatilgan) jayi'nasha boladi'.

Ko'rsetkishlerdi ja'riyalawda wog'an baslang'i'sh ma'nisler beriw mu'mkin. Baslang'i'sh ma'nis(initsializator) ko'rsetkish ati'nan keyin yamasa qawi's ishinde yamasa '=' belgiden keyin beriledi. Baslang'i'sh ma'nisler to'mendegi usi'llar menen beriliwi mu'mkin:

I. Ko'rsetkishge a'melde bolg'an obyektin' adresin beriw:

a) adresdi ali'w a'mel arqali':

```
int i=5, k=4; // pu'tin wo'zgeriwshiler  
int *p=&i; //p ko'rsetkishge i wo'agariwshinin'  
// adresi jazi'ladi'  
int *p1(&k);// p1 ko'rsetkishge k wo'zgeriwshinin'  
// adresi jazi'ladi'
```

b) basqa initsializatsiyalangan ko'rsetkishni ma'nisin beriw:

```
int *r=p;//p aldi'n dag'aza yetilgen ha'm ma'nisge iye
// bolg'an ko'rsetkish
```

v) massiv yamasa funksiya ati'ni'n' beriw:

```
int b[10]; // massivti ja'riyalaw
int *t=b; //massivtin' baslang'i'sh adresin beriw
void f(int a) {/*... */} // funksiyanı' ani'qlaw
void(*pf)(int) ; //funksiyag'a ko'rsetkishni ja'riyalaw
pf=f; //funksiya adresin ko'rsetkishge beriw
```

II. Oshkor tu'rde yadni'n' absolyut adresin beriw:

```
char *vp =(char*) 0xB8000000;
```

Bunda 0xB8000000-won alti'liq turaqli' san ha'm(char*) -tu'rge keltiriw a'meli boli'p, wol vp wo'zgeriwshin yadni'n' absolyut adresidegi baytlardi char si'pati'nda qayta isleytug'in ko'rsetkish tu'rine aylantirilishini an'latadi'.

III. Bos ma'nis beriw:

```
int *s=NULL;
int *r=0;
```

Birinshi qatarda arnawli' NULL turaqli'i' isletilingen, yekinshi qatarda 0 ma'nis isletilingen. Yeki halda da ko'rsetkish hesh qanday obyektga murajat yetpeydi. Bos ko'rsetkish tiykari'nan ko'rsetkishni ani'q bir obyektga ko'rsetip ati'rg'an yamasa joq yekenligin ani'qlaw ushi'n isletiledi.

IV. Dinamikali'q yadta new a'meli menen jay aji'rati'w ha'm woni' adresin ko'rsetkishge beriw:

```
int*n=new int; // birinshi operator
int*m=new int(10) ; // yekinshi operator,,
int*q=new int[10]; // u'shinshi operator
```

Birinshi operatorada new a'meli ja'rdeminde dinamikali'q yadta int ushi'n jetkilikli jay aji'rati'p ali'ni'p, woni'n' adresi n ko'rsetkishge juklenedi. Ko'rsetkishnin' wo'zi ushi'n jay kompilyatsiya waqti'nda ajrati'ladi'.

Yekinshi operatorada jay aji'rati'wdan ti'sqari' m adresine baslang'i'sh ma'nis-10 sani'n jaylasti'radi'.

U'shinshi operatorada int tu'rindagi 10 element ushi'n jay aji'rati'lg'an ha'm woni'n' baslang'i'sh adresi q ko'rsetkishge berilayapti.

Yad new a'meli menen aji'rati'lg'an bolsa, wol delete a'meli menen bosatilishi kerek. Joqari'dag'i' dinamikali'q wo'zgeriwshiler menen baylani'sqan Yad to'mendegishe bosatiladi:

delete n; delete m; delete[] q;

Yegerde yad new[] a'meli menen aji'rati'lg'an bolsa, woni' bosatish delete[] a'melin wo'lshevi ko'rsatilmagan halda qo'llaw kerek.

Yad bosatilganligiga qaramastan ko'rsetkishni wo'zin keyinirek qayta isletiw mu'mkin.

Ko'rsetkish u'stinde a'meller: Ko'rsetkish u'stinde to'mendegi a'meller wori'nlawg'a bolatug'i'n:

- 1) obyektga quralli' murajat yetiw a'meli;
- 2) ma'nis beriw a'meli;
- 3) ko'rsetkishge turaqli' ma'nisti qosi'w a'meli;
- 4) ayi'ri'w a'meli;
- 5) inkrement ha'm dekrement a'melleri;
- 6) sali'sti'ri'w a'meli;
- 7) tu'rge keltiriw a'meli.

Quralli' murajat yetiw a'meli ko'rsetkishdegi adres boyi'nsha jaylasqan ma'nisti ali'w yamasa ma'nis beriw ushi'n isletiledi:

```
char a; // char tu'rindagi wo'zgeriwshi dag'azasi',  
char *p=new char;//Ko'rsashkichni ja'riyalay otirip,  
wog'an
```

```
// dinamikali'q yaddan aji'rati'lg'an
```

```
// yadni'n' adresin beriw
```

```
*p='b'; //p adresine ma'nis jaylasti'ri'w
```

```
a=*p; //a wo'zgeriwshisine p adresindagi ma'nisti beriw
```

Soni' belgilengenler yetip wo'tiw kerek, yadni'n' ani'q bir jayi'ndag'i' bir waqti'n' wo'zinde bir neshe ha'm har-xil tu'rdegi ko'rsetkishlerge beriw mu'mkin ha'm wolar arqali' murajat yetilgende berilgendin' xil tu'rdegi ma'nisleri'n ali'w mu'mkin:

```
unsigned long int A=0xcc77ffaa;
```

```
unsigned short int *pint= unsigned short int*) &A;
```

```
unsigned char* pchar=( unsigned char*) &A;
```

```
cout <<hex<<A<<' '<<hex<<*pint<<' '<<hex<<(int) *pchar;
```

Ekrang'a har-xil ma'nisler baspadan shi'g'ari'ladi':

```
cc77ffaa ffaa aa
```

Wo'zgeriwshiler bir adresde jaylasqan halda pu'tin ma'nisni'n' ha'r qi'yli' bo'leklerin wo'zlestiredi. Bunda san ma'nisinin' yadta «teskari» jaylasi'wi' inabatqa ali'ni'wi' kerek.

Yeger har-xil tu'rdegi ko'rsetkishlerge ma'nisler berilsa, a'l'bette tu'rge keltiriw a'meliden paydalani'w kerek:

```
int n=5;
float x=1.0;
int *pi=&n;
float *px=&x;
void *p;
int *r, *r1;
px=(float*) &n;
p=px;
r=(int*) px;
r1=pi;
```

Ko'rsetkish tu'rin void tu'rine keltiriw a'melde ma'niske iye yemes. Tap sonday, tu'rleri birdey bolg'an ko'rsetkishler ushi'n tu'rdi keltiriw a'melin wori'nlawdi'n' ha'jeti joq.

Ko'rsetkish u'stinen atqari'latug'i'n arifmetikali'q a'mellerde avtomatik tu'rde tu'rlerdin' wo'lshemi yesapqa ali'nadi'.

Arifmetikali'q a'meller tek birdey tu'rdegi ko'rsetkishler u'stinen atqari'ladi' ha'm wolar tiykari'nan, massiv strukturalari'na ko'rsetkishler u'stinde atqari'ladi'.

Inkrement a'meli ko'rsetkishni massivtin' keyingi elementine, dekrement bolsa kerisinshe, bir aldi'ng'i' elementinin' adresine ko'shiredi. Bunda ko'rsetkishnin' ma'nisi sizeof(<massiv elementinin' turi>) ma'nisine wo'zgeredi. Yeger ko'rsetkishnin' ma'nisi k turaqli' ma'nisge asi'ri'lsa yamasa kamaytirilsa, wol k*sizeof(<massiv elementinin' turi>) ma'nisine wo'zgeredi.

Mi'sali':

```
short *p=new short[5];
long *q=new long [5];
p++; // p ma'nisi 2 asadi'
q++; // q ma'nisi 4 ge asadi'
```

```
q+=3; // ma`nisi 3*4=12 asadi'
```

Ko'rsetkishlerdin' ayi'rmasi' dep, wolar ayi'rmasi'ni'n' tu'r wo'lshemine i'di'rawi'na ayti'ladi'. Ko'rsetkishlerdi wo'z-ara qosi'w mu'mkin yemes.

Murajatlar: Murajatlar dag'azada, ko'rsetilgen attin' sinonimi si'pati'nda isletiledi, yag'ni'y bir wo'zgeriwshige xar xil at menen murajat yetiw imkaniyati'n beredi. Murajatdi' doimiy ma'nisge iye bolg'an ko'rsetkish dep qaraw mu'mkin. Murajat to'mendegishe dag'aza yetiledi:

```
<tu'r> & <nom>;
```

Bul jerde <tur>-murajat ko'rsetiwshi ma'nisni'n' tu'ri, '&' belgisi, wodan keyin jazi'lg'an <nom>-murajat tu'rindegi wo'zgeriwshi ati'. Basqasha aytqanda, '&' belgisine adresdi ali'w a'meli dep ataladi'.

Mi'sal:

```
int kol;
```

```
int & pa1=kol; // pa1 shaqi'ri'g'i, y kol
```

```
// wo'zgeriwshisinin' alternativ ati'
```

```
const char &cr='\n'; // cr-turaqli'g'a murajat
```

Murajatdi' isletiwde to'mendegi qag'i'ydalarg'a a'mel qi'li'w kerek: murajat, yeger wol funksiya parametri si'pati'nda isletilingen, extern menen suwretlewlengen ha'm klass maydani'na murajat yetilgennen ti'sqari' barli'q jag'daylarda baslang'i'sh ma'nisge i'yelewi kerek.

Murajat tiykari'nan funksiyalarda adres arqali' uzatiluvchi parametrlar si'pati'nda isletiledi.

Murajatdi' ko'rsetkishden ayi'rmasi'li'g'i' sonda, wol bo'lek yad egallamaydi ha'm tek wo'zgeriwshinin' basqa ati' si'pati'nda isletiledi.

Ko'rsetkishler ha'm murajatlar funksiya parametri si'pati'nda: Funksiya prototipida yamasa ani'qlani'w basi'nda ko'rsetilgen parametrlar formal parametrlar dep ataladi', funksiya shaqi'ri'wi'nda ko'rsetilgen argumentlarga faktik parametrlar dep ataladi'.

Funksiya shaqi'rilishida faktik parametrdin' tu'ri saykes wori'ndag'i' formal parametr tu'rine tuwri' kelmasa, kompilyatsiya qa'tesi ju'z beredi.

Faktik parametrlardi funksiyag'a yeki xil usi'l menen uzati'w mu'mkin: ma'nisi yamasa adresi menen.

Funksiya shaqi'rilishida argument ma'nis menen uzatilganda, argument yamasa woni'n' worni'ndag'i' kelgen an'latpa ma'nisi ha'm basqa argumentlerding nusqasi'(ma'nisleri) stek yadi'na jazi'ladi'. Funksiya tek sol nusqalar menen a'mel yetedi, kerek bolsa bul nusqalarg'a wo'zgeriwler yetiliwi mu'mkin, lekin bul wo'zgerisler argumenttin' wo'zine ta'sir yetpeydi ha'm funksiya wo'z jumi'si'n tamamlawi' menen nusqalar wo'shiriledi(stek tazalanadi').

Yeger parametr adres menen uzatilsa, stekka adres nusqasi' jazi'ladi' ha'm tap sol adres boyi'nsha ma'nisler woqi'ladi'(jazi'ladi'). Funksiya wo'z jumi'si'n tamamlag'annan keyin sol adres boyi'nsha yetilgen wo'zgerisler saqlani'p qolinadi ha'm bul ma'nislerdi' basqa funksiyalar isletiw mu'mkin.

Argument ma'nis menen uzati'li'wi' ushi'n saykes formal parametr si'pati'nda wo'zgeriwshin tu'ri ha'm ati' jazi'ladi'. Funksiya shaqi'rilishida saykes argument si'pati'nda wo'zgeriwshinin' ati' yamasa an'latpa boli'wi' mu'mkin.

Faktik parametr adres menen uzatilg'anda wog'an saykes keliwshi formal parametrdi yeki xil usi'l menen jazi'w mu'mkin: ko'rsetkish arqali' yamasa murajat arqali'. Ko'rsetkish arqali' jazi'lg'anda formal parametr tu'rinden keyin '*' belgisi jazi'ladi', saykes argumentda bolsa wo'zgeriwshinin' adresi(& a'mel arqali') yamasa massiv ati', yamasa funksiya ati' boli'wi' mu'mkin. Murajat arqali' parametr uzati'wda formal parametrdi tu'rinden keyin '&' belgisi jazi'ladi' ha'm funksiya shaqi'rilishida saykes argument si'pati'nda wo'zgeriwshi ati' keledi.

Mi'sal:

```
#include <iostream>
using namespace std;
void f(int, int*, int &) ;
int main()
{
    int i=1, j=2, k=3;
    cout<<i<<" "<<j<<" "<<k;
    f(i, &j, k) ;
    cout<<i<<" "<<j<<" "<<k;
```

```

}
void f(int i, int*j, int &k)
{
    i++;
    (*j)++;
    k++;
    *j=i+k;
    k=*j+i;
}

```

Programma islewi na'tiyjesinde ekrang'a to'mendegi ma'nisler baspadan shi'g'ari'ladi':

```

1 2 3
1 6 8

```

Bul mi'salda birinshi parametr I ma'nis menen uzatiladi("int I"). Woni'n' ma'nisi funksiya ishinde wo'zgeredi, lekin si'rtdag'i i ma'nisi wo'zgermeydi. Yekinshi parametrni ko'rsetkish arqali' adresi menen uzati'li'wi' talap yetiledi("int j"), adresni uzati'w ushi'n '&-adresni ali'w a'meli isletilingen("&j"), Funksiya denesinde argument adresinen ma'nis ali'w ushi'n '*-ma'nis ali'w a'meli qollani'lg'an. U'shinshi para-metrde murajat arqali'("&k") argumenttin' adresi uzati'w na'zerde tuti'lg'an. Bul halda funksiya shaqi'rilishida saykes argument worni'nda wo'zgeriwshi ati'turadi', funksiya ishinde bolsa ma'nis ali'w a'melin isletiwdn' ha'jeti joq. Funksiya islew na'tiyjesindegi ma'nislerdi' argumentler dizimi arqali' ali'w qolay ha'm tu'sinikli usi'l yesaplanadi'.

Yeger funksiya ishinde adres menen uzati'latug'i'n parametr ma'nisi turaqli'nan qali'wi' za'ru'r bolsa, bul parametr const modifikator menen jazi'li'wi' kerek:

```

f(int n, const char*str) ;

```

Yegerde funksiyaning shaqi'ri'wda argumentler tek atlari' menen berilgen bolsa, sha'rtlesiw boyi'nsha massivler ha'm funksiylar adresi menen qalg'an tu'rdegi parametrlar ma'nisleri' menen uzatilgan dep yesaplanadi'.

Mi'sal si'pati'nda diskriminantni yesaplaw usi'li' ja'rdeminde $ax^2 + bx + c = 0$ ko'risindegi kvadrat ten'leme korenlerin funksiya parametrlari jardeminde ali'w ma'selesin ko'reylik.

```

#include <iostream>
#include <math.h>
using namespace std;
int Kvadrat_Koren(float a, float b, float c, float & x1,
float &x2)
{
    float D;
    D=b*b-4*a*c;
    if(D<0)
        return 0;
    if(D==0)
    {
        x1=x2=-b/(2*a) ;
        return 1;
    }
    else
    {
        x1=(-b+sqrt(D) ) /(2*a) ;
        x2=(-b-sqrt(D) ) /(2*a) ;
        return 2;
    }
}
int main()
{
    int a, b, c;
    float D, x1, x2;
    cout<<"ax^2+bx+c=0 ten'leme korenin tabi'w.";
    cout<<"\n a-koeffisiyentni kiritin: ";
    cin>>a;
    cout<<"\n b-koeffisiyentni kiritin: ";
    cin>>b;
    cout<<"\n c-koeffisiyentni kiritin: ";
    cin>>c;
    switch(Kvadrat_Koren(a, b, c, x1, x2) )
    {
        case 0:cout<<"Ten'leme haqi'yqi'y koreng e iye
                    yemes!";
                break;
    }
}

```

```

        case 1: cout<<"Ten'leme birden-bir korengi iye: ";
                cout<<"\n x="<<x1;
                break;
        default: cout<<"Ten'leme yeki korengi iye:";
                cout<<"\n x1= "<<x1;
                cout<<"\n x2= "<<x2;
    }
    return 0;
}

```

Programmada `Kvadrat_Koren()` funksiyasi kvadrat ten'leme koreni hisoblaydi. Woni'n qaytaratug'i'n ma'nisi ten'lemenin' neshe koreni bar yekenini an'latadi. Yeger ten'lemenin' haqiqiy koreni bar bo'lmasa ($D < 0$), funksiya 0 ma'nisin qaytaradi. Yeger $D = 0$ bolsa, funksiya 1 ma'nisin qaytaradi. Yeger $D > 0$ bolsa funksiya 2 ma'nisin qaytaradi. Korenlar - x_1 ha'm x_2 murajatli parametrlerde qaytariladi.

Wo'zgeriwshen parametrli funksiyalar: C++ tilinde parametrlar sani' belgisiz bolgan funksiyalardi' da isletiw mu'mkin. Bunnan ti'sqari wolardi'n' tu'rleri da belgisiz bolwi' mu'mkin. Parametrlar sani' ha'm tu'ri funksiyani' shaqiri'wdag'i' argumentlar sani' ha'm wolardi'n' tu'rine qaray ani'qlanadi'. Bunday funksiyalar basi' to'mendegi formatda jaziladi':

<funksiya turi> <funksiya nomi>(<oshkor parametrlar dizimi>, ...)

Bul jerde oshkor parametrlar dizimi-oshkor tu'rde jazilgan parametrlar ati' ha'm tu'ri. Bul parametrlar majburiy parametrlar dep ataladi'. Bunday parametrlardan keminde biewi bolwi' sha'rt. Qalg'an parametrlar sani' ha'm tu'ri belgisiz hisoblanadi'. Wolardi' ani'qlaw ha'm isletiw toliq tu'rde programma du'ziwshi gerdenine yuklenadi.

Wo'zgeriwshen sandag'i' parametrlardi' sho'lkemlestiriw usili' uluwma alg'anda yeki.

Birinshi usulda, parametrlar dizimi aqiri'nda tag'i' bir arnawli' parametr jaziladi' ha'm woni'n' ma'nisi parametrlar tawsi'lg'anlig'i'n an'latadi'. Kompilyator ta'repinen funksiya denesinde parametrlar birmebir ko'rip shig'ili'p parametrlar ani'qlanadi'. Barliq parametrlar tu'ri son'g'i' arnawli' parametr tu'ri menen ustma-u'st tu'sedi dep hisoblanadi'.

Yekinshi usulda qandayda-bir arnawli' parametr si'pati'nda belgisiz parametrlar sani' kiritiledi ha'm wog'an qaray parametrlar sani' ani'qlanadi'.

Yeki usulda da parametrlerge murajat yetiw ushi'n ko'rsetkishler isletiledi. Mi'sallar keltiremiz.

Birinshi usi'l:

```
#include <iostream.h>
float Sanlar_kupaytmasi(float arg, ...)
{
float p=1.0;
float *ptr=&arg;
if(*ptr==0.0) return 0.0;
for(;*ptr;*ptr++) p*=*ptr;
return p;
}
void main()
{
cout<<"\n "<<Sanlar_kupaytmasi(2e0, 3e0, 4e0, 0e0) <<"\n";
cout<<Sanlar_kupaytmasi(1.0, 2.0, 3.0, 10.0, 8.0, 0.0) ;
}
```

Na'tiyje:

24

480

Yekinshi usi'l:

```
#include <iostream.h>
int Yigindi(int, ...) ;
void main()
{
cout<<"\nYigindi(2, 6, 4) =="<<Yigindi(2, 6, 4) ;
cout<<"\nYigindi(6, 1, 2, 3, 4, 5, 6) ="
cout<<Yigindi(6, 1, 2, 3, 4, 5, 6) ;
}
int Yigindi(int k, ...)
{
int *ptr=&k
int s=0;
for(;k;k--) s+=*(++ptr) ;
```

```

    return s;
}

```

Na'tiyje:

```
Yigindi(2, 6, 4) =10
```

```
Yigindi(6, 1, 2, 3, 4, 5, 6) =21
```

Yeki mi'salda da belgisiz parametrlar berilgen arnawli' parametr tu'rin qabil yetken. Har-xil tu'rdegi parametrlardi isletiw ushi'n tu'rdi aniqlaytug'i'n parametr kirgiziw kerek:

```

#include <iostream.h>
float Summa(char, int, ...) ;
void main()
{
    cout<<Summa('i', 3, 10, 20, 30) ;
    cout<<Summa('f', 3, 10.0, 20.0, 5.0) ;
    cout<<Summa('d', 3, 10, 20, 30) ;
}
int Summa(char z, int k, ...)
{
    switch(z)
    {
        case 'i':
        {
            int *ptr=&k+1; int s=0;
            for(;k--;ptr++) s+=*(ptr) ;
            return(float) s;
        }
        case 'f':
        {
            float*ptr=(float*)&k+1 ; float s=0.0;
            for(;k--;ptr++) s+=*(ptr) ;
            return s;
        }
        default:
        {
            cout<<"\n parametr qa'te berilgen";
            return 9999999.0;
        }
    }
}

```

}

Joqari'da keltirilgen mi'salda belgisiz parametrlardi tu'rin ani'qlaw ma'selesi kompilyator ta'repinen yemes, ba'lki programma du'ziwshisi ta'repinen hal yetilgen.

Soraw ha'm tapsi'rmalar

1. *Ko'rsetkish wo'zgeriwshilerge mi'sallar keltiring.*
2. *Ko'rsetkishnin' tu'rleri.*
3. *Funksiyada ko'rsetkish qanday wazi'ypani' atqaradi'?*
4. *Ko'rsetkish arqali' funksiyani' shaqi'ri'w tu'rleri ha'm wolarg'a mi'sallar.*
5. *Malum tu'rdegi(tayansh yamasa tuwi'ndi'li'q) berilgenlerdin' yaddag'i' adresi.*
6. *Obyekt ha'm ko'rsetkishler.*
7. *Ko'rsetkishge ma'nis beriw.*
8. *Dinamikali'q yadqa murajat.*
9. *Ko'rsetkish u'stinde a'meller.*
10. *Murajatlarding isletiliw usi'llari'.*
11. *Funksiya parametrlerinde ko'rsetkishlerdin' isletiliwin mi'sallar arqali' an'latpa berin'.*

Assigning Values Through a Pointer

You can use a pointer on the left side of an assignment statement to assign a value to the location pointed to by the pointer. Assuming that **p** is an integer pointer, this assigns the value 101 to the location pointed to by **p**:

```
*p = 101;
```

You can verbalize this assignment like this: "at the location pointed to by **p**, assign the value 101." To increment or decrement the value at the location pointed to by a pointer, you can use a statement like this:

```
(*p)++;
```

The parentheses are necessary because the ***** operator has lower precedence than the **++** operator.

Pointer Expressions

Pointers can be used in most valid C++ expressions. However, some special rules apply. Remember also that you may need to surround some parts of a pointer expression with parentheses in order to ensure that the outcome is what you desire.

After a pointer is declared, but before it has been assigned a value, it will contain an arbitrary value. Should you try to use the pointer prior to giving it a value, you will probably crash not only your program, but perhaps even the operating system of your computer (a very nasty type of error!). While there is no sure way to avoid using an uninitialized pointer, C++ programmers have adopted a procedure that helps prevent some errors. By convention, if a pointer contains the null (zero) value, it is assumed to point to nothing. Thus, if all unused pointers are given the null value and you avoid the use of a null pointer, you can avoid the accidental misuse of an uninitialized pointer. This is a good practice to follow.

Any type of pointer can be initialized to null when it is declared. For example, the following initializes **p** to null:

```
float *p = 0; // p is now a null pointer
```

To check for a null pointer, use an **if** statement, like one of these:

```
if(p) // succeeds if p is not null
```

```
if(!p) // succeeds if p is null
```

16-Lekciya: Massivler

Joba:

1. Berilgenler massivi tu'sinigi
2. Ko'p wo'lshemli statik massivler
3. Ko'p wo'lshemli massivlerdi initsializatsiyalash
4. Dinamikali'q massivler menen islew
5. Funktsiya ha'm massivler

Tayansh so'zler: *Berilgenler massivi tu'sinigi, ko'p wo'lshemli statikali'q massivler, ko'p wo'lshemli massivlerdi initsializatsiyalash, dinamikali'q massivler menen islew, funksiya ha'm massivler*

Berilgenler massivi tu'sinigi: Yadta izbe-iz (regular) joylasqan birdey tu'rdegi ma'nislerge massiv dep ataladi'.

A'detde massivlerge za'ru'rat, u'lken hajmdagi, lekin sheklengen mug'dardag'i' ha'm ta'rtiplengen ma'nislerdi' qayta islew menen baylani'sli' ma'selelerdi sheshiwde ju'zege keledi. Shama menen woylayi'q, studentler topari'ni'n' reyting zi'yapatlari' menen islew ma'sese qoyi'lg'an. Wol jag'dayda gruppani'n' wortasha reytingin ani'qlaw, reytinglerdi azayi'wi' boyi'nsha ta'rtiplaw, konkret studenttin' reytingi haqqi'nda si'patlama beriw ha'm basqa ma'sele asti'lari'n sheshiw za'ru'r bo'lsin. Belgilengenler yetilgen ma'selelerdi sheshiw ushi'n berilgenlerdin'(reytinglerdin') ta'rtiplengen izbe-izligi za'ru'r boladi'. Bul jerde ta'rtiplengenlik ma'nisi sonda, izbe-izliktin' ha'r bir ma'nisi wo'z worni'na iye boladi'(birinshi studenttin' reytingi massivde birinshi wori'nda, yekinshi studentniki-yekinshi wori'nda ha'm hakoza). Berilgenler izbe-izligin yeki xil usulda payda yetiw mu'mkin. Birinshi jol-ha'r bir reyting ushi'n bo'lek wo'zgeriwshi ani'qlaw: Reyting₁, Reyting₂, ... Reyting_N. Lekin, gruppada'gi' studentler sani' jetkiliklishe u'lken bolg'anda, bul wo'zgeriwshiler qatnasqan programmani du'ziw u'lken qi'yi'nshi'li'qlardi' ju'zege keltiredi. Yekinshi jol-berilgenler izbe-izligin birden-bir at menen ani'qlap, woni'n' ma'nisleri'na murajatdi', sol ma'nislerdi'n' izbe-izlilikde

jaylasqan worni'nn' nomeri(indeksi) arqali' a'melge asi'ri'w boli'p tabi'ladi'. Reytingler izbe-izligin Reyting dep nomlab, wodag'i' ma'nisleri'na Reyting 1 , Reyting 2 , ... Reyting N ko'riniside murajat yetiw mu'mkin. A'detde berilgenlerdin' bunday ko'rinisine massivler dep ataladi'.

Massivlerdi matematikadagi sanlar vektori'na o'xshatish mu'mkin, sebebi vektor da wo'zinin' individual ati'na iye ha'm wol fiksirlengen mug'dardag'i' bir tu'rdegi ma'nislerdan-sanlardan ibarat yesaplanadi.

Sunday yeken, massiv-bul fiksirlengen mug'dardag'i' ayi'ri'm ma'nislerdi'n'(massiv elementlerinin') ta'rtpiengen kompleksi boli'p tabi'ladi'. Barli'q elementler birdey tu'rde boli'wi' kerek ha'm bul tu'r element tu'ri yamasa massiv ushi'n tayansh tu'r dep ataladi'. Joqari'dag'i' keltirilgen mi'salda Reyting-haqi'yqi'y tu'rdegi vektor dep ataladi'.

Programmada isletiletug'i'n ha'r bir konkret massiv wo'zinin' individual ati'na i'yelewi kerek. Bul atdi' toli'q wo'zgeriwshi dep ataladi', sebebi woni'n' ma'nisi toli'q bir massivtin' wo'zi boladi'. Massivtin' ha'r bir elementi massiv ati', ha'mde kvadrat qavsga ali'ng'an ha'm element selektori dep atali'wshi indeksni ko'rsetiw arqali' oshkor tu'rde belgilenedi. Murajat sintaksisi:

`<massiv ati' >[<indeks>]`

Bul ko'rinishke jeke wo'zgeriwshi dep ataladi', sebebi woni'n' ma'nisi massivtin' bo'lek elementi boli'p tabi'ladi', Bizin' mi'salda Reyting massivinin' bo'lek komponentalariga Reyting[1], Reyting[2],..., Reyting[N] jeke wo'zgeriwshiler arqali' murajat yetiw mu'mkin. Basqasha bul wo'zgeriwshilerdi indeksli wo'zgeriwshiler dep ataladi'.

Uluwma alg'anda indeks si'pati'nda an'latpa isletiliwi mu'mkin. An'latpa ma'nisi massiv elementi nomerini ani'qlaydi'. An'latpa si'pati'nda wo'zgeriwshi da ali'ni'wi' mu'mkin, wo'zgeriwshinin' ma'nisi wo'zgeriwi menen murajat qi'li'ni'p ati'rg'an massiv elementin ani'qlawshi' indeks da wo'zgeredi. Sunday yetip, programmadagi bir indeksli wo'zgeriwshi arqali' massivtin' barli'q elementlerin belgilew(ani'qlaw) mu'mkin. Mi'sali', Reyting[1] wo'zgeriwshisi arqali' I wo'zgeriwshinin' ma'nisine baylani'sli' tu'rde Reyting massivinin' ha'r qi'yli'(barli'q) elementlerine murajat yetiw

mu'mkinshiligi bar. Soni' dizimnen wo'tkeriw kerek, massiv indeks si'pati'nda pu'tin san qollani'ladi'.

Haqi'yqi'y tu'rdegi (float, double) ma'nisler kompleksi sheksiz bolg'anli'g'i' sebepli wolar indeks si'pati'nda isletilmaydi.

C++ tilinde indeks mudami 0 den baslanadi', woni'n' yen' u'lken ma'nisi massiv dag'azasi'ndag'i' uzi'nli'qdan birge kem boladi'.

Massiv dag'azasi' to'mendegishe boladi':

```
<tur> <at> [<uzi'nlik>]={ baslang'ish ma'nisler}
```

Bul jerde <uzi'nliq> - turaqli' an'latpa. Mi'sallar:

```
int m[6]={1, 4, -5, 2, 10, 3} ;  
float a[4] ;
```

Massiv statik ha'm dinamikali'q boli'wi' mu'mkin, Statik massivtin' uzi'nli'g'i' aldi'nan ma'lum boli'p, wol yadta ma'lum adresden baslap izbe-iz jaylasadi'. Dinamikali'q massivti uzi'nli'g'i' programma atqari'li'w processinde ani'qlani'p, wol dinamikali'q yaddag'i' usi' waqi'tta bos bolg'an adreslerge jaylasadi'. Mi'sali',

```
int m[6] ;
```

ko'riniside dag'aza yetilgen bir wo'lshemli massiv elementleri yadta to'mendegishe jaylasadi':

Ma'nisler

adres	m[0]	m[1]	m[2]	m[3]	m[4]	m[5]

7.1-su'wret. Bir wo'lshemli massivtin' yaddag'i' jaylasi'wi'

Massivtin' i-elementine m[i] yamasa *(m+i) -quralli' murajat yetiw mu'mkin. Massiv uzi'nli'g'i'n sizeof(m) a'meli arqali' ani'qladi.

Massiv dag'azasi'nda woni'n' elementlerine baslang'i'sh ma'nisler ali'w (initsializatsiyalash) mu'mkin ha'm woni'n' bir neshe variantlari' bar.

1) Wo'lshemi ko'rsetilgen massiv elementlerin toli'q initsializatsiyalash:

```
int t[5]={-10, 5, 15, 4, 3};
```

Bunda 5 dana elementden ibarat bolg'an t atli' bir wo'lshemli massiv dag'aza yetilgen ha'm woni'n' barli'q elementlerine baslang'i'sh ma'nisler berilgen. Bul dag'aza to'mendegi dag'aza menen ekvivalent:

```
int t[5];
t[0]=-10; t[1]=5; t[2]=15; t[3]=4; t[4]=3;
```

2) Wo'lshemi ko'rsetilgen massiv elementlerin toli'qmas initsializatsiyalash:

```
int t[5]={-10, 5, 15};
```

Bul jerde tek massiv basi'daga ush elementke baslang'i'sh ma'nisler berilgen. Soni' ayti'p wo'tiw kerek, massivtin' basi'ndag'i' yamasa wortadag'i' ementlariga ma'nisler bermesten, woni'n' aqi'ri'ndag'i' elementlerge baslang'i'sh ma'nis beriw mu'mkin yemes. Yegerde massiv elementlerine baslang'i'sh ma'nis berilmasa, wol jag'dayda sha'rtlesiw boyi'nsha static ha'm extern modifikatori menen dag'aza yetilgen massiv ushi'n elementlerinin' ma'nisi 0 sani'na ten' dep, automatic massivler elementlerinin' baslang'i'sh ma'nisleri' belgisiz yesaplanadi'.

3) Wo'lshemi ko'rsatilmagan massiv elementlerin toli'q initsializatsiyalash:

```
int t[]={-10, 5, 15, 4, 3};
```

Bul mi'salda massivti barli'q elementlerine ma'nisler berilgen yesaplanadi', massiv uzi'nli'g'i' kompilyator ta'repinen baslang'i'sh ma'nisler sani'na qaray ani'qlanadi'. Yegerde massivti uzi'nli'g'i' berilmasa, baslang'i'sh ma'nisi beriliwi sha'rt.

Massivti ja'riyalaw mi'sallari':

```
char ch[4]={ 'a', 'b', 'c', 'd'}; //belgiler massivi
int i'n[6]={10, 20, 30, 40}; // pu'tin sanlar massivi
char str[]="abcd";//qatar uzi'nli'g'i' 5 ge ten', sebebi
//onin' aqi'ri'na '\0' belgisi qosi'ladi'
char str[]={ 'a', 'b', 'c', 'd'}; // joqari'dag'i'
qatardi'n'basqasha jazi'li'wi'
```

Ma'sele. Bir ay ishindeg'i' ku'ndelik temperaturasi' berilgen, ay ushi'n wortasha temperaturani' yesaplaw programmasi tuzilsin.

Programma teksti:

```
#include <iostream>
using namespace std;
int main()
{
    const int n=30;
    int temp[n];
    int i, s=0, temp_ortasha;
    cout<<"Ku'nlik temperaturani' kiritin':\n";
```

```

for(i=0; i<n; i++)
{
    cout<<"\n temp["<<i<<"]=";
    cin>>temp[i];
}
for(i=0; i<n; i++)
    s+=temp[i];
temp_ortasha=s/n;
cout<<"Ku'nlik temperatura:\n";
for(i=0; i<n; i++)
    cout<<"\temp["<<i<<"]="<<temp[i];

    cout<<"\n Aydag'i' wortasha temperatura =
"<<temp_ortasha;
    return 0;
}

```

Ko'p wo'lshemli statik massivler: C++ tilinde massivler elementinin' tu'rine cheklovlar qoyi'lmaydi', lekin bul tu'rler chekli wo'lshemdegi obyektlarding tu'ri boli'wi' kerek. Sebebi kompilyator massivtin' yaddan qansha jay(bayt) iyelewin yesaplay ali'wi' kerek. Atap aytqanda, massiv komponentasi massiv boli'wi' mu'mkin, yag'ni'y «vektorlar-vektori» na'tiyjede matritsa dep atali'wshi yeki wo'lshemli massiv payda boladi'.

Yeger matritsanin' elementi da vektor bolsa, u'sh wo'lshemli massivler-kub payda boladi'. Sol jol menen yechilayotgan ma'selege baylani'sli' tu'rde qa'legen wo'lshemdegi massivlerdi jarati'w mu'mkin. Yeki wo'lshemli massivtin' sintaksisi to'mendegi ko'riniste boladi':

```
<tu'r> <at> [<uzi'nli'q >] [<uzi'nli'q>]
```

Mi'sali', 10 x 20 wo'lshemli haqi'yqi'y sanlar massivinin' dag'azasi':

```
float A[10] [20] ;
```

Dag'aza yetilgen A matritsani ko'rinisi 7.2-suwretde keltirilgen.

$$a_0:(a_{00}, a_{02} \dots \dots a_{018}, a_{019}),$$

$a_1:(a_{10}, a_{11} \dots \dots a_{118}, a_{119} \dots),$
 ...
 $a_i:(\dots, \dots, \dots a_{ij} \dots, \dots, \dots),$
 ...
 $a_9:(a_{90}, a_{91} \dots \dots a_{918} \dots, a_{919} \dots).$

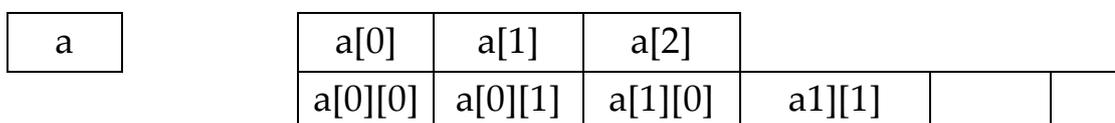
7.2-su'wret. Yeki wo'lshemli massivtin' yaddag'i' jaylasi'wi'

Yendi adres ko'z qarasi'nan ko'p wo'lshemli massiv elementlerine murajat yetiwdi ko'reylik. To'mendegi dag'azalar berilgen bo'lsin:

```
int a[3][2];
float b[2][2][2];
```

Birinshi dag'azada yeki wo'lshemli massiv, yag'ni'y 2 qatar ha'm 3 u'stinnen ibarat matritsa elon yetilgen, yekinshisinde u'sh wo'lshemli-3 dana 2x2 magritsadan ibarat bolg'an massiv dag'aza qlingan. Woni'n' elementlerine murajat sxemasi':

Ades Ko'rsetkishler massivi Ma'nisler



7.3 su'wret. Yeki wo'lshemli elementlerine murajat

Bul jerde a[i] ko'rsetkishde i-chi qatardi'n' baslang'i'sh adresi jaylasadi', massiv elementine a[i][j] ko'rinisindegi tiykarg'i' murajatchan ti'sqari quralli' murajat yetiw mu'mkin: *((a+i) +j) yamasa *(a[j]+j).

Ko'rsetkishler massivi

Ma'nisler

7.3-su'wret. U'sh wo'lshemli massivtin' yadta tashkil boli'wi'

Massiv elementlerine murajat yetiw ushi'n nomdan keyin kvadrat qavsda ha'r bir wo'lshem ushi'n indeks jazi'li'wi' kerek, mi'sali' b[i][j][k]. Bul elementke quralli' murajat da yetiw mu'mkin ha'm woni'n' variantlari':

(((b+i) +j) +) k) yamasa *((b[i]+j) +k) yamasa *(b[i][j] +k) ;

Ko'p wo'lshemli massivlerdi initsializatsiyalash: Massivlerdi initsializatsiyalash to'mendegi mi'sallarda ko'rsetilgen:

```
int a[2][3]={0, 1, 2, 10, 11, 12};
int b[3][3]={{0, 1, 2}, {10, 11, 12}, {20, 21, 22}} ;
int c[3][3][3]={{0}}, {{100, 101}, {110}}, {{200, 201,
202}, {210, 211, 212}, {220, 221, 222}};
```

Birinshi operatorida baslang'i'sh ma'nisler izbe-iz jazi'lg'an, yekinshi operatorida ma'nisler gruppallasqan, u'shinshi operatorida da gruppallasqan, lekin birpara gruppalarda son'g'i' ma'nisler berilmagan.

Mi'sal ushi'n, matritsalar ha'm vektor ko'beymesin $C=A*b$ yesaplaw ma'selesin ko'reylik. Bul jerde $A = \{a^{ij}\}$, $b = \{b^j\}$, $c = \{c^i\}$, $i = 0..m-1$.

Yesaplaw formulasi' $c_i = \sum_{j=0}^{n-1} a_{ij}b_j$

Programa teksti:

```
#include <iostream>
using namespace std;
int main()
{
const int n=3, m=3;
float a[m] [n], b[n], c[m];
int i, j;
float s;
for(i=0; i<m; i++)
for(j=0; j<n; j++)
    cin>>a[i][j];

for(i=0; i<m; i++)
    cin>>b[i];

for(i=0; i<m; i++)
{
    s=0;
    for(j=0; j<n; j++)
        s+=a[i][j]*b[j];
    c[i]=s;
}
```

```

for(i=0; i<m; i++)
    cout<<"\t c["<<i<<"]="<<c[i];
return 0;
}

```

Dinamikali'q massivler menen islew: Statistikalı'q massivlerdin' kemshilikleri sonda, woları'n' wo'lshemi aldi'nan ma'lum boli'wi' kerek, wodan ti'sqari' bul wo'lshem berilgenlerge aji'rati'lg'an yad segmentinin' wo'lshemi menen shegaralang'an. Yekinshi ta'repden, jetkiliklishe u'lken wo'lshemdegi massiv ja'riyalay otirip, konkret ma'sele sheshiliwida aji'rati'lg'an yad toli'q isletilmasligi mu'mkin. Bul kemshilikler dinamikali'q massivlerden paydalani'w arqali' jo'nge sali'w yetiledi, sebebi wolar programma islewi processinde za'ru'r bolg'anda kerek wo'lshemdegi massivlerdi jarati'w ha'm za'ru'rat qolmaganda joyti'w mu'mkinshiligin beredi.

Dinamikali'q massivlerge yad aji'rati'w ushi'n malloc(), calloc() funksiyalari'nan yamasa new operatori'nan paydalani'w mu'mkin. Dinamikali'q obyektga aji'rati'lg'an yadni' bosatish ushi'n free() funksiyasi' yamasa delete operatori' isletiledi.

Joqari'da belgilengenler yetilgen funksiyalar alloc.h kitapxanası'nda jaylasqan.

```

malloc() funksiyasi'ni'n' sintaksisi
void * malloc(size_t size) ;

```

ko'rinisinde boli'p, wol yadni'n' u'yin bo'legiden size bayt wo'lshemindegi u'zliksiz tarawdi' ajratadi'. Yeger yad aji'rati'w tabi'sli' bolsa, malloc() funksiyasi' sol tarawdi'n' baslani'wi'ni'n' adresin qaytaradi'. Talap yetilgen yadni' aji'rati'w a'wmetsiz bolsa, funksiya NULL ma'nisin qaytaradi'.

Sintaksidan ko'rinip turi'pti', wolda, funksiya void tu'rindegi ma'nis qaytaradi'. A'melde bolsa konkret tu'rdegi obyekt ushi'n yad aji'rati'w za'ru'r boladi'. Woni'n' ushi'n void tu'rin konkret tu'rge keltiriw texnologiyasi'nan paydalani'ladi'. Mi'sali', pu'tin tu'rdegi uzi'nlig'i 3 ge ten' massivke jay aji'rati'wdi' to'mendegishe a'melge asi'ri'w mu'mkin:

```

int * pInt=(int*) malloc(3*sizeof(int) ) ;

```

malloc() funksiyasi'nan ayri'qsha tu'rde calloc() funksiyasi' massiv ushi'n jay aji'rati'wdan ti'sqari' massiv elementlerin 0 ma'nisi menen initsializatsiya yetedi. Bul funksiya sintaksisi

```
void * calloc(size_t num, size_t size) ;
```

ko'riniste boli'p, num parametri aji'rati'lg'an tarawda neshe element bar yekenini, size ha'r bir element wo'lshemin an'latadi'.

free() yadni' bosatish funksiyasi' wo'shiriletug'i'n yad bo'legine ko'rsetkish bolg'an birden-bir parametrga iye boladi':

```
void free(void * blok)
```

free() funksiyasi' parametrinin' void tu'rinde boli'wi' qa'legen tu'rdegi yad bo'legini wo'shiriw imkaniyati'n beredi.

To'mendegi programmada 10 ta pu'tin sandan ibarat dinamikali'q massiv jarati'w, wog'an ma'nis beriw ha'm wo'shiriw a'melleri wori'nlang'an.

```
#include <iostream>
#include <alloc.h>
using namespace std;
int main()
{
    int * pVector;
    if((pVector=(int*) malloc(10*sizeof(int) ) )==NULL)
    {
        cout<<"Yad jetkilikli yemes!!!";
        return 1;
    }
    // aji'rati'lg'an yad ken'isligin tolti'ri'w
    for(int i=0; i<10; i++)
        *(pVector+i) =i;
    // vektor elementlerin baspadan shi'g'ari'w
    for(int i=0; i<10; i++)
        cout<<*(pVector+i) <<"endl";
    // aji'rati'lg'an yad bo'legini qaytari'w(wo'shiriw)
    free(pVector) ;
    return 0;
}
```

Keyingi programmada nxn wo'lshefli haqi'yqi'y sanlar massivinin' bas qiyiqidan joqari'da jaylasqan elementler ji'yi'ndi'si'n yesaplaw ma'selesi sheshilgen.

```
#include <iostream>
#include <alloc.h>
using namespace std;
int main()
{
    int n ;
    float * pMatr, s=0;
    cout<<"A(n, n) : n=";
    cin>>n;
    if((pMatr=(float*) malloc(n*n*sizeof(float) ) )==NULL)
    {
        cout<< "Yad jetkilikli yemes!!!";
        return 1;
    }
    for(int i=0; i<n; i++)
        for(int j=0; j<n; j++)
            cin>>*(pMatr+i*n+j) ;

    for(int i=0; i<n; i++)
        for(int j=i+1; j<n; j++)
            s+=*(pMatr+i*n+j);

    cout<<"Matritsa bas diaonalidan joqari'dag'i";
    cout<<"elementler ji'yi'ndi'si' S="<<s<<"endl";
    return 0;
}
```

new operatori' ja'rdeminde, massivke yad aji'rati'wda. Obyekt tu'rinden keyin kvadrat qawi's ishinde obyektlar sani' ko'rsetiledi. Mi'sali', pu'tin tu'rdegi 10 ta sandan ibarat massivke jay aji'rati'w ushi'n

```
pVector=new int[10];
```

an'latpasi' jazi'li'wi' kerek. Bug'an keru tu'rde, bul usulda aji'rati'lg'an yadni' bosatish ushi'n

```
delete []pVector;
```

ko'rsetpesin beriw kerek boladi'.

Yeki wo'lshemli dinamikali'q massivti sho'lkemlestiriv ushi'n

```
int **a;
```

ko'rinishidagi «ko'rsatkichga ko'rsetkish» isletiledi.

Boshda massiv qatarlari' sani'na qaray ko'rsetkishler massivine dinamikali'q yaddan jay aji'rati'w kerek:

```
a=new int *[m] // bul jerde m massiv qatarlari' sani'
```

Keyin, ha'r bir qatar ushi'n ta'kirarlaw operatori' ja'rdeminde yad aji'rati'w ha'm wolardi'n' baslang'i'sh adreslerin a massiv elementlerine jaylasti'ri'w za'ru'r boladi':

```
for(int i=0; i<m; i++)
```

```
    a[i]=new int[n]; //n u'stinler sani'
```

Soni' atap ko'rsetiw za'ru'r, dinamikali'q massivtin' ha'r bir qatari' yadni'n' ha'r qi'yli' jaylari'nda jaylasi'wi' mu'mkin(7.1 ha'm 7.3-su'wretler).

Yeki wo'lshemli massivti wo'shiriwde aldi'n massivtin' ha'r bir elementi(qatari'), keyininen massivtin' wo'zi joyti'ladi':

```
for(i=0;i<m;i++)
```

```
    delete [] a[i] ;
```

```
delete[] a;
```

Matritsani vektorg'a ko'beytiw ma'selesi ushi'n dinamikali'q massivlerden paydalani'wg'a mi'sal:

```
#include <iostream>
```

```
#include <alloc.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n, m;
```

```
    int i, j; float s;
```

```
    cout<<"\n n=";>>n; // matritsa qatarlari' sani'
```

```
    cout<<"\n m=";>>m; // matritsa u'stinleri sani'
```

```
    float *b=new float[m];
```

```
    float *c=new float[n];
```

```
    // ko'rsadakichlar massivine yad aji'rati'w
```

```
    float **a=new float *[n];
```

```
    for(int i=0; i<n; i++) //har bir qatar ushi'n
```

```
        a[i]=new float[m]; //dinamikali'q yad aji'rati'w
```

```
    for(j=0; j<m; j++)
```

```

        cin>>b[j];
    for(i=0; i<m; i++)
        for(j=0;j<m;j++)
            cin>>a[i, j];
    for(i=0;i<n;i++)
    {
        for(j=0, s=0; j<m; j++)
            s+=a[i][j]*b[j];
        c[i]=s;
    }
    for(i=0; i<n; i++)
        cout<<"\t c["<<i<<"]="<<c[i];
    delete[]b;
    delete[]c;
    for(i=0; i<n; i++)
        delete[] a[i];
    delete[] a;
    return 0;
}

```

Funksiya ha'm massivler: Funksiyalar massivni parametr si'pati'nda isletiw ha'm woni' funksiyani'n' na'tiyjesi ko'riniside qaytari'wi mu'mkin.

Yeger massiv parametr arqali' funksiyag'a uzatilsa, elementler sani'n ani'qlaw mashqalasi' tuwi'ladi', sebebi massiv ati'nan woni'n' uzi'nli'g'i'n ani'qlawdi'n' i'laji' joq. Ayi'ri'm jag'daylarda, mi'sali', belgiler massivi si'pati'nda ani'qlang'an qatar (ASCIIZ qatarlar) menen islegende massiv uzi'nli'g'i'n ani'qlaw mu'mkin, sebebi qatarlar '\0' belgisi menen tawsi'ladi'(8-bobga qarang).

Mi'sal ushi'n:

```

#include <iostream>
using namespace std;
int len(char s[]) //massivni parametr si'pati'nda isletiw
{
    int m=0;
    while(s[m++]);
    return m-1;
}

```

```
int main()
{
    char z[]="Usi' qatar uzi'nli'g'i' ";
    cout<<z<<len(z) ;
}
```

Funksiya parametri qatar bolmag'an jag'daylarda fiksirlengen uzi'nli'qdag'i' massivler isletiledi. Yeger ha'r qi'yli' uzi'nli'qdag'i' massivlerdi uzati'w za'ru'r bolsa, massiv wo'lshemlerin parametr si'pati'nda uzati'w mu'mkin yamasa bul maqsette global wo'zgeriwshiden paydalani'wg'a tuwri' keledi.

Mi'sal:

```
#include <iostream>
using namespace std;
float sum(int n, float *x) //bul yekinshi usi'l
{
    float s=0;
    for(int i=0; i<n;i++)
        s+=x[i];
    return s;
}
int main()
{
    float E[]={1.2, 2.0, 3.0, 4.5, -4.0};
    cout<<sum(5, E);
}
```

Massiv ati' ko'rsetkish bolg'anli'g'i' sebepli massiv elementlerin funksiyada wo'zgertiw mu'mkin ha'm bul wo'zgertiwler funksiyadan shi'qqannan keyin da saqlani'p qaladi'.

```
#include <iostream>
using namespace std;
void vector_01(int n, int*x, int*y) //bul yekinshi usi'l
{
    for(int i=0; i<n; i++)
        y[i]=x[i]>0?1:0;
}
```

```

int main()
{
    int a[] = {1, 2, -4, 3, -5, 0, 4};
    int c[7];
    vector_01(7, a, c) ;
    for(int i=0; i<7; i++)
        cout<<" "<<c[i];
}

```

Ma'sele. Pu'tin tu'rdegi ha'm elementleri kamaymaydigan halda ta'rtilplengen bir wo'lshemli yeki massivlerdi bir massivke, ta'rtilpleniw saqlang'an halda birlestiriw a'melge asi'rilsin.

Programma teksti:

```

#include <iostream>
using namespace std;
//butun tu'rdegi massivke ko'rsetkish qaytaratug'i'n
funksiya
int *massiv_ulasht(int, int*, int, int*) ;
int main()
{
    int c[]={-1, 2, 5, 10}, d[]={1, 7, 8};
    int *h;
    h=massiv_ulasht(5, c, 3, d) ;
    for(int i=0; i<7; i++)
        cout<<" "<<h[i];
    delete[]h;
}
int *massiv_ulasht(int n, int *a, int m, int *b)
{
    int *x=new int[n+m];
    int ia=0, ib=0, ix=0;
    while(ia<n && ib<m)
        a[ia]>b[ib] ? x[ix++]=b[ib++] : x[ix++]=a[ia++];

    while(ib<m)
        x[ix++]=b[ib++];

    while(ia<n)

```

```

        x[ix++]=a[ia++];
    return x;
}

```

Ko'p wo'lshemli massivler menen islew ma'lum bir quramali'li'qqa iye, sebebi massivler yadta jaylasiw rejimi ha'r qi'yli' variantda boli'wi' mu'mkin. Mi'sali', funksiya parametrlar diziminde $n \times n$ wo'lshemdegi haqi'yqi'y tu'rdegi $x[n][n]$ massivke saykes keliwshi parametrni

```
float sum(float x[n][n])
```

ko'riniste jazi'p bolmaydi'. Mashqala sheshimi - bul massiv wo'lshemi parametr si'pati'nda uzati'w ha'm funksiya ani'qlaması'n to'mendegishe jazi'w kerek:

```
float sum(int n, float x[][]);
```

Ko'p wo'lshemli massivlerdi parametr si'pati'nda isletiwde bir qansha usi'llardan paydalani'w mu'mkin.

1-usi'l. Massivtin' yekinshi wo'lshemin turaqli' an'latpa (san) menen ko'rsetiw:

```
float sum(int n, float x[][10])
{
    float s=0.0;
    for(int i=0; i<n;i++)
        for(int j=0; j<n;j++)
            s+=x[i][j];
    return s;
}

```

2-usi'l. Yeki wo'lshemli massiv ko'rsetkishler massivi ko'rinisinde ani'qlang'an jag'daylar ushi'n ko'rsetkishler massivin(matrisa qatarlar adreslerin) beriw arqali':

```
float sum(int, float *p[])
{
    float s=0.0;
    for(int i=0;i<n;i++)

```

```

        for(int j=0;j<n;j++)
            s+=p[i][j]; //p[i][j] yemes, sebebi massivke
        murajat
    return s;
}
int main()
{
    float *x[][4]={
        {11, 12, 13, 14},
        {21, 22, 23, 24},
        {31, 32, 33, 34},
        {41, 42, 43, 44}
    };
    float *ptr[4];
    for(int=0;i<4;i++)
        ptr[i]=(float*) &x[i];
    cout<<sum(4, ptr) <<endl;
}

```

3-usi'l. Ko'rsetkishlarga ko'rsetkish ko'riniside ani'qlang'an dinamikali'q massivlerdi isletiw menen:

```

float sum(int n, float **x)
{
    float s=0.0;
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)
            s+=x[i][j];
    return s;
}
int main()
{
    float **ptr;
    int n;
    cin>>n;
    ptr=new float *[n];
    for(int i=0;i<n;i++)
    {
        ptr[i]=new float[n]
    }
}

```

```

        for(int j=0;j<n; j++)
            ptr[i][j]=(float)((i+1) *10+j) ;
    }
    cout<<sum(n, ptr) ;
    for(int i=0;i<n;i++)
        delete ptr[i];
    delete[]ptr;
}

```

Na'wbettegi programmada funksiya ta'repinen yeki wo'lshemli massivti na'tiyje si'pati'nda qaytari'wi'na mi'sal keltirilgen. Massiv elementlerdin' ma'nisleri' tosi'narli' sanlardan tashkil tabadi'. Tosi'narli' sanlar «math.h» kitapxanasini'ndagi' random() funksiya ja'rdeminde payda yetiledi:

```

#include <iostream>
#include <cstdlib>
#include <math.h>
#include <time.h>
using namespace std;
int **rmatr(int n, int m)
{
    int **ptr;
    ptr=new int *[n];
    for(int i=0;i<n;i++)
    {
        ptr[i]=new int[m];
        for(int j=0;j<m;j++)
            ptr[i][j]=rand()%100 ;
    }
    return ptr;
}
int sum(int n, int m, int **ix)
{
    float s=0;
    for(int i=0;i<n;i++)
        for(int j=0;j<m;j++)
            s+=ix[i][j];
    return s;
}

```

```

}
int main()
{
    int n, m;
    cin>>n>>m;
    int **matr;
    srand (time(NULL));
    matr=rmatr(n, m) ;
    for(int i=0;i<n;i++)
    {
        cout<<endl<<i<<"-qatar:";
        for(int j=0;j<m;j++)
            cout<<'\\t'<<matr[i][j];
    }
    cout<<endl<<"Summa="<<sum(n, m, matr) ;
    for(int i=0;i<n;i++)
        delete matr[i];
    delete[] matr;
}

```

Soraw ha'm tapsi'rmalar

1. *Massiv haqqi'nda tu'sinik.*
2. *Statikali'q ha'm dinamikali'q massivler haqqi'nda mag'li'wmat.*
3. *Ko'p wo'lshewli statik massivler.*
4. *Dinamikali'q massivlerge hotiradan jay aji'rati'w qanday a'melge asi'ri'ladi'?*
5. *Massiv tu'rleri.*
6. *Massiv dag'azasi'nda woni'n' elementlerine baslang'i'sh ma'nis beriw usi'llari'.*
7. *Dinamikali'q massivlerge yad aji'rati'w qanday a'melge asi'ri'ladi'?*
8. *Massivti funksiya parametri si'pati'nda isletiw.*

One-Dimensional Arrays

A one-dimensional array is a list of related variables. The general form of a one-dimensional array declaration is

```
type var_name[size];
```

Here, *type* declares the base type of the array. The base type determines the data type of each element that comprises the array. *size* defines how many elements the array will hold. For example, the following declares an integer array named **sample** that is ten elements long:

```
int sample[10];
```

In C++, all arrays consist of contiguous memory locations. (That is, the array elements reside next to each other in memory.) The lowest address corresponds to the first element, and the highest address to the last element. For example, after this fragment is run,

```
int i[7];
int j;

for(j=0; j<7; j++) i[j] = j;
```

i looks like this:

i[0]	i[1]	i[2]	i[3]	i[4]	i[5]	i[6]
0	1	2	3	4	5	6

Two-Dimensional Arrays

C++ allows multidimensional arrays. The simplest form of the multidimensional array is the two-dimensional array. A two-dimensional array is, in essence, a list of one-dimensional arrays. To declare a two-dimensional integer array **twod** of size 10,20 you would write

```
int twod[10][20];
```

Pay careful attention to the declaration. Unlike some other computer languages, which use commas to separate the array dimensions, C++ places each dimension in its own set of brackets.

Similarly, to access point 3,5 of array **twod**, you would use **twod[3][5]**. In the next example, a two-dimensional array is loaded with the numbers 1 through 12.

Multidimensional Arrays

C++ allows arrays with more than two dimensions. Here is the general form of a multidimensional array declaration:

```
type name[size1][size2]...[sizeN];
```

For example, the following declaration creates a $4 \times 10 \times 3$ integer array:

```
int multidim[4][10][3];
```

Arrays of more than three dimensions are not often used, due to the amount of memory required to hold them. As stated before, storage for all array elements is allocated during the entire lifetime of an array. When multidimensional arrays are used, large amounts of memory can be consumed. For example, a four-dimensional character array with dimensions 10,6,9,4 would require $10 \times 6 \times 9 \times 4$ (or 2,160) bytes. If each array dimension is increased by a factor of 10 each (that is, $100 \times 60 \times 90 \times 40$), then the memory required for the array increases to 21,600,000 bytes! As you can see, large multidimensional arrays may cause a shortage of memory for other parts of your program. Thus, a program with arrays of more than two or three dimensions may find itself quickly out of memory!

Array Initialization

C++ allows the initialization of arrays. The general form of array initialization is similar to that of other variables, as shown here:

```
type-specifier array_name[size] = {value-list};
```

The *value-list* is a comma-separated list of values that are type-compatible with the base type of the array. The first value will be placed in the first position of the array, the second value in the second position, and so on. Notice that a semicolon follows the `}`.

In the following example, a 10-element integer array is initialized with the numbers 1 through 10:

```
int i[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
```

This means that **`i[0]`** will have the value 1, and **`i[9]`** will have the value 10.

17-Lekciya: Qatarlar

Joba:

1. Belgi ha'm qatarlar
2. Qatar uzi'nli'g'i'n ani'qlaw funktsiyalari'
3. Qatarlardi' nusqalaw
4. Qatarlardi' jalg'aw

Tayansh so'zler: *Belgiler, qatarlar, belgi klasslari', belgi turaqli'lar, qatar uzi'nli'g'i'n ani'qlaw funktsiyalari', qatarlardi' nusqalaw, satrlardi jalg'aw*

Belgi ha'm qatarlar: Standart C++ tili yeki tu'rdegi belgiler kompleksin qollap quwatlaydi'. Birinshi kategoriyag'a, da'stu'riy, «tar» belgiler dep atali'wshi 8-bitli belgiler kompleksi kiredi, yekinshisine 16-bitli «ken'» belgiler kiredi. Til kitapxanasida ha'r bir gruppaga belgileri ushi'n arnawli' funktsiyalar kompleksi ani'qlang'an.

C++ tilinde qatar ushi'n arnawli' tu'r ani'qlanbag'an. Qatar char tu'rindegi belgiler massivi si'pati'nda qaraladi ha'm bul belgiler izbe-izligi qatar terminatori dep atali'wshi 0 kodli' belgi menen tawsi'ladi('\'0'). A'detde, nol-terminator menen tawsi'latug'i'n qatarlardi' ASSIIZ-satlar deyiledi.

To'mendegi kestedeki C++ tilinde belgi si'pati'nda isletiliwi mu'mkin bolg'an turaqli'lar kompleksi keltirilgen.

8.1-keste. C++ tilindegi belgi turaqli'lar

Belgiler klasslari'	Belgi turaqli'lar
U'lken ha'ripler	'A'...'Z', 'A'...'Ya'
Kishi qarflar	'a'...'z', 'a'...'ya'
Nomerler	'0'...'9'
Bos jay	gorizontal tabulyatsiya(ASCII kodi' 9), qatardi' wo'tkeriw(ASCII kodi' 10), vertikal tabulyatsiya(ASCII kodi' 11), formani' wo'tkeriw(ASCII kodi' 12) karetkani qaytari'w(ASCII kodi' 13)
Punktuaasiya belgileri	! " # \$ % & ' () * + , - . / : ; < = > ? @ [\] ^ _ { } ~

(aji'rati'wshi'lar)	
Basqari'w belgileri	ASCII kodi' 0...1Fh arali'g'inda ha'm 7Fh bolg'an belgiler
Probel	ASCII kodi' 32 bolg'an belgi
Won alti'li'q nomerler	'0'...'9', 'A'...'F', 'a'...'f'

Qatar massivi dag'aza yetiliwinde, qatar aqi'ri'na terminator qoyi'li'wi' ha'm na'tiyjede qatarg'a qosi'msha bir bayt boli'wi'n inabatqa ali'ni'wi' kerek:

```
char qatar[10];
```

Usi' dag'azada qatar ushi'n ja'mi 10 bayt ajrati'ladi' - 9 qatar payda yeti'wshi' belgiler ushi'n ha'm 1 bayt terminator ushi'n.

Qatar wo'zgeriwshiler dag'aza yetiliwinde baslang'i'sh ma'nislerdi' qabi'llawi' mu'mkin. Bul halda kompilyator avtomatik tu'rde qatar uzi'nli'g'i'n yesaplaydi' ha'm qatar aqi'ri'na terminatorni qosi'p qo'yadi:

```
char Ha'pte_kuni[]="Juma" ;
```

Usi' dag'aza to'mendegi dag'aza menen ekvivalent:

```
char Hapte_kuni[]={ 'J', 'u', 'm', 'a', '\0' } ;
```

Qatar ma'nisin woqi'li'wi'nda ag'i'mli' woqi'w operatori >> worni'na getline() funksiyasin isletgen maqul yesaplanadi', sebebi ag'i'mli' woqi'li'wi'nda probeller biykar yetiledi (wolar qatar belgisi yesaplansa da) o'qilipati'rg'an belgiler izbe-izligi qatardan «oshib» kyetganda da belgilerdi kirgiziw dawam yetiwi mu'mkin. Na'tiyjede qatar wo'zine aji'rati'lg'an wo'lshemden arti'q belgilerdi «qabi'l» yetedi. Usi'ni'n' sebebini, getline() funksiyasi' yeki parametrge iye boli'p, birinshi parametr woqi'w a'melge asi'ri'li'p ati'rg'an qatarg'a ko'rsetkish, yekinshi parametrda bolsa o'qiliwi kerek bolg'an belgiler sani' ko'rsetiledi. Qatardi' getline() funksiyasi' arqali' woqi'wg'a mi'sal ko'reylik:

```
#include <iostream>
using namespace std;
int main()
{
```

```

char qatar[6];
cout<<"Qatardi' kiritin':"<<'\n';
cin.getline(qatar, 6) ;
cout<<"Siz kirgizgen qatar:"<<qatar;
return 0;
}

```

Programmada isletilgen qatar uzi'nli'g'i 5 dana belgini qabi'llawi mu'mkin, arti'qshalari' tastap jiberiledi. `getline()` funksiyasi'na murajatda yekinshi parametr ma'nisi o'qilipati'rg'an qatar uzi'nli'g'i'nan u'lken bolmaytug'i'ni' kerek.

Qatar menen isleytug'i'n funksiyalardi'n' ko'pshiligi `string.h` kitapxanasida jiyinalg'an. Sali'sti'rg'anda ko'p isletiletug'i'n funksiyalardi'n' suwretlewin kyeltiramiz.

Qatar uzi'nli'g'i'n ani'qlaw funksiyalari': Qatarlar menen islewde, ko'binese qatar uzi'nli'g'i'n biliw za'ru'r boladi'. Woni'n' ushi'n `string.h` kitapxanasida `strlen()` funksiyasi' ani'qlang'an boli'p, woni'n' sintaksisi to'mendegishe boladi':

```

size_t strlen(const char* string )

```

Bul funksiya uzi'nli'g'i' yesaplani'wi' kerek bolg'an qatar basi'na ko'rsetkish bolg'an bir parametrga iye ha'm wol na'tiyje si'pati'nda belgisiz pu'tin sandi' qaytaradi'. `strlen()` funksiyasi' qatardi'n' real uzi'nli'g'i'nan bir kem ma'nis qaytaradi', yag'ni'y nol-terminator worni' yesapqa ali'nbaydi'.

Tap sol maqsette `sizeof()` funksiyasi'nan da paydalani'w mu'mkin ha'm wol `strlen()` funksiyasi'nan ayri'qsha tu'rde qatardi'n' real uzi'nli'g'i'n qaytaradi'. To'mende keltirilgen mi'salda qatar uzi'nli'g'i'n yesaplawdi'n' ha'r yeki varianti' keltirilgen:

```

#include <iostream>
#include <string.h>
using namespace std;
int main()
{

```

```

char Str[]="1234567890";
cout <<"strlen(Str) ="<<strlen(Str) <<endl;
cout<<"sizeof(Str) ="<<sizeof(Str) <<endl;
return 0;
}

```

Programma islewi na'tiyjesinde ekrang'a

```
strlen(Str) =10
```

```
sizeof(Str) =11
```

xabarlari' shi'g'adi'.

A'detde sizeof() funksiyasi'nan getline() funksiyasi'ni'n' yekinshi argumenti sapasi' isletiledi ha'm qatar uzi'nli'g'i'n ayqi'n ko'rsetpeslik imkaniyati'n beredi:

```
cin.getline(Qatar, sizeof(Qatar) ) ;
```

Qatarlardi' nusqalaw: Qatar ma`nisin birinen yekinshisine nusqalaw mu'mkin. Bul maqsette bir qatar standart funksiyalar ani'qlang'an boli'p, wolardi' ayi'ri'mlari'ni'n' suwretlewlarin keltiremiz.

strcpy() funksiyasi' prototipi

```
char* strcpy(char* str1, const char* str2);
```

ko'riniske iye ha'm bul funksiya str2 qatardag'i' belgilerdi str1 qatarg'a baytma-bayt nusqalaydi'. Nusqalaw str2 ko'rsetip turg'an qatardag'i' nol-terminal u'srasqansha dawam yetedi. Usi'ni'n' sebebini, str2 qatar uzi'nli'g'i' str1 qatar uzi'nli'g'i'nan u'lken yemesligine isenim payda yetiw kerek, keru jag'day berilgen ken'isliginde (segmentte) str1 qatardan keyin joylasqan berilgenler «ustinde» str2 qatardi'n' «artip qalg'an» bo'legi jazi'li'wi' mu'mkin.

Na'wbettegi programma bo'legi "Qatardi' nusqalaw!" qatari'n Str qatarg'a nusqalaydi':

```
char Str[20];
```

```
strcpy(Str, "Qatardi' nusqalaw!");
```

Za'ru'r bolg'anda qatardi'n' qandayda bir jayi'ndan baslap, aqi'ri'ina shekem nusqalaw mu'mkin. Mi'sali, "Qatardi' nusqalaw!" qatari'n 8-

belgisinen baslap nusqa ali'w za'ru'r bolsa, woni' to'mendegishe sheshiw mu'mkin:

```
#include <iostream>
#include <string.h>
using namespace std;
int main()
{
    char Str1[20]="Qatardi' nusqalaw!";
    char Str2[20];
    char* korsetkish = Str1;
    korsetkish += 7;
    strcpy(Str2, korsetkish) ;
    cout<<Str2<<endl;
    return 0;
}
```

strncpy() funksiyasi'ni'n' strcpy() funksiyasi'nan ayri'qsha jayi' sonda, wol jag'dayda bir qatardan yekinshisine nusqalanatug'in belgiler sani' ko'rsetiledi. Woni'n' sintaksisi to'mendegi ko'riniske iye:

```
char* strncpy(char* str1, const char* str2, size_t num)
```

Yeger str1 qatar uzi'nli'g'i' str2 qatar uzi'nli'g'i'nan kishi bolsa, arti'qsha belgiler «kesip» taslanadi'. strncpy() funksiyasi' isletiliwine mi'sal ko'reylik:

```
#include <iostream>
#include <string.h>
using namespace std;
int main()
{
    char Uzin_str[]="01234567890123456789";
    char Qisqa_str[]="ABCDEF";
    strncpy(Qisqa_str, Uzin_str, 4) ;
    cout<<"Uzin_str= "<<Uzin_str<<endl;
    cout<<"Qi'sqa_str= "<<Qisqa_str<<endl;
    return 0;
}
```

Programmada Uzun_str qatari' basi'nan 4 belgi Qi'sqa_str qatari'na, woni'n' aldi'ng'i' ma'nisleri' u'stine jaylanadi' ha'm na'tiyjede ekrang'a

```
01234567890123456789
```

```
0123EF
```

qatarlar baspadan shi'g'ari'ladi'.

strdup() funksiyasi'na birden-bir parametr si'pati'nda qatar-derekke ko'rsetkish uzati'ladi'. Funksiya, qatarg'a saykes yaddan jay ajratadi', wog'an qatardi' nusqalaydi' ha'm ju'zege kelgen qatar - nusqa adresin juwap si'pati'nda qaytaradi', strdup() funksiya sintaksisi:

```
char* strdup(const char* source)
```

To'mendegi programma bo'leginde qatar1 qatari'ni'n' nusqasi' yadni'n' qatar2 ko'rsetken jayi'nda payda boladi':

```
char* qatar1="Qatar nusqasi'n ali'w."; char* qatar2;  
qatar2=strdup(qatar1) ;
```

Qatarlardi' jalg'aw: Qatarlardi' jalg'aw (konkatenatsiya) a'meli jan'a qatarlardi' payda yetiwde ken' qo'llaniladi. Bul maqsette string.h kitapxanasida strcat() ha'm strncat() funksiyalari' ani'qlang'an. strcat() funksiyasi' sintaksisi to'mendegi ko'rinishke iye:

```
char* strcat(char* str1, const char* str2)
```

Funksiya islewi na'tiyjesinde str2 qatar, funksiya qaytaratug'i'n qatar str1 qatar aqi'ri'na jalg'anadi'. Funksiyani' shaqi'ri'wdan aldi'n str1 qatar uzi'nli'g'i', wog'an str2 qatari' jalg'ani'wi' ushi'n yjeterli boli'wi' yesapqa ali'ng'an boli'wi' kerek.

To'mende keltirilgen a'meller izbe-izliginin' atqari'li'wi' na'tiyjesinde qatar qatari'na qosi'msha qatar jalg'ani'wi' ko'rsetilgen:

```
char qatar[80];  
strcpy(qatar, "Bul qatarg'a") ;  
strcat(qatar, "qatar jalg'andi'.") ;
```

A'meller izbe-izligini atqari'li'wi' na'tiyjesinde ko'rsetip ati'rg'an wori'nda "Bul qatarg'a qatar asti' jalg'andi.'" qatari' payda boladi'.

`strncat()` funksiyasi' `strcat()` funksiyadan ayri'qsha tu'rde `str1` qatarg'a `str2` qatardin' ko'rsetilgen uzi'nli'qdag'i' qatar bo'legin jalg'aydi. Jalg'anatug'i'n qatar bo'legi uzi'nli'g'i' funksiyani'n' u'shinshi parametri si'pati'nda beriledi.

Funksiya sintaksisi:

```
char* strncat(char* str1, const char* str2, size_t num)
```

To'mende keltirilgen programma bo'leginde `str1` qatarg'a `str2` qatardi'n' baslang'i'sh 10 ta belgiden ibarat qatar bo'legin jalg'aydi:

```
char satr1[80]="Programmalisti'ri'w tillerine mi'sal bul  
- ";  
char satr2[80]="C++, Paskal, Basic";  
strncat(satr1, satr2, 10) ;  
cout<<satr1;
```

A'meller atqari'li'wi' na'tiyjesinde ekrang'a

```
Programmalisti'ri'w tillerine mi'sal bul-C++, Pascal
```

qatari' baspadan shi'g'ari'ladi'.

Ma'sele. Nol-terminator menen tawsi'latug'i'n `S`, `S1` ha'm `S2` qatarlar berilgen. `S` qatardag'i' `S1` qatar asti'lari' `S2` qatar asti' menen almasti'rilsin. Ma'seleni sheshiw ushi'n to'mendegi ma'sele asti'lari'n sheshiw za'ru'r:

- 1) `S` qatari'nda `S1` qatar asti'n kiriw warni'n ani'qlaw;
- 2) `S` qatari'nan `S1` qatar asti'n wo'shiriw;
- 3) `S` qatari'nda `S1` qatar asti' warni'na `S2` qatar asti'n wornati'w.

Biraq bul ma'sele asti'lari'ni'n' sheshimlari C++ tilinin' standart kitapxanalari'nda funksiyalar ko'rinishinde bar bolsada, wolar kodti' qayta jazi'w paydalani'wshig'a bul a'mellerdin' ishki ma'nisin tu'siniwge

mu'mkinshilik beredi. To'mende ma'sele sheshiminin' programma teksti keltirilgen:

```
#include <iostream>
#include <string.h>
using namespace std;
const int n=80;
int Izlew(char *, char *) ;
void Qirqiw(char *, int, int) ;
void Jaylastiriw(char *, char *, int) ;
int main()
{
    char Qatar[n], Qatar1[n], Qatar2[n];
    cout<<"Qatardi' kiritin:'";
    cin.getline(Qatar, n) ;
    cout<<"Almasti'ri'latug'i'n qatar asti'n kiritin': ";
    cin.getline(Qatar1, n) ;
    cout<<"o'rni qoyi'latug'i'n qatardi' kiritin:'";
    cin.getline(Qatar2, n) ;
    int Qatar1_uzinligi=strlen(Qatar1) ;
    int Qatar_asti_jayi;
    do
    {
        Qatar_asti_jayi=Izlew(Qatar, Qatar1) ;
        if(Qatar_asti_jayi!=-1)
        {
            Qirqiw(Qatar, Qatar_asti_jayi, Qatar1_uzinligi) ;
            Jaylastiriw(Qatar, Qatar2, Qatar_asti_jayi);
        }
    } while(Qatar_asti_jayi!=-1) ;
    cout<<"Almasti'ri'w na'tiyjesi: "<<Qatar;
    return 0;
}
int Izlew(char qatar[],char qatar_asti[])
{
    int qatar_farqi=strlen(qatar)-strlen(qatar_asti) ;
    if(qatar_farqi>=0)
    {
        for(int i=0;i<=qatar_farqi;i++)
```

```

    {
        bool ustma_ust=true;
        for(int j=0; qatar_asti[j]!='\0' && ustma_ust; j++)
            if(qatar[i+j]!=qatar_asti[j])
                ustma_ust=false;
        if(ustma_ust)
            return i;
    }
    }
    return-1;
}

void Qirqiw(char qatar[], int jay, int qirqiw_sani)
{
    int qatar_uzinligi=strlen(qatar) ;
    if(jay<qatar_uzinligi)
    {
        if(jay+qirqiw_sani>=qatar_uzinligi)
            qatar[jay]='\0';
        else
            for(int i=0; qatar[jay+i-1]!='\0'; i++)
                qatar[jay+i]=qatar[jay+qirqiw_sani+i];
    }
}

void Jaylastiriw(char qatar[],char qatar_asti[], int jay)
{
    char waqtinsha[n];
    strcpy(waqtinsha, qatar+jay);
    qatar[jay]='\0';
    strcat(qatar, qatar_asti) ;
    strcat(qatar, waqtinsha) ;
}

```

Programmada ha'r bir ma'sele asti'na saykes funksiyalar du'zilgen:

1) int Izlew(char qatar[],char qatar_asti[]) - funksiyasi' qatar qatari'na qatar_asti qatari'ni'n' shep ta'repden birinshi kiriwinin'

worni'n qaytaradi. Yeger qatar qatari'nda qatar_asti u'shraspasa - 1 ma`ninin qaytaradi'.

2) void Qirq'w(char qatar[], int jay, int qirqiw_sani) - funksiyasi' qatar qatari'ni'n' jay worni'nan baslap qirqiw_sani sandag'i' belgilerdi qirqib tashlaydi. Funksiya na'tiyjesi qatar qatari'nda payda boladi';

3) void Jaylaw(char qatar[],char qatar_asti[], int jay) - funksiyasi' qatar qatari'na, woni'n' jay worni'nan baslap qatar_osti qatari'n jaylasti'radi'.

Bas funksiyada qatar(S), wol jag'dayda almasti'ri'latug'i'n qatar(S1) ha'm S1 worni'na jaylasti'ri'latug'i'n qatar(S2) ag'i'mdan woqi'ladi'. Ta'kirarlaw operatori atqari'li'wi'ni'n' ha'r bir qa'deminde S qatari'ni'n' shep ta'repinen baslap S1 qatari' i'zlenedi. Yeger S qatari'nda S1 bar bolsa, wol qi'rqi'ladi' ha'm sol wori'ng'a S2 qatari' jaylasti'ri'ladi'. Ta'kirarlaw procesi Izlew() funksiyasi' - 1 ma`ninin qaytarguncha dawam yetedi.

Soraw ha'm tapsi'rmalar

1. C++ tilinde qatar ma`ninin woqi'w qanday a'melge asi'ri'ladi'?
2. C++ tilinde qatarlar qanday dag'aza yetiledi?
3. C++ tilinde qatar uzi'nli'g'i'n ani'qlaw.
4. C++ tilinde qatarlardi' nusqalaw.
5. C++ tilinde qollani'latug'i'n belgiler kompleksinin' tu'rleri.
6. Qatar terminatori tu'sinigi.
7. Qatar tipli wo'zgeriwshilerge mi'sal keltiring.
8. Qatarlardi' nusqalawg'a tiyisli mi'sal keltiring.

Strings

By far, the most common use for one-dimensional arrays is to create character strings. In C++, a *string* is defined as a character array that is terminated by a null. A null character is specified using `'\0'`, and is zero. Because of the null terminator, it is necessary to declare a character array to be one character longer than the largest string that it will hold.

For example, if you want to declare an array `str` that could hold a 10-character string, you would write:

```
char str[11];
```

Specifying the size as 11 makes room for the null at the end of the string.

One way to solve the whitespace problem is to use another of C++'s library functions, `gets()`. The general form of a `gets()` call is:

```
gets(array-name);
```

If you need your program to read a string, call `gets()` with the name of the array, without any index, as its argument. Upon return from `gets()`, the array will hold the string input from the keyboard. The `gets()` function will continue to read characters until you press ENTER. The header used by `gets()` is `<cstdio>`.

This version of the preceding program uses `gets()` to allow the entry of strings containing spaces.

Some String Library Functions

C++ supports a wide range of string-manipulation functions. The most common are

```
strcpy()  
strcat()  
strlen()  
strcmp()
```

The string functions all use the same header, `<cstring>`. Let's take a look at these functions now.

strcpy

A call to `strcpy()` takes this general form:

```
strcpy(to, from);
```

The `strcpy()` function copies the contents of the string *from* into *to*. Remember, the array that forms *to* must be large enough to hold the string contained in *from*. If it isn't, the *to* array will be overrun, which will probably crash your program.

strcmp

A call to `strcmp()` takes this general form:

```
strcmp(s1, s2);
```

The `strcmp()` function compares two strings and returns 0 if they are equal. If *s1* is greater than *s2* lexicographically (i.e., according to dictionary order), then a positive number is returned; if it is less than *s2*, a negative number is returned.

The `password()` function, shown in the following program, is a password-verification routine. It uses `strcmp()` to check a user's input against a password.

strlen

The general form of a call to **strlen()** is

```
strlen(s);
```

where *s* is a string. The **strlen()** function returns the length of the string pointed to by *s*.

Using the Null Terminator

The fact that all strings are null-terminated can often be used to simplify various operations on strings. For example, look at how little code is required to uppercase every character in a string:

18-Lekciya: Qatarlar u'stinde a'meller

Joba:

1. Qatarlardi' sali'sti'ri'w
2. Qatardag'i' ha'ripler registrini almasti'ri'w
3. Qatardi' teris ta'rtiplew

Tayansh so'zler: *Qatarlardi' sali'sti'ri'w, satrdagi ha'ripler registrini almasti'ri'w, qatardi' kerri ta'rtiplew.*

Qatarlardi' sali'sti'ri'w: Qatarlardi' sali'sti'ri'w wolardi'n' saykes wori'ndag'i' belgilerding kodlari'n wo'z-ara sali'sti'ri'w menen ani'qlanadi'. Woni'n' ushi'n string.h kitapxanasida standart funksiyalar bar.

strcmp() funksiyasi' sintaksisi

```
int strcmp(const char* str1, const char* str2)
```

ko'rinishine iye boli'p, funksiya str1 ha'm str2 sali'sti'ri'w na'tiyjesi si'pati'nda san ma'nisti qaytaradi' (mi'sali, pu'tin i wo'zgeriwshisinde) ha'm wolar to'mendegishe ani'qlama berdedi:

- a) $i < 0$ - yeger str1 qatari' str2 qatari'nan kishi bolsa;
- b) $i = 0$ - yeger str1 qatari' str2 qatari'na ten' bolsa;
- s) $i > 0$ - yeger str1 qatari' str2 qatari'nan u'lken bolsa.

Funksiya ha'riplerinin' registrini pari'qlaydi'. Buni' mi'salda ko'riwimiz mu'mkin:

```
char satr1[80]="programmalasti'ri'w tilleri:C++, pascal.";
char satr2[80]="programmalasti'ri'w tilleri:C++, Pascal.";
int i;
i=strcmp(satr1, satr2);
```

Na'tiyjede i wo'zgeriwshisi won' ma'nis qabi'l yetedi, sebebi sali'sti'ri'li'pati'rg'an qatarlardag'i' «pascal» ha'm «Pascal» qatar bo'limlerinde birinshi ha'ripler pari'q yetedi. Keltirilgen mi'salda i ma'nisi

32 boladi'. Bul pari'qlani'wshi' ha'ripler kodlari'ni'n' ayi'rmasi'. Yeger funksiyag'a

```
i = strcmp(satr2, satr1) ;
```

ko'rinishde murajat yetilse i ma'nisi kerri san 32 boladi'.

Yeger qatarlardag'i' bas yamasa kishi ha'riplerdi pari'qlamasta'n sali'sti'ri'w a'melin wori'nlaw za'ru'r bolsa, woni'n' ushi'n strcmpi() funksiyasi'nan paydalani'w mu'mkin. Joqari'da keltirilgen mi'saldag'i' qatarlar ushi'n

```
i = strcmpi(satr2, satr1) ;
```

a'meli wori'nlang'anda i ma'nisi 0 boladi'.

strncmp() funksiyasi' sintaksisi:

```
int strncmp(const char* str1, const char* str2, size_t num);
```

ko'rinishde boli'p, str1 ha'm str2 qatarlardi' baslang'i'sh num sani'ndag'i' belgilerini sali'wtiradi. Funksiya ha'ripler registrin inabatqa aladi'. Joqari'da mi'salda ani'qlang'an str1 ha'm str2 qatarlar ushi'n

```
i = strncmp(satr1, satr2, 31);
```

a'meli atqari'li'wi'nda i ma'nisi 0 boladi', sebebi qatarlar basi'ndag'i' 31 belgiler birdey.

strncmpi() funksiyasi' strncmp() funksiyasidyek a'mel yetedi, farkli ta'repi sonda, sali'sti'ri'wda ha'riplerdin' registrini yesapqa ali'nbaydi'. Tap sol qatarlar ushi'n

```
i = strncmpi(satr1, satr2, 32);
```

a'meli atqari'li'wi' na'tiyjesinde i wo'zgeriwshi ma'nisi 0 boladi'.

Qatardag'i' ha'ripler registrin almasti'ri'w: Berilgen qatardag'i' kishi ha'riplerdi bas ha'riplerge yamasa kerisiga almastiriwg'a saykes tu'rde _strupr() ha'm _strlwr() funksiyalar ja'rdeminde a'melge asiriw

mu'mkin. Kompilyatorlardin' ayi'ri'm variantlari'nda funksiyalar ati'ndag'i tagchiziq ('_') bolmasli'g'i ha'm mu'mkin.

`_strlwr()` funksiyasi' sintaksisi

```
char _strlwr(char* str);
```

ko'riniside boli'p, argument si'pati'nda berilgen qatardag'i' bas ha'riplardi kishi ha'riplerge almasti'radi' ha'm payda bolg'an qatar adresini funksiya na'tiyjesi si'pati'nda qaytaradi'. To'mendegi programma bo'legi `_strlwr()` funksiyasi'nan paydalanishga mi'sal boladi'.

```
char str[]="10 DANA U'LKEN HA'RIPLER";
_strupr(str) ;
cout<<str;
```

Na'tiyjede ekrang'a

```
10 ta u'lken ha'ripler
```

qatari' baspadan shi'g'ari'ladi'.

`_strupr()` funksiyasi' tap `_strlwr()` funksiyasiday a'mel yetedi, lekin qatardag'i' kishi ha'riplardi bas ha'riplerge almasti'radi':

```
char str[]="10 ta u'lken ha'ripler";
_strupr(str) ;
cout<<str;
```

Na'tiyjede ekrang'a

```
10 TA U'LKEN HA'RIPLER
```

qatari' baspadan shi'g'ari'ladi'..

Programmalisti'ri'w a'meliyati'nda belgilerdi qanday arali'qqa ti'yi'sli yekenligin biliw za'ru'r boladi'. Buni' `ctype.h` fayli'nda dag'aza yetilgen funksiyalar ja'rdeminde ani'qlaw mu'mkin. To'mende wolardi'n' bir bo'leginin' suwretlewi keltirilgen:

`isalnum()` - belgi nomer yamasa ha'rip (`true`) yamasa joq yekenligin (`false`) ani'qlaydi';

`isalpha()` - belgini ha'rip (`true`) yamasa joq yekenligin (`false`) ani'qlaydi';

isascii() -belgini kodi' 0..127 arali'g'inda (true) yamasa joq yekenligin (false) ani'qlaydi';

isdigit() -belgini nomerler diapazoni'na ti'yi'sli (true) yamasa joq yekenligin (false) ani'qlaydi'.

Bul funksiyalardan paydalani'wg'a mi'sal kyeltiramiz.

```
#include <iostream.h>
#include <ctype.h>
#include <string.h>
int main()
{
    char qatar[5];
    int qa'te;
    do
    {
        qate=0;
        cout<<"\nTug'ilgan ji'li'ngizni kiritin: ";
        cin.getline(qatar, 5) ;
        for(int i=0;i<strlen(qatar) && !qa'te; i++)
        {
            if(isalpha(qatar) )
            {
                cout<<"Siz ha'rip kirgizdin'iz !";
                qa'te=1;
            }
            else
            if(iscntrl(qatar) )
            {
                cout<<"Siz basqari'w belgisin kirgizdin'iz !";
                qa'te=1;
            }
            else
            if(ispunct(qatar) )
            {
                cout<<"Siz punktuatsiya belgisin kirgizdin'iz!";
                qa'te=1;
            }
            else
            if(!isdigit(qatar) )
```

```

{
cout<<"Siz nomerden ayri'qsha belgi kirgizdin'iz!";
qa'te=1;
}
}
if(!qa'te)
{
cout<<"Sizdi tug'ilgan ji'li'n'i'z: "<<qatar;
return 0;
}
}while(1) ;
}

```

Programmada paydalaniwshi'g'a tuwi'lg'an ji'li'n kirgiziw usi'ni's yetiledi. Kiritilgen sa'ne qatar wo'zgeriwshisine woqi'ladi' ha'm yeger qatardi'n' ha'r bir belgisi (qatar) ha'rip yamasa basqari'w belgisi yamasa punktuatsiya belgisi bolsa, sol haqqi'nda xabar beriledi ha'm tuwi'lg'an ji'ldi' qayta kirgiziw usi'ni's yetiledi. Programma tuwi'lg'an ji'l(to'rtew nomer) tuwri' kiritilgende "Sizdi tuwi'lg'an ji'li'n'i'z: XXXX" qatari'n baspadan shi'g'ari'w menen wo'z jumi'si'n toqtatadi'.

Qatardi' kerri ta'rtiplew: Qatardi' kerri ta'rtiplewdi ushi'n `strrev()` funksiyasi'nan paydalani'w mu'mkin. Bul funksiya to'mendegishe prototipga iye:

```
char* strrev(char* str);
```

Qatar revyersin payda yeti'wg'a mi'sal:

```
char str[]="telefon";
```

```
cout<<strrev(str) ;
```

a'meller atqari'li'wi' na'tiyjesinde ekrang'a

```
nofelet
```

qatari' baspadan shi'g'ari'ladi'.

Soraw ha'm tapsi'rmalar

1. C++ tilinde qatarlardi' nusqalaw.
2. C++ tilinde qatarlardi' jalg'aw ushi'n qanday funksiyalar qollani'ladi'?
3. Qatardi' teris ta'rtiplew.
4. Qatarlardi' jalg'awg'a mi'sal keltiring.
5. Qatarlardi' sali'sti'ri'w qanday a'melge asi'ri'ladi'?
6. `strnmpi()` funksiyasi' qanday wazi'ypani' atqaradi'?
7. `strncmp()` funksiyasi' qanday wazi'ypani' atqaradi'?
8. `strchr()` funksiyasi'ni'n' wazi'ypasi'n ani'qlama berin'.
9. `_strupr()` ha'm `_strwr()` funksiyalardi' ani'qlama berin'.
10. Qatardag'i' ha'ripler registrini almasti'ri'wg'a mi'sal keltiring.

TIP: In addition to **toupper()**, the C++ standard library contains several other character-manipulation functions. For example, the complement to **toupper()** is **tolower()**, which returns the lowercase equivalent of its character argument. Other character functions include **isalpha()**, **isdigit()**, **isspace()**, and **ispunct()**. These functions each take a character argument and determine if it belongs to that category. For example, **isalpha()** returns true if its argument is a letter of the alphabet.

strcmp

A call to **strcmp()** takes this general form:

```
strcmp(s1, s2);
```

The **strcmp()** function compares two strings and returns 0 if they are equal. If *s1* is greater than *s2* lexicographically (i.e., according to dictionary order), then a positive number is returned; if it is less than *s2*, a negative number is returned.

The **password()** function, shown in the following program, is a password-verification routine. It uses **strcmp()** to check a user's input against a password.

strcpy

A call to **strcpy()** takes this general form:

```
strcpy(to, from);
```

The **strcpy()** function copies the contents of the string *from* into *to*. Remember, the array that forms *to* must be large enough to hold the string contained in *from*. If it isn't, the *to* array will be overrun, which will probably crash your program.

19-Lekciya: Qatarlarda funksiyalar

Joba:

1. Qatarda belgi izlew funksiyalari'
2. Qatar bo'limlerin izlew funksiyalari'
3. Tu'rlerdi wo'zgertiw funksiyalari'

Tayansh so'zler: *Qatarda belgi izlew funksiyalari', qatar bo'limlerin izlew funksiyalari', tu'rlerdi wo'zgertiw funksiyalari'*

Qatarda belgi izlew funksiyalari': Qatarlar menen islewde wol jag'dayda qandayda-bir belgi izlew ushi'n `string.h` kutubxonasinda bir qatar standart funksiyalar bar.

Qandayda-bir belgi berilgen qatarda bar yamasa joq yekenligin aniqlap beretug'i'n `strchr()` funksiyasi'ni'n' prototipi

```
char* strchr(const char* string, int c) ;
```

ko'rinishde boli'p, wol `s` belginin' `string` qatari'nda izleydi. Yeger izlew a'wmetli bolsa, funksiya sol belginin' qatardag'i' worni'n (adresin) funksiya na'tiyjesi si'pati'nda qaytaradi', bolmasa, yag'ni'y belgi qatarda uchramasa funksiya `NULL` ma'nisin qaytaradi'. Belgini izlew qatar basi'nan baslanadi'.

To'mende keltirilgen programma bo'legi belgini qatardan izlew menen baylani'sli'.

```
char qatar[]="0123456789";  
char* pQatar;  
pQatar=strchr(qatar, '6') ;
```

Programma islewi na'tiyjesinde `pQatar` ko'rsetkishi qatar qatari'ni'n' '6' belgisi jaylasqan worni' adresin ko'rsetedi.

`strchr()` funksiyasi' berilgen belgi(`s`) berilgen qatar (`char`) aqi'ri'nan baslap izlaydi. Yeger izlew tabi'sli' bolsa, belgi qatarg'a son'g'i' kiriwinin' worni'n qaytaradi', bolmasa `NULL`.

Mi'sal ushi'n

```
char qatar[]="0123456789101112";
char* pQatar;
pSatr=strrchr(qatar, '0') ;
```

a'mellarin atqari'li'wi'nda pQatar ko'rsetkishi qatar qatari'ni'n' "01112" qatar bo'leginn' baslani'wi'na ko'rsetedi.

strspn() funksiyasi' yeki qatardi' belgilerdi sali'wtiradi. Funksiya to'mendegi

```
size_t strspn(const char* str1, const char* str2) ;
```

ko'rinske iye boli'p, wol str1 qatardag'i' str2 qatarg'a kiretug'i'n qandayda-bir belgi izlaydi ha'm yeger bunday element topilsa, woni'n' indeksi funksiya ma'nisi si'pati'nda qaytari'ladi', bolmasa funksiya qatar uzi'nli'g'i'nan bir arti'q ma'nisti qaytaradi'.

Mi'sal:

```
char satr1[]="0123ab6789012345678";
char satr2[]="a32156789012345678" ;
int ayri'qsha_belgi;
ayriqsha_belgi=strspn(satr1, satr2) ;
```

```
cout<<"Satr1 qatari'ndag'i' Satr2 qatarg'a kirmaydigan
birinshi belgi indeksi = "<<farqli_belgi;
```

```
cout<<"ha'm wol '"<<satr1[farqli_belgi]<<"' belgisi.";
```

a'meller atqari'li'wi' na'tiyjesinde ekrang'a qatarlardag'i' saykes tushmagan belgi indeksi = 5 qatari' baspadan shi'g'ari'ladi'.

```
strcspn() funksiyasi'ni'n' prototipi
```

```
size_t strcspn(const char* str1, const char* str2) ;
```

ko'rinsinde boli'p, wol str1 ha'm str2 qatarlardi' sali'wtiradi ha'm str1 qatari'ni'n' str2 qatari'na kirgen birinshi belgani indeksini qaytaradi'.

Mi'sali'.

```
char qatar[]="Birinshi qatar";
int index;
index=strcspn(qatar, "sanaq sistemasi'") ;
```

a'melleri wori'nlang'annan keyin index wo'zgeriwshisi 1 ma'nisin qabi'l yetedi, sebebi birinshi qatardi'n' birinshi wori'ndag'i' belgisi yekinshi qatarda ushraydi'.

strupbrk() funksiyasi'ni'n' prototipi

```
char strpbrk(const char* str1, const char* str2) ;
```

ko'riniske iye boli'p, str1 qatardag'i' str2 qatarg'a kiretug'i'n qandayda-bir belgini izlaydi ha'm yeger bunday element topilsa, woni'n' adresi funksiya ma'nisi si'pati'nda qaytari'ladi', bolmasa funksiya NULL ma'nisi qaytaradi'. To'mendegi mi'sal funksiyani' qanday islewin ko'rsetedi.

```
char satr1[]="0123456789ABCDEF";  
char satr2[]="ZXYabcdefABC";  
char* element;  
element = strpbrk(satr1, satr2) ;  
cout<<element<<'\n';
```

Programma islewi na'tiyjesinde ekrang'a str1 qatarinin'

```
ABCDEF
```

qatar asti'si' baspadan shi'g'ari'ladi'.

Qatar bo'limlerin izlew funksiyalari': Qatarlar menen islewde bir qatarda yekinshi bir qatardi'n'(yamasa woni'n' qandayda bir bo'legin) toli'q kiriwin ani'qlaw menen baylani'sli' ma'seleler sali'sti'rg'anda ko'p ushraydi'. Mi'sali', tekst tahrirlaridagi qatardag'i' qandayda-bir qatar bo'legin yekinshi qatar bo'legi menen almasti'ri'w ma'selesin mi'sal keltiriw mu'mkin(joqari'da tap sonday ma'sele ushi'n programma keltirilgen). Standart string.h kitapxanasi' bul kategoriyadag'i' ma'seleler ushi'n bir neshe funksiyalardi' usi'ni's yetedi.

strstr() funksiyasi' to'mendegishe dag'aza yetiledi:

```
char* strstr(const char* str, const char* substr);
```

Bul funksiya str qatari'na substr qatar bo'legi kiriwi tyekshiradi, yeger substr qatar bo'legi str qatari'na toli'q kiriwi bar bolsa, qatardi'n' shep ta'repden birinshi kiriwdegi birinshi belginin' adresi juwap si'pati'nda qaytari'ladi', bolmasa funksiya NULL ma'nisin qaytaradi'.

To'mendegi mi'sal strstr() funksiyasi'n isletiwdi ko'rsetedi.

```
char satr1[]="Qatardan qatar asti'si' izertlep ati'r,  
qatar asti'si' bar";  
char satr2[]="qatar asti'si'";  
char* qatar_osti;  
qatar_osti=strstr(satr1, satr2) ;  
cout<<qatar_osti<<'\\n';
```

Programma buyri'qlari' atqari'li'wi' na'tiyjesinde ekrang'a qatar asti'si' izertlep ati'r, qatar asti'si' bar qatari' baspadan shi'g'ari'ladi'.

Keyingi programma bo'leginde qatarda basqa bir qatar bo'legi bar yamasa joq yekenligin basqariw varianti ko'rsetilgen:

```
char Atlar[]= "Suw ayi'rg'i'sher, Farxod, Munisa,  
Yerkin, Akmal, Nodira";  
char At[10];  
char* Qatardagi_ism;  
cout<<"Ismni kiritin': "; cin>>At;  
Qatardagi_ism=strstr(Atlar, At) ;  
cout<<"Bunaqa i'si'm ru'yxatda ";  
if(Qatardagi_ism==NULL)  
    cout<<"yo'q."<<'\\n';  
else  
    cout<<"bor."<<'\\n';
```

Programmada paydalanuvchidan qatar bo'legi si'pati'nda qandayda-bir atdi' kirgiziw talap yetiledi ha'm bul ma'nis I'si'm qatari'na woqi'ladi'. Kiritilgen i'si'm programmada ani'qlang'an dizimde(I'si'mlar qatari'nda) bar yamasa joq yekenligi ani'qlanadi' ha'm xabar beriledi.

strtok() funksiyasi'ni'n' sintaksisi

```
char* strtok(char* str, const char* delim) ;
```

ko'riniste boli'p, wol str qatari'nda delim qatar-diziminde berilgen aji'rati'wshi'lar arali'g'ina ali'ng'an qatar bo'limlerdi aji'rati'p ali'w

imkaniyati'n beredi. Funksiya birinshi qatarda yekinshi qatar-dizimdeg
aji'rati'wchini uchratsa, wodan keyin nol-terminatorni qoyi'w arqali' str
qatardi' yekige ajratadi'. Qatardi'n' yekinshi bo'leginden aji'rati'wshi'lar
menen «o'rap ali'ng'an» qatar bo'limleri tabi'w ushi'n funkسيyani' keyingi
shaqi'rilishida birinshi parametr worni'na NULL ma'nisin qoyi'w kerek
boladi'. To'mendegi mi'salda qatardi' bo'leklerge aji'rati'w ma'selesi
qaralgan:

```
#include <iostream>
#include <string.h>
#include <ctype.h>
#include <string.h>
using namespace std;
int main()
{
    char Atlar[]="Kamola, Farxod, Munisa, Yerkin? Akmal0,
Nodira";
    char Ajiratiwshi[]=" , !?.0123456789";
    char* Qatardagi_ism;
    Qatardagi_ism=strtok(Atlar, Ajiratiwshi) ;
    if(Qatardagi_ism) cout<<Qatardagi_ism<<'\n';
    while(Qatardagi_ism)
    {
        Qatardagi_ism=strtok(NULL, Ajiratiwshi) ;
        if(Qatardagi_ism)
            cout<<Qatardagi_ism<<'\n';
    }
    return 0;
}
```

Programma islewi na'tiyjesinde ekrang'a Atlar qatari'ndag'i' ' ' (probel), ' ' (vergul), '?' (soraw belgisi) ha'm '0' (nomer) menen aji'rati'lg'an qatar bo'limleri isimlari chop qlinadi:

```
Kamola
Farxod
Munisa
Yerkin
Akmal
```

Nodira

Tu'rlerdi wo'zgertiw funksiyalari': Qatarlar menen islewde qatar ko'rinishinde berilgen sanlardi', san tu'rleridegi ma'nislerg'a aylandi'ri'w yamasa kerri a'meldi wori'nlawg'a tuwri' keledi. C++ tilinin' strlib.h kitapxanasida bul a'mellerdi wori'nlawshi' funksiyalar kompleksi bar. To'mende sali'sti'rg'anda ko'p isletiletug'in funksiyalar suwretlewi keltirilgen.

atoi() funksiyasi'ni'n' sintaksisi

```
int atoi(const char* ptr) ;
```

ko'rinishke iye boli'p, ptr ko'rsetiwshi ASCIIZ-qatardi' int tu'rindegi sang'a o'tka-zishni a'melge asi'radi'. Funksiya qatar basi'nan belgilerdi sang'a aylantira baslaydi' ha'm qatar aqi'ri'i'na shekem yamasa birinshi nomer bolmag'an belgigacha isleydi. Yeger qatar basi'nda sang'a aylandi'ri'w mu'mkin bolmag'an belgi bolsa, funksiya 0 ma'nisin qaytaradi'. Lekin, sog'an itibar beriw kerekki, "0" qatari' ushi'n da funksiya 0 qaytaradi'. Yeger qatardi' sang'a aylandi'ri'wdag'i' payda bolg'an san int chyegarasidan shi'g'i'p kyetsa, sanni'n' kishi yeki bayti na'tiyje si'pati'nda qaytari'ladi'. Mi'sal ushi'n

```
#include <iostream>
#include <stdlib.h>
using namespace std;
int main()
{
    char str[]="32secund";
    int i=atoi(str) ;
    cout<<i<<endl;
    return 0;
}
```

programmasinin' na'tiyjesi si'pati'nda ekrang'a 32 sani'n baspadan shi'g'aradi'.

atol() funksiyasi' tap atoi() funksiyasidek a'mel yetedi, tek funksiya na'tiyjesi long tu'rinde boladi'. Yeger payda bolg'an san ma'nisi long shegarasi'na sig'masa, funksiya kutilmegen ma'nisti qaytaradi'.

atof() funksiyasi' dag'azasi'

```
double atof()(const char* ptr) ;
```

ko'rinisinde boli'p, ptr ko'rsetiwshi ASCIIZ-qatardi' double tu'rindagi ju'ziwshi noqatli' sang'a wo'tkeriwdi a'melge asi'radi'. Qatar to'mendegi formatda boli'wi' kerek:

```
<probeller><ishora><tsifrlar>.<tsifrlar><e|E><ishora>  
<tsifrlar>
```

Bul jerde:

<probeller>-probeller yamasa tabulyatorlar izbe-izligi;

<ishora>- '+' ha'm '-' belgileri;

<tsifrlar>-wonli'q nomerler;

<e|E>-wonli'q tiykar da'rejesin an'lati'wshi' belgiler.

Sang'a aylandi'ri'w birinshi formatg'a saykes kelmaytug'i'n belgi u'srasqansha yamasa qatar aqi'ri'i'na shekem dawam yetedi.

strtod() funksiyasi' atof() funksiyasi'nan ayri'qsha tu'rde qatardi' double turindagi sang'a wo'tkeriwde konvertatsiya procesi uzulgan waqi'tta aylandi'ri'w mu'mkin bolmag'an birinshi belgi adresin da qaytaradi'. Bul wo'z gezeginde qatardi' qa'te bo'legin qayta islew mu'mkinshilikti beredi.

strtod() funksiyasi'ni'n' sintaksisi

```
double strtod(const char *s, char **endptr) ;
```

ko'rinsike iye ha'm endptr ko'rsetkishi konvyertatsiya yetiliwi mu'mkin bolmag'an birinshi belgi adresi. Konvyertatsiya yetiletug'i'n qatarda qa'te bolg'an jag'dayni' ko'rsetiwshi mi'sal:

```
#include <stdlib.h>
```

```
#include <iostream.h>
```

```

int main(int argc, char* argv[])
{
    char qatar[]="3.14D15E+2";
    char **kursatkich;
    double x= strtod(qatar, kursatkich) ;
    cout<<"Konvertatsiya yetiletug'i'n qatar:
"<<qatar<<endl;
    cout<<"Konvertatsiya yetilgen x sani': "<<x<<endl;
    cout<<"Konvertatsiya u'zilgen qatar asti'si': ";
    cout<<*kursatkich;
    return 0;
}

```

Programma atqari'li'wi'nda x wo'zgeriwshi 3.14 sani'n qabi'l yetedi, kursatkich wo'zgeriwshisi qatardagi' 'D' belgisinin' adresini ko'rsetedi. Ekran'ga to'mendegi qatarlar izbe-izligi baspadan shi'g'ari'ladi':

```

Konvertatsiya yetiletug'i'n qatar: 3.14D15E+2
Konvertatsiya yetilgen x sani': 3.14
Konvertatsiya u'zilgen qatar asti'si': D15E+2

```

Bir qatar fo'nksiyalar kerri a'meldi, yag'ni'y berilgen sandi' qatarg'a aylandi'ri'w a'mellerin atqaradi'.

itoa() ha'm ltoa() funksiyalari' saykes tu'rde int ha'm long tu'rindegil sanlardi' qatarg'a ko'rinisine wo'tkeredi. Bul funksiyalar saykes tu'rde to'mendegi sintaksisga iye:

```

char* itoa(int num, char *str, int radix);

```

ha'm

```

char* ltoa(long num, char *str, int radix);

```

Bul funksiyalar num sani'n radix argumentda ko'rsetilgen sanaq cistyemasidagi ko'rinisin str qatarda payda yetadi'. Mi'sal ushi'n 12345 sani'n ha'r qi'yli' sanaq sistemadagi qatar ko'rinisin payda yetiw ma'selesin ko'reylik:

```

#include <iostream>
#include <stdlib.h>
using namespace std;
int main(int argc, char* argv[])
{
    char qatar2[20], qatar8[15], qatar10[10], qatar16[5];
    int san=12345;
    itoa(san, qatar2, 2) ;
    itoa(san, qatar8, 8) ;
    itoa(san, qatar10, 10) ;
    itoa(san, qatar16, 16) ;
    cout<<"San ko'rinishlari"<<endl;
    cout<<"2 lik sanaq sistemasi'nda :"<<qatar2<<endl;
    cout<<"8 lik sanaq sistemasi'nda :"<<qatar8<<endl;
    cout<<"10 lik sanaq sistemasi'nda:"<<qatar10<<endl;
    cout<<"16 lik sanaq sistemasi'nda:"<<qatar16<<endl;
    return 0;
}

```

Programma ekrang'a to'mendegi qatarlardi' shi'g'aradi':

```

San ko'rinishlari
2 lik sanaq sistemasi'nda :11000000111001
8 lik sanaq sistemasi'nda :30071
10 lik sanaq sistemasi'nda:12345
16 lik sanaq sistemasi'nda:3039

```

gcvt() funksiyasi'

```
char* gcvt(double val, int ndec, char *buf) ;
```

ko'rinistegi prototipga iye boli'p, doublye tu'rindegil val sani'n buf ko'rsetiwshi ASCIIZ qatarg'a aylanti'radi'. Yekinshi argument si'pati'nda beriletugi'n ndec ma'nisi san ko'riniside nomerler mug'dari'n ko'rsetedi. Yeger nomerler sani' ndec ma'nisinen ko'p bolsa, mu'mkinshilik bolsa sannin' bo'lshek bo'legiden arti'qsha nomerler qirqib taslanadi'(pu'tinlengen halda), ha'kis hodda san eksponyensial ko'riniste payda yetiledi. To'mendegi keltirilgen programmada gcvt() funksiyasi'nan paydalani'wdi'n' ha'r qi'yli' variantlari' ko'rsetilgen.

```
#include <iostream>
```

```

#include <stdlib.h>
using namespace std;
int main(int argc, char* argv[])
{
    char qatar[10];
    double san;
    int nomerler_sani=4;
    cout<<"San ko'rinisidagi nomerler sani': ";
    cout<<"tsifrlar_soni"<<endl;
    san=3.154;
    gcvt(san, nomerler_sani, qatar) ;
    cout<<"3.154 sani'ni'n' qatar ko;'rinishi:
" << qatar << endl;
    san=-312.456;
    gcvt(san, nomerler_sani, qatar) ;
    cout<<"-312.456 sani'ni'n' qatar ko'rinishi: ";
    cout<<qatar<<endl;
    san=0.123E+4;
    gcvt(san, nomerler_sani, qatar) ;
    cout<<"0.123E+4 sani'ni'n' qatar ko'rinishi: ";
    cout<<qatar<<endl;
    san=12345.456;
    gcvt(san, nomerler_sani, qatar) ;
    cout<<"12345.456 sani'ni'n' qatar ko'rinishi: ";
    cout<<qatar<<endl;
    return 0;
}

```

Programma ekrang'a izbe-iz tu'rde san ko'rinishlerin baspadan shi'g'aradi':

```

San ko'rinisidagi nomerler sani':4
3.154 sani'ni'n' qatar ko;'rinishi: 3.154
-312.456 sani'ni'n' qatar ko'rinishi:-312.5
0.123E+4 sani'ni'n' qatar ko'rinishi:1230
12345.456 sani'ni'n' qatar ko'rinishi:1.235e+04

```

Soraw ha'm tapsi'rmalar

1. Qatarlar menen islewde qandayda-bir belgi izlew qanday a'melge asi'ri'ladi'?
2. strchr() funksiyasi'ni'n' wazi'ypasi'.
3. Qanday jag'daylarda strpn() funksiyasi'nan paydalani'ladi'?
4. Qatar bo'limlerin izlaydigan qanday funksiyalardi' bilesiz?
5. strstr() funksiyasi' haqqi'nda mag'li'wmat berin'.
6. Tu'rlerdi wo'zgertiw funksiyalari'.
7. Qatardi' sang'a aylanti'riwshi' qanday funksiyalardi' bilesiz?
8. atol() funksiyasi'ni'n' atoi() funksiyasi'nan ayi'rmashi'li'g'i'?
9. Sandi' qatarg'a aylanti'ratug'i'n funksiyalar.
10. itao() ha'm ltao() funksiyalari'ni'n' wazi'ypasi'.

Converting Numeric Strings to Numbers

The C++ standard library includes several functions that allow you to convert the string representation of a number into its internal format. These are **atoi()**, **atol()**, and **atof()**, which convert a numeric string into an integer, long integer, and double floating-point value, respectively. These functions all require the header file **<cstdlib>**. The following program illustrates their use:

```
// Demonstrate atoi(), atol(), and atof().
#include <iostream>
#include <cstdlib>
using namespace std;

int main()
{
    int i;
    long j;
    double k;

    i = atoi("100");
    j = atol("100000");
    k = atof("-0.123");

    cout << i << ' ' << j << ' ' << k;
    cout << '\n';

    return 0;
}
```

The output is shown here:

```
100 100000 -0.123
```

The string-conversion functions are especially useful when passing numeric data to a program through a command line argument. They are also useful in a variety of other programming situations.

20-Lekciya: Qatarlar u'stinde a'meller

Joba:

1. Qatar bo'legin basqa qatarg'a nusqalaw funksiyasi'
2. Qatar bo'legin basqa qatarg'a qosi'w funksiyasi'
3. Qatar bo'legin basqa qatar ishine jaylasti'ri'w funksiyasi'
4. Qatar bo'legin wo'shiriw funksiyasi'
5. Qatar bo'legin almasti'ri'w funksiyasi'
6. Qatar bo'legin aji'rati'p ali'w funksiyasi'

Tayansh so'zler: *Qatar bo'legin basqa qatarg'a nusqalaw funksiyasi', qatar bo'legin basqa qatarg'a qosi'w funksiyasi', qatar bo'legin basqa qatar ishine jaylasti'ri'w funksiyasi', qatar bo'legin wo'shiriw funksiyasi', qatar bo'legin aji'rati'p ali'w funksiyasi'*

C++ tilinin' 32 razryadli' operator platformalar ushi'n jarati'lg'an variantlari'nda, mi'sali, C++ Builder varianti'nda standart qatar tu'rine qosi'msha si'pati'nda `string` tu'ri kiritilgen ha'm wol `string` klasi' ko'rinishinde a'melge asi'ri'lg'an. Bul tu'rdegi qatar ushi'n '\0' belgisi tugash belgisi yesaplanmaydi ha'm wol a'piwayi'g'ana belgiler massivi si'pati'nda qarala-di. `string` tu'rinde qatarlar uzi'nli'g'i'nn' atqari'latug'i'n a'meller na'tiyjesinde dinamikali'q usi'l-de wo'zgerip turi'wi', woni'n' qurami'nda bir qatar funksiyalar ani'qlang'anli'gi' bul tu'r menen islewde ma'lum bir qolayli'qlar jaratadi'.

`string` tu'rindagi wo'zgeriwshiler to'mendegishe dag'aza yetiliwi mu'mkin:

```
string s1, s2, s3;
```

Bul tu'rdegi qatarlar ushi'n arnawli' a'meller ha'm funksiyalar ani'qlang'an.

`string` qatarg'a baslang'i'sh ma'nisler har-xil usi'llar arqali' beriw mu'mkin:

```
string s1="birinshi usi'l";  
string s2="yekinshi usi'l";  
string s3(s2) ;
```

```
string s4=s2;
```

Tap sonday, `string` tu'rindegi wo'zgeriwshiler u'stinde ma'nis beriw a'melleri da har-xil:

```
string s1, s2, s3; char *str="misol";
//satrli turaqli' ma`nisi beriw
S1="Ma'nis beriw birinshi usi'l";
S2=str; //shar tu'rindegi qatar ju'klenmoqda
S3='A'; //bir belgi ma'nis si'pati'nda beriw
S3=s3+s1+s2 +"0123abc"; //qiymat si'pati'nda qatar
an'latpa
```

To'mendegi keste `string` tu'rindegi qatarlar u'stinen a'meller keltirilgen.

8.2-keste. *string* tu'rindegi qatarlar u'stinen a'meller

A'mel	Mazmuni'	Mi'sal
=, +=	Ma'nis beriw a'meli	s="satr01234" s+="2satr000"
+	Qatarlar jalg'aw a'meli(atlar tarti'p ju'retug'i'n tramvayntenatsiya)	s1+s2
=, !=, <, <=, >, >=	Qatarlardi' sali'sti'ri'w a'melleri	s1=s2 s1>s2 && s1!=s2
[]	Indeks beriw	S[4]
<<	Ag'i'mg'a shi'g'ari'w	cout<<s
>>	Ag'i'mdan woqi'w	cin>>s(probelgacha)

Qatar elementine indeks jardeminden ti'sqari' `at()` funksiyasi' arqali' murajat yetiw mu'mkin:

```
string s1="qatar mi'sali";
cout<<s.at(3) //natijada'r' belgisi ekrang'a shi'g'adi'
Soni' ayti'p wo'tiw kerek, string klassta sol tu'rdegi wo'zgeriwshiler
menen isleytug'i'n funksiyalar ani'qlang'an. Basqasha aytqanda, string
```

tu'rinde dag'aza yetilgen wo'zgeriwshiler (ob'yektlar) «wo'z» funksiyalari'na iye yesaplanadi' ha'm wolardi' shaqi'ri'w ushi'n aldi'n wo'zgeriwshi ati', keyin '.' (noqat) ha'm za'ru'r funksiya ati'(argumentleri menen) jazi'ladi'.

Qatar bo'legin basqa qatarg'a nusqalaw funksiyasi: Bir qatar bo'legin basqa qatarg'a ju'klew ushi'n to'mendegi funksiyalardi' isletiw mu'mkin, wolardi' prototipi to'mendegishe:

```
assign(const string &str) ;
assign(const string &str, unsigned int pos,
unsigned int n) ;
assign(const char *str, int n) ;
```

Birinshi funksiya ma'nis beriw a'meli menen ekvivalent boli'p tabi'ladi': string tu'rindegi str qatar wo'zgeriwshi yamasa qatar turaqli'di' a'meldi shaqi'ri'wshi' qatarg'a beredi:

```
string s1, s2;
s1="birinshi qatar";
s2.assign(s1) ; // s2=s1 a'melge ekvivalent
```

Yekinshi funksiya shaqi'ri'wshi' qatarg'a argumentdagi str qatardi'n' pos warni'nan n ta belgidan ibarat bolg'an qatar bo'legin nusqalaydi'. Yegerde ros ma'nisi str qatar uzi'nli'g'i'nan u'lken bolsa, qa'telik haqqi'nda ogohlantiriladi, yeger pos + n an'latpa ma'nisi str qatar uzi'nli'g'i'nan u'lken bolsa, str qatari'ni'n' ros warni'nan baslap qatar aqi'ri'na shekemgi belgiler nusqalanadi'. Bul qag'i'yda barli'q funksiyalar ushi'n ti'yi'sli boli'p tabi'ladi'.

Mi'sal:

```
string s1, s2, s3;
s1="0123456789";
s2.assign(s1, 4, 5) ; //s2="45678"
s3.assign(s1, 2, 20) ; //s3="23456789"
```

U'shinshi funksiya argumentdagi char tu'rindegi str qatardi' string tu'rine aylanti'ri'p, funksiyaning shaqi'ri'wshi' qatarg'a wo'zlestiredi:

```
char * strold;
cin.getline(strold, 100) ;//”0123456789” qatari'
                                kiritiledi

string s1, s2;
s2.assign(strold, 6) ; // s2=”012345”
s3.assign(strold, 20) ; // s3=”0123456789”
```

Qatar bo'legin basqa qatarg'a qosi'w funksiyasi: Qatar bo'legin basqa qatarg'a qosi'w funksiyalari' to'mendegishe:

```
append(const string &str) ;
append(const string &str, unsigned int pos, unsigned int
        n) ;
append(const char *str, int n) ;
```

Bul funksiyalardi' joqari'da keltirilgen saykes assign funksiyalardan ayi'rmashi'li'g'i' - funksiyaning shaqi'ri'wshi' qatar aqi'ri'na str qatardi' wo'zin yamasa woni'n' bo'legin qosadi'.

```
char * sc;
cin.getline(sc, 100) ; //”0123456789” qatari' kiritiledi
string s1, s, s2;
s2=sc; s1=”misol”;
s=”aaa”; //s2=”0123456789”
s2.append(“abcdef”) ;
//s2+=”abcdef” a'meline ekvivalent
//ha'm s2=”0123456789abcdef”
s1.append(s2, 4, 5) ; //s1=”misol45678”
s.append(sc, 5) ; //s=”aaa012345”
```

Qatar bo'legin basqa qatar ishine jaylasti'ri'w funksiyasi: Bir qatarg'a yekinshi qatar bo'legin jaylasti'ri'w ushi'n to'mendegi funksiyalar isletiledi:

```
insert(unsigned int pos1, const string &str) ;
insert(unsigned int pos1, const string & str, unsigned
        int pos2, unsigned int n) ;
```

```
insert(unsigned int pos1, const char *str, int n) ;
```

Bul funksiyalar append si'yaqli' isleydi, ayi'rmashi'li'g'i' sonda, str qatari'n yamasa woni'n' bo'legin funksiyani' shaqi'ri'wshi' qatardi'n' ko'rsetilgen pos1 worni'nan baslap jaylasti'radi'. Bunda a'mel shaqi'ri'wshi' qatardi'n' pos1 wori'ndan keyin jaylasqan belgiler o'nga ji'lji'ti'ladi'.

Mi'sal:

```
char * sc;
cin.getline(sc, 100) ; //”0123456789” qatari' kiritiledi
unsigned int i=3;
string s1, s, s2;
s2=sc; s1=”misollar”; s=”xyz”; //s2=”0123456789”
s2.insert(i, “abcdef”) ; //s2=”012abcdef3456789”
S1.insert(i-1, s2, 4, 5) ; //s1=”mi45678sollar”
s.insert(i-2, sc, 5) ; //s=”x01234yz”
```

Qatar bo'legin wo'shiriw funksiyasi: Qatar bo'legin wo'shiriw ushi'n to'mendegi funksiyani' isletiw mu'mkin:

```
erase(unsigned int pos=0, unsigned n=pos) ;
```

Bul funksiya, woni' shaqi'ri'wshi' qatardi'n' pos worni'nan baslap n ta belgi wo'shirip tashlaydi. Yegerde pos ko'rsatilmasa, qatar basi'nan baslap wo'shiriledi. Yeger n ko'rsatilmasa, qatardi' aqi'ri'na shekemgi belgiler wo'shiriledi:

```
string s1, s2, s3;
s1=”0123456789”
s2=s1;s3=s1;
s1.erase(4, 5) ; //s1=”01239”
s2.erase(3) ; //s2=”012”
s3.erase() ; //s3=””
```

void clear() funksiyasi', woni' shaqi'ri'wshi' qatardi' toli'q tazalaydi.

Mi'sali':

```
s1.clear() ; //qatar bos yesaplanadi'(s1=” “)
```

Qatar bo'legin almasti'ri'w funksiyasi': Bir qatar bo'leginn' worni'na basqa qatar bo'legin qoyi'w ushi'n to'mendegi funksiyalardan paydalani'w mu'mkin:

```
replace(unsigned int pos1, unsigned int n1, const string
        &str) ;
replace(unsigned int pos1, unsigned int n1, const string
        &str, unsigned int pos2, unsigned int n2) ;
replace(unsigned int pos1, unsigned int n1, const char *str,
        int n) ;
```

Bul funksiyalar insert si'yaqli' isleydi, wodan ayri'qsha tu'rde a'mel shaqi'ri'wshi' qatardi'n' ko'rsetilgen worni'nan(pos1) n1 belgiler worni'na str qatari'n yamasa woni'n' pos2 wori'ndan baslang'an n2 belgiden ibarat bo'legin qo'yadi(almasti'radi').

Mi'sal:

```
char * sc="0123456789";
unsigned int i=3, j=2;
string s1, s, s2;
s2=sc; s1="misollar"; s="xyz"; //s2="0123456789"
s2.replace(i, j, "abcdef") ; //s2="012abcdef56789"
S1.replace(i-1, j+1, s2, 4, 5) ; //s1="mi45678ler"
s.replace(i-2, j+2, sc, 5) ; //s="x012345"
```

swap(string &str) funksiyasi' yeki qatarlardi' wo'z-ara almasti'ri'w ushi'n isletiledi. Mi'sali':

```
string s1, s2;
s1="01234";
s2="98765432";
s1.swap(s2) ; // s2="01234" ha'm s1="98765432" boladi'.
```

Qatar bo'legin aji'rati'p ali'w funksiyasi': Funksiya prototipi to'mendegishe:

```
string substr(unsigned int pos=0, unsigned int n=npos)
const;
```

Bul funksiya, woni' shaqi'ri'wshi' qatardi'n' pos worni'nan baslap n belgi na'tiyje si'pati'nda qaytaradi'. Yegerde pos ko'rsatilmasa, qatar basi'nan baslap aji'rati'p ali'nadi', yeger n ko'rsatilmasa, qatar aqi'ri'na shekemgi belgiler na'tiyje si'pati'nda qaytari'ladi':

```
string s1, s2, s3;
s1="0123456789";
s2=s1; s3=s1;
s2=s1.substr(4, 5) ; // s2="45678"
s3=s1.substr(3) ; // s3="3456789"
// "30123456789" qatar ekrang'a shi'g'adi'
cout<<s1.substr(1, 3) +s1.substr() ;
```

Soraw ha'm tapsi'rmalar

1. C++ tilinde bir qatar bo'legin basqa qatarg'a ju'klew.
2. Qatar elementine qanday murajat yetiledi?
3. Append funksiyasi'ni'n' wazi'ypasi'.
4. Qatar bo'legin basqa qatarg'a qosi'w funksiyalari'.
5. Qanday jag'daylarda insert() funksiyasi'nan paydalani'ladi'?
6. Qatar bo'legin wo'shoretug'i'n funksiyalar.
7. void clear() funksiyasi'ni'n' qollani'li'wi'na mi'sal keltiring.
8. replace() funksiyasi'ni'n' wazi'ypasi'.
9. Qatarlardi' wo'z-ara almasti'ri'wda qollani'latug'i'n funksiyalar.
10. Qatar bo'limleri qanday almasti'ri'ladi'?

21-Lekciya: Qatarlar menen isleytug'i'n funksiyalar

Joba:

1. **string** tu'rindegi qatardi' **char** tu'rine wo'tkeriw
2. **Qatar bo'legin izlew funksiyalari'**
3. **Qatarlardi' sali'sti'ri'w**
4. **Qatar wo'zgesheliklerin ani'qlaw funksiyalari'**

Tayansh so'zler: *String* tu'rindegi qatardi' *char* tu'rine wo'tkeriw, qatar bo'legin izlew funksiyalari', qatarlardi' sali'sti'ri'w, qatar wo'zgesheliklerin ani'qlaw funksiyalari'

string tu'rindegi qatardi' **char** tu'rine wo'tkeriw: *string* tu'rindegi qatardi' *char* tu'rine wo'tkeriw ushi'n

```
const char * c_str() const;
```

funksiyani' isletiw kerek. Bul funksiya *char* tu'rdegi '\0' belgisi menen tawsi'latug'i'n qatarg'a turaqli' ko'rsetkishni qaytaradi':

```
char *s1; string s2="0123456789";  
s1=s2.c_str() ;
```

Tap sol maqsette

```
const char * data() const;
```

funksiyasi'nan da paydalani'w mu'mkin. Lekin bul funksiya qatar aqi'ri'na '\0' belgisin qospaydi'.

Qatar bo'legin izlew funksiyalari': *string* klasi'nda qatar bo'legin izlew ushi'n har-xil variantdag'i' funksiyalar ani'qlang'an. To'mende wolardan api'wayi'lerinin' suwretlewin keltiremiz.

```
unsigned int find(const string &str,  
unsigned int pos=0) const;
```

Funksiya, woni' shaqi'rg'an qatardi'n' ko'rsetilgen wori'ndan(pos) baslap *str* qatardi' qi'di'radi' ha'm birinshi saykes keliwshi qatar bo'leginn' baslani'w indeksini juwap si'pati'nda qaytaradi', bolmasa maksimal won'

pu'tin npos sandi' qaytaradi'(npos=4294967295), yeger izlew womi'(pos) berilmasa, qatar basi'nan baslap i'zlenedi.

```
unsigned int find(char c,unsigned int pos=0) const;
```

Bul funksiya aldi'ng'i'nan ayri'qsha tu'rde qatardan c belgisin izlaydi.

```
unsigned int rfind(const string &str,unsigned int  
pos=npow) const;
```

Funksiya, woni' shaqi'rg'an qatardi'n' ko'rsetilgen pos womi'nasha str qatardi'n' birinshi dus kelgen jayi'n indeksini qaytaradi', bolmasa npos ma'nisin qaytaradi', yeger pos ko'rsatilmasa qatar aqi'ri'i'na shekem izlaydi.

```
unsigned int rfind(char c,unsigned int pos=npow) const;
```

Bul funksiyanin' aldi'ng'i'nan ayirmashi'li'g'i'-qatardan c belgisi i'zlenedi.

```
unsigned int find_first_of(const string &str, unsigned  
int pos=0) const;
```

Funksiya, woni' shaqi'rg'an qatardi'n' ko'rsetilgen(pos) jayi'ndan baslap str qatari'ni'n' qa'legen qandayda-bir belgisin qi'di'radi' ha'm birinshi dus kelgeninin' indeksini, bolmasa npos sani'n qaytaradi'.

```
unsigned int find_first_of(char c, unsigned int pos=0)  
const;
```

Bul funksiyanin' aldi'ng'i'nan ayirmashi'li'g'i'-qatardan c belgisin izlaydi;

```
unsigned int find_last_of(const string &str,unsigned int  
pos=npow) const;
```

Funksiya, woni' shaqi'rg'an qatardi'n' ko'rsetilgen(pos) wori'ndan baslap str qatardi' qa'legen qandayda-bir belgisin qi'di'radi' ha'm won' ta'repden birinshi dus kelgeninin' indeksini, bolmasa npos sani'n qaytaradi'.

```
unsigned int find_last_of(char c,unsigned int pos=npow)  
const;
```

Bul funksiya aldi'ng'i'nan ayirmashi'li'g'i'-qatardan c belgisin izlaydi;

```
unsigned int find_first_not_of(const string &str,
                               unsigned int pos=0) const;
```

Funksiya, woni' shaqi'rg'an qatardi'n' ko'rsetilgen(pos) wori'ndan baslap str qatardi'n' qandayda-bir da belgisi kirmaydigan qatar bo'legin qi'di'radi' ha'm shep ta'repden birinshi dus kelgeninin' indeksini, bolmasa npos sani'n qaytari'ladi'.

```
unsigned int find_first_not_of(char c, unsigned int
                               pos=0) const;
```

Bul funksiyani'n' alding'i'nan ayirmashi'li'g'i'-qatardan c belgisinen ayri'qsha birinshi belgi izlaydi;

```
unsigned int find_last_of(const string &str, unsigned
                          int pos=npow) const;
```

Funksiya, woni' shaqi'ri'wshi' qatardi'n' ko'rsetilgen wori'ndan baslap str qatari'n quraytug'i'n belgiler kompleksine kirmagan belgi qi'di'radi' ha'm yen' won' ta'repden birinshi topilgan belginin' indeksini, bolmasa npos sani'n qaytaradi'.

```
unsigned int find_last_of(char c, unsigned int pos=npow)
                          const;
```

Bul funksiyani'n' alding'i'nan ayirmashi'li'g'i'-qatar aqi'ri'nan baslap c belgisine qusamag'an belgi izlaydi.

Izlew funksiyalari'n qo'llawg'a mi'sal:

```
#include <iostream>
#include <string.h>
using namespace std;
int main(int argc, char* argv[])
{
    string s1="01234567893456ab2csef", s2="456", s3="ghk2";
    int i, j;
    i=s1.find(s2) ;
    j=s1.rfind(s2) ;
    cout<<i; // i=4
    cout<<j; // j=11
    cout<<s1.find('3') <<endl; // na'tiyje 3
```

```

cout<<s1.rfind('3') <<endl; // na'tiyje 10
cout<<s1.find_first_of(s3) <<endl; // na'tiyje 2
cout<<s1.find_last_of(s3) <<endl; // na'tiyje 16
cout<<s1.find_first_not_of(s2) <<endl; // na'tiyje 14
cout<<s1.find_last_not_of(s2) <<endl; // na'tiyje 20
}

```

Qatarlardi' sali'sti'ri'w: Qatarlar bo'limlerin sali'sti'ri'w ushi'n compare funksiyasi' isletiledi:

```

int compare(const string &str) const;
int compare(unsigned int pos1, unsigned int n1, const
             string & str) const;
int compare(unsigned int pos1, unsigned int n1, const
             string & str, unsigned int pos2, unsigned int
             n2) const;

```

Funksiyani'n' birinshi formasi'nda yeki qatarlar toli'q sali'wti'ri'ladi': funksiya kerri san qaytaradi', yeger funksiyanii' shaqi'ri'wshi' qatar str qatardan kishi bolsa, 0 qaytaradi' yeger wolar ten' bolsa ha'm won' san qaytaradi', yeger funksiya shaqi'ri'wshi' qatar str qatardan u'lken bolsa.

Yekinshi formada tap birinshi si'yaqli' a'meller atqari'ladi', tek funksiya shaqi'ri'wshi' qatardi'n' pos1 warni'nan baslap n1 dana belgili qatar asti' str qatar menen sali'wti'ri'ladi'.

U'shinshi ko'riniste funksiya shaqi'ri'wshi' qatardi'n' pos1 warni'nan baslap n1 dana belgili qatar bo'legi ha'm str qatardan pos2 warni'nan baslap n2 dana belgili qatar bo'limleri wo'z-ara sali'wti'ri'ladi'.

Mi'sal:

```

#include <iostream>
#include <string.h>
using namespace std;
int main(int argc, char* argv[])
{
    string s1="01234567893456ab2csef", s2="456", s3="ghk";
    cout<<"s1="<<s1<<endl;
    cout<<"s2="<<s2<<endl;
    cout<<"s3="<<s3<<endl;
}

```

```

    if(s2.compare(s3) >0)
        cout<<"s2>s3"<<endl;
    if(s2.compare(s3) ==0)
        cout<<"s2=s3"<<endl;
    if(s2.compare(s3) <0)
        cout<<"s2<s3"<<endl;
    if(s1.compare(4, 6, s2) >0)
        cout<<"s1[4-9]>s2"<<endl;
    if(s1.compare(5, 2, s2, 1, 2) ==0)
        cout<<"s1[5-6]=s2[1-2]"<<endl;
    return 0;
}

```

Ma'sele. Fami'li'ya, ati' ha'm shariflari menen studentler dizimi berilgen. Dizim alfavit boyi'nsha ta'rtilansin. Programma teksti:

```

#include <iostream.h>
#include <alloc.h>
int main(int argc, char* argv[])
{
    const int FISH_uzunligi=50;
    string* Student;
    char * Qatar=(char*) malloc(FISH_uzunligi) ;
    unsigned int studentler_soni;
    char san[3];
    do
    {
        cout<<"Talabalar sani'n kiritin': ";
        cin>>son;
    }
    while((() studentler_soni=atoi(san) ) <=0) ;
    Student =new string[studentler_soni];
    cin.ignore() ;
    for(int i=0; i<studentler_soni; i++)
    {
        cout<<i+1<<"-studenttin' Familya ati' sharifi: ";
        cin.getline(qatar, 50) ;
    }
}

```

```

Student.assign(Qatar) ;
}
bool almasdi=true;
for(int i=0; i<talabalar_soni-1 && almasdi'; i++)
{
    almasdi=false;
    for(int j=i; j<talabalar_soni-1; j++)
        if(Student[j].compare(Student[j+1]) >0)
        {
            almasdi=true;
            strcpy(qatar, Student[j].data() ) ;
            Student[j].assign(Student[j+1]) ;
            Student[j+1].assign(Qatar) ;
        }
}
    cout<<"Alfavit boyi'nsha ta'rtpiengen dizim:\n";
for(int i=0; i<talabalar_soni; i++)
    cout<<Student<<endl;
delete [] Student;
free(Qatar) ;
return 0;
}

```

Programmada studentler dizimi string tu'rindegi Student dinamikali'q massiv ko'risinde dag'aza yetilgen ha'm woni'n' wo'lshemi paydalani'wshi' ta'repinen kiritilgen studentlat_soni menen ani'qlanadi'. Studentler sani'n kirgiziwde qadag'alaw yetiledi: klaviaturadan qatar woqi'ladi' ha'm wol atoi() funksiyasi' ja'rdeminde sang'a aylantiriladi. Yeger payda bolg'an san nolden u'lken san bo'lmasa, sandi' kirgiziw procesi ta'kirarlanadi'. Studentler sani' ani'q bolg'andan keyin ha'r bir studenttin' familia, ati' ha'm sharifi bir qatar si'pati'nda ag'i'mdan woqi'ladi'. Keyin, string tu'rinde ani'qlang'an compare() funksiyasi' ja'rdeminde massivdegi qatarlar wo'z-ara sali'wti'ri'ladi' ha'm saykes wori'ndagi' belgiler kodlari'n wo'siwi boyi'nsha «pufakchali saralaw» arqali' ta'rtpienedi. Programma aqi'ri'nda payda bolg'an massiv baspadan shi'g'ari'ladi', ha'mde dinamikali'q massivler joyti'ladi'.

Qatar wo'zgesheliklerin ani'qlaw funksiyalari: string klasi'nda qatar uzi'nli'g'i, woni'n' bosli'g'in yamasa iyelegen yad ko'lemin ani'qlaytug'i'n funksiyalar bar:

```
unsigned int size() const; //qatar wo'lshemi
unsigned int length() const; //qatar elementler sani'
unsigned int max_size() const; //qatardin' maksimal
//uzunligi(4294967295)
unsigned int capacity() const; //qatar iyelegen yad
// ko'lemi
bool empty() const; //true, yeger qatar bos bolsa
```

Soraw ha'm tapsi'rmalar

1. *find() ha'm rfind() funksiyalari' qanday wazi'ypani' atqaradi'?*
2. *Qatarlardi' sali'sti'ri'wg'a mi'sallar keltiring.*
3. *Qatarlardi' sali'sti'ri'wda qollani'latug'i'n funksiyalar.*
4. *compare funksiyasi'ni'n' wazi'ypasi'.*
5. *Qatardi'n' wo'zgesheliklerin ani'qlaytug'i'n qanday funksiyalardi' bilesiz?*
6. *size() funksiyasi' haqqi'nda mag'li'wmat berin'.*
7. *Qanday jag'daylarda max_size() funksiyasi'nan paydalani'ladi'?*
8. *Qatar iyelegen yad ko'lemi qanday ani'qlanadi'?*
9. *compare funksiyasi'ni'n' atqari'li'wi'na mi'sal keltiring.*
10. *capacity() funksiyasi' qanday maqsette isletiledi?*

22-Lekciya: A'piwayi' strukturalar

Joba:

1. **Strukturalar**
2. **Struktura funksiya argumenti si'pati'nda**
3. **Strukturalar massivi**
4. **Strukturalarg'a ko'rsetkish**

Tayansh so'zler: *Strukturalar, struktura funksiya argumenti si'pati'nda, strukturalar massivi, Strukturalarga ko'rsetkish*

Strukturalar: Qandayda bir predmet ken'isligindag'i' ma'seleni sheshiwde wodag'i' obyektler bir qansha, har-xil tu'rdege parametrler menen ani'qlani'wi' mu'mkin. Mi'sali', tekislikdagi noqat haqi'yqi'y tu'rdegi x-absissa ha'm y-ordinata jupligi - (x, y) ko'rinishinde beriledi. Student haqqi'ndag'i' mag'li'wmatlar: qatar tu'rindagi student fami'li'ya, ati' qa'nigelik bag'dari, student jasaw adresi, pu'tin tu'rdegi tuwi'lg'an ji'li', woqi'w basqi'shi', haqi'yqi'y tu'rdegi reyting bali, logikali'q tu'rdegi student jinsi haqqi'ndag'i' mag'li'wmat ha'm basqalardan qa'liplese.

Programmada jag'day yamasa tu'sinikti suwretlewleytug'i'n ha'r bir berilgenler ushi'n bo'lek wo'zgeriwshi ani'qlap ma'seleni sheshiw mu'mkin. Lekin bul halda obyekt haqqi'ndag'i' mag'li'wmatlar «tarqoq» boladi', wolardi' qayta islew quramali'lasadi', obyekt haqqi'ndag'i' berilgenlerdi pu'tin halda ko'riw qi'yi'nlasadi'.

C++ tilinde bir yamasa har-xil tu'rdegi berilgenlerdi kompleksi struktura dep ataladi'. Struktura paydalani'wshi' ta'repinen ani'qlang'an berilgenlerdin' jan'a tu'ri yesaplanadi', Struktura to'mendegishe ani'qlanadi':

```
struct <struktura ati'>
{
    <tu'r1 > <ati'1>;
    <tu'r2 > <ati'2>;
    ...
    <tu'rn > <ati'n>;
}
```

```
};
```

Bul jerde <struktura nomi> - struktura ko'riniside jarati'li'p ati'rg'an jan'a tu'rdin' ati', <tu'ri> <ati'i> - strukturani'n' i-maydani'ni'n' (ati') dag'azasi'.

Basqasha aytqanda, struktura dag'aza yetilgen wo'zgeriwshilerden (maydanlardan) tashkil tabadi'. Wog'an har-xil tu'rdegi berilgenlerdi wo'z ishine ali'wshi' qabi'q dep qaraw mu'mkin. Qabi'qdag'i' berilgenlerdi pu'tin halda ko'shiriw, si'rtqi' qurallar (binar fayllarg'a) jazi'w, woqi'w mu'mkin boladi'.

Student haqqi'ndag'i' berilgenlerdi wo'z ishine ali'wshi' struktura tu'rinin' dag'aza yetiliwin ko'reylik.

```
struct Student
{
    char FISH[30];
    unsigned int Tuw_jil;
    unsigned int Kurs;
    char Bagdar[50];
    float Reyting;
    unsigned char Jinsi[5];
    char Manzil[50];
    bool status;
};
```

Programmada strukturalardan paydalani'w, sol tu'rdegi wo'zgeriwshiler ja'riyalaw ha'm wolardi' qayta islew arqali' a'melge asi'ri'ladi':

```
Student student;
```

Struktura tu'rinin' dag'azasi'nda tu'rdin' ati' bolmaytug'i'ni' mu'mkin, lekin bul halda struktura ani'qlani'wi'nan keyin a'l'bette wo'zgeriwshiler atlari' jazi'li'wi' kerek:

```
struct
{
    unsigned int x, y;
    unsigned char Ren';
} Noqat1, Noqat2;
```

Keltirilgen mi'salda struktura tu'rindegi Noqat1, Noqat2 wo'zgeriwshileri dag'aza yetilgen.

Struktura tu'rindegi wo'zgeriwshiler menen islew, woni'n' maydanlari' menen islewdi an'latadi'. Struktura maydani'na murajat yetiw '.' (noqat) arqali' a'melge asi'ri'ladi'. Bunda struktura tu'rindegi wo'zgeriwshi ati', wodan keyin noqat qo'yi'ladi' ha'm maydan wo'zgeriwshisinin' ati' jazi'ladi'. Mi'sali, student haqqi'ndag'i' struktura maydanlari'na murajat to'mendegishe boladi':

```
student.Kurs=2;
student.Tuw_jil=1988;
strcpy(student.FISh, "Abdullayev A.A.") ;
strcpy(student.Bagdar, "A'meliy matematika ha'm
                    informatika");
strcpy(student.Jinsi, "Erkek") ;
strcpy(student.Manzil, "Tashkent, Yunusobod-8, tel:6-24-78");
student.Reyting=123.52;
```

Keltirilgen mi'salda student strukturasi'ni'n' san tu'rindegi maydanlari'na a'piwayi' ko'riniste ma'nisler berilgen, qatar tu'rindegi maydanlar ushi'n strcpy funksiyasi' arqali' ma'nis beriw a'melge asi'ri'lg'an.

Struktura tu'rindegi obyektin' yaddan qansha jay iyelegenligin sizeof funksiyasi' (operatori) arqali' ani'qlaw mu'mkin:

```
int i=sizeof(Student);
```

Ayi'ri'm jag'daylarda struktura maydanlari' wo'lshemin bi'ytlarda ani'qlaw arqali' iyeletu'g'i'n' yadni' kemeytiw mu'mkin. Woni'n' ushi'n struktura maydani' to'mendegishe dag'aza yetiledi:

```
<maydan ati'> : <wo'zgermes an'latpa>
```

Bul yerda <maydan ati'> - maydan tu'ri ha'm ati', <wo'zgermes an'latpa> - maydanni'n' baytlardag'i' uzi'nlig'i'. Maydan tu'ri pu'tin tu'rler boli'wi' kerek (int, long, unsigned, char).

Yeger paydalani'wshi' strukturani'n' maydani' tek 0 ha'm 1 ma'nisin qabi'llawi'n bilsa, bul maydan ushi'n bir bit jay aji'rati'wi' mu'mkin (bir yamasa yeki bayt worni'na). Yadti' tejew worni'na maydan u'stinde a'mel wori'nlawda razryadli' arifmetikani qo'llaw za'ru'r boladi'.

Mi'sal ushi'n sa'ne-waqi't menen baylani'sli' strukturani' jarati'wdi'n' yeki varianti'n ko'reylik. Struktura ji'l, ay, ku'n, saat, minut ha'm sekund maydanlari'nan ibarat bo'lsin ha'm woni' to'mendegishe ani'qlaw mu'mkin:

```
struct Sane_waqt
{
    unsigned short Jil;
    unsigned short Ay;
    unsigned short Kun;
    unsigned short Saat;
    unsigned short Minut;
    unsigned short Sekund;
};
```

Bunday ani'qlawda Sane_waqt strukturasi' yadta 6 maydan*2 bayt=12 bayt jay iyeleydi. Yeger itibar berilse strukturada awi'si'q jay iyelegen jag'daylar bar. Mi'sali', ji'l ushi'n ma'nisi 0 sani'nan 99 sani'i'na shekem ma'nis menen ani'qlani'wi' jeterli (mi'sali', 2008 ji'ldi' 8 ma'nisi menen an'lati'w mu'mkin). Sol sebepli wog'an 2 bayt yemes, ba'lki 7 bayt aji'rati'w jeterli. Tap sonday ay ushi'n 1..12 ma'nisleri'n an'lati'wg'a 4 bayt jay jeterli ha'm t.b..

Joqari'dag'i' keltirilgen sheklewlerden keyin sa'ne-waqi't strukturasi'n u'nemli varianti'n ani'qlaw mu'mkin:

```
struct Sane_waqt2
{
    unsigned Jil: 7;
    unsigned Ay: 4;
    unsigned Kun: 5;
    unsigned Saat: 6;
    unsigned Minut: 6;
    unsigned Sekund: 6;
};
```

Bul struktura yaddan 5 bayt jay iyeleydi.

Struktura funksiya argumenti si'pati'nda: Strukturalar funksiya argumenti si'pati'nda isletiliwi mu'mkin. Woni'n' ushi'n funksiya prototipida struktura tu'ri ko'rsetiliwi kerek boladi'. Mi'sali', student haqqi'ndagi' berilgenlerdi wo'z ishine ali'wshi' Student strukturasi' tu'rindegi berilgenlerdi Student_Adresi() funksiyasi'na parametr si'pati'nda beriw ushi'n funksiya prototipi to'mendegi ko'riniste boli'wi' kerek:

```
void Student_Adresi(Student);
```

Funksiyag'a strukturani' argument si'pati'nda uzati'wg'a mi'sal si'pati'ndagi' programmanin' teksti:

```
#include <iostream>
#include <string.h>
using namespace std;
struct Student
{
    char FISH[30];
    unsigned int Tuw_jil;
    unsigned int Kurs;
    char Bagdar[50];
    float Reyting;
    char Jinsi[5];
    char Manzil[50];
    bool status;
};
void Student_Adresi(Student) ;
int main(int argc, char* argv[])
{
    Student student;
    student.Kurs=2;
    student.Tuw_jil=1988;
    strcpy(student.FISH, "Abdullayev A.A.") ;
    strcpy(student.Bagdar,"A'meliy matematika ha'm
informatika");
    strcpy(student.Jinsi, "Erkek") ;
    strcpy(student.Manzil, "Tashkent,Yunusobod-8,tel:6-24-
78");
```

```

    student.Reyting=123.52;

    Student_Adresi(student) ;
    return 0;
}
void Student_Adresi(Student t)
{
    cout<<"Student FIO: "<<t.FIO<<endl;
    cout<<"Adresi: "<<t.Manzil<<endl;
}

```

Programma bas funksiyasi'nda student strukturasi' ani'qlanip, woni'n' maydanlari'na ma'nisler beriledi. Keyin student strukturasi' Student_Adresi() funksiyasi'na argument si'pati'nda uzati'ladi'. Programma islewi na'tiyjesinde ekrang'a to'mendegi mag'li'wmatlar baspadan shi'g'ari'ladi'.

Student FIO: Abdullayev A.A.

Adresi: Tashkent, Yunusobod-8, tel:6-24-78

Strukturalar massivi: Wo'z-o'zinen ma'lum, struktura tu'rindegi berilgen menen sheshiw mu'mkin bolg'an ma'seleler shen'beri ju'da' tar ha'm ko'pshilik jag'daylarda, qoyi'lg'an ma'sele strukturalar kompleksi menen islewdi talap yetedi. Bul tu'rdegi ma'selelerge berilgenler bazasin qayta islew ma'seleleri dep qaraw mu'mkin.

Strukturalar massivini ja'riyalaw tap standart massivlerdi ja'riyalawday, ayi'rmashi'li'g'i' massiv tu'ri worni'nda paydalani'wshi' ta'repinen ani'qlang'an struktura tu'rinin' ati' jazi'ladi'. Mi'sali', studentler haqqi'ndagi' berilgenlerdi wo'z ishine alg'an massiv jarati'w dag'azasi' to'mendegishe boladi':

```
const int n=25;
```

```
Student studentler[n];
```

Strukturalar massivini' elementlerine murajat a'detdegi massiv elementlerine murajat usi'llari' arqali', ha'r bir elementinin' maydanlari'na murajat bolsa '.' arqali' a'melge asi'ri'ladi'.

To'mendegi mi'salda studentler topari'ndagi' ha'r bir student berilgenlarini klaviaturadan kirgiziw ha'm gruppada studentlerin fami'li'ya, ati' ha'm sharifini baspadan shi'g'aratug'i'n programma teksti keltirilgen.

```
#include <iostream>
#include <string.h>
#include <conio.h>
using namespace std;
const int n=3;
struct Student
{
    char FISH[30];
    unsigned int Tuw_jil;
    unsigned int Kurs;
    char Bagdar[50];
    float Reyting;
    char Jinsi[6];
    char Manzil[50];
    bool status;
};
Student studentler[n];
void Student_Kirgiziw() ;
void Student_FISH() ;

int main(int argc, char* argv[])
{
    Student_Kirgiziw() ;
    Student_FISH() ;
    return 0;
}
void Student_FISH()
{
    for(int i=0; i<n; i++)
        cout<<i+1<<" " <<studentler[i].FISH<<endl;
}
void Student_Kirgiziw()
{
    for(int i=0; i<n; i++)
    {
```

```

        cout<<i+1<<"-student mag'li'wmatlari'n
kiritin':"<<endl;
        cout<<" Student FISH :";
        cin.getline(studentler[i].FISH, 30) ;
        cout<<" Kurs:";
        cin>>studentler[i].Kurs;
        cout<<" Reyting bali:";
        cin>>studentler[i].Reyting;cout<<"tug''ilgan
ji'li':"";
        cin>>studentler[i].Tuw_jil;
        cout<<"Ta'lim_bag'dari':"";
        cin.getline(studentler[i].Bagdar, 50) ;
        cout<<" Jinsi (erkek, hayal) :";
        cin.getline(studentler[i].Jinsi, 6) ;
        cout<<" Jasaw adresi:";
        cin.getline(studentler[i].Manzil, 50) ;
    }
}

```

Soraw ha'm tapsi'rmalar

1. C++ tilinde struktura ne?
2. C++ tilinde birlespe ne?
3. Strukturani' funksiya argumenti si'pati'nda isletiw.
4. Struktura tu'rindegi wo'zgeriwshilerdi ja'riyalaw.
5. Struktura tipidagi wo'zgeriwshiler yaddan qansha jay iyeleydi?
6. Strukturalar funksiya argumenti si'pati'nda.
7. Strukturalar massivi qanday dag'aza yetiledi?
8. Struktura massivinin' elementlerine murajat qanday a'melge asi'ri'ladi'?
9. C++ tilinde strukturalarg'a ko'rsetkish.
10. Strukturag'a ko'rsetkish wo'zgeriwshisi qanday dag'aza yetiledi?
11. Strukturali' ko'rsetkishlerge mi'sal keltiring.

Structures

In C++, a *structure* is a collection of variables that are referenced under one name, providing a convenient means of keeping related information together. Structures are called *aggregate data types* because they consist of several different, yet logically connected, variables. You will also see structures referred to as *compound* or *conglomerate* data types, for the same reason.

Before a structure object can be created, the form of the structure must be defined. This is accomplished by means of a structure declaration. The structure declaration determines what type of variables the structure contains. The variables that comprise the structure are called *members* of the structure. Structure members are also commonly referred to as *elements* or *fields*.

Generally, all members of the structure will be logically related to each other. For example, structures are typically used to hold information such as mailing addresses, compiler symbol tables, library card catalog entries, and the like. Of course, the relationship between the members of a structure is purely subjective, and thus determined by you. The compiler doesn't know (or care).

The general form of a structure declaration is shown here:

```
struct struct-type-name {  
    type element_name1;  
    type element_name2;  
    type element_name3;  
    .  
    .  
    .  
    type element_nameN;  
} structure-variables;
```

Accessing Structure Members

Individual structure members are accessed through the use of a period (generally called the "dot" operator). For example, the following code will assign the value 10.39 to the **cost** field of the structure variable **inv_var**, declared earlier.

```
inv_var.cost = 10.39;
```

The structure variable name, followed by a period and the member name, refers to that member. All structure elements are accessed in the same way. The general form is

structure-varname.member-name

Therefore, to print **cost** on the screen, you could write

```
cout << inv_var.cost;
```

Arrays of Structures

Structures may be arrayed. In fact, structure arrays are quite common. To declare an array of structures, you must first define a structure, then declare an array of its type. For example, to declare a 100-element array of structures of type **inv_type** (defined earlier), you would write

```
inv_type invtry[100];
```

To access a specific structure within an array of structures, you must index the structure name. For example, to display the **on_hand** member of the third structure, you would write

```
cout << invtry[2].on_hand;
```

Like all array variables, arrays of structures begin their indexing at zero.

When a structure is used as an argument to a function, the entire structure is passed by using the standard call-by-value parameter passing mechanism. This, of course, means that any changes made to the contents of the structure inside the function to which it is passed do not affect the structure used as an argument. However, be aware that passing large structures can incur significant overhead. (As a general rule, the more data passed to a function, the longer it takes.)

23-Lekciya: Dinamikali'q strukturalar

Joba:

1. **Strukturalarda ko'rsetkish**
2. **Dinamikali'q strukturalar**
3. **Birlespeler ha'm wolar u'stinde a'meller**
4. **Paydalani'wshi' ta'repinen ani'qlang'an berilgenler tu'ri**

Tayansh so'zler: *Strukturalarg'a ko'rsetkish, dinamikali'q strukturalar, birlespeler ha'm wolar u'stinde a'meller, paydalani'wshi' ta'repinen ani'qlang'an berilgenler tu'ri*

Strukturalarda ko'rsetkish: Struktura elementlerine ko'rsetkishler arqali' murajat yetiw mu'mkin. Woni'n' ushi'n strukturag'a ko'rsetkish wo'zgeriwshisi dag'aza yetiliwi kerek. Mi'sali', joqari'da keltirilgen mi'salda Student strukturasina ko'rsetkish to'mendegishe dag'aza yetiledi:

```
Student * k_talaba;
```

Ko'rsetkish arqali' ani'qlang'an struktura elementlerine murajat «.» menen yemes, ba'lki -> jardeminde a'melge asi'ri'ladi':

```
cout<<k_talaba->FISh;
```

Strukturalardi ko'rsetkish ha'm murajat (&) jardeminde funksiya argumenti si'pati'nda uzati'w mu'mkin. To'mende keltirilgen programma bo'leginde strukturani' Student_Kiritiw() funksiyasi'na ko'rsetkish arqali', Student_FISh() funksiyasi'na murajat arqali' uzati'wg'a mi'sal keltirilgen.

...

```
void Student_Kirgiziw(Student *t) ;  
void Studentler_FISh(Student & t) ;  
int main( )  
{  
    Student * k_student;  
    k_talaba=(Student*) malloc(n*sizeof(Student) ) ;  
    Student_Kirgiziw(k_student) ;  
    Studentler_FISh(*k_student) ;  
    return 0;
```

```

}
void Studentler_FISh(Student & t) ;
{
    for(int i=0; i<n; i++) ;
        cout<<(&t+i). ->FISh<<endl;
}
void Student_Kirgiziw(Student *t) ;
{
    for(int i=0; i<n; i++) ;
    {
        cout<<i+1<<"-student mag'li'wmatlari'n
            kiritin':"<<endl;
        cout<<" Student FISh :";
        cin.getline(t+i]->FISh, 30) ;
        cout<<" Kurs:";
        cin>>(t+i) ->Kurs;
        ...
    }
}

```

Sog'an itibar beriw kerek, dinamikali'q tu'rde payda yetilgen strukturalar massivi elementi bolg'an strukturani'n' maydani'na murajatda * belgisi qollanilmaydi.

Ma'sele. Futbol komandalari haqqi'ndag'i' mag'li'wmatlar - komanda ati', usi' waqi'ttag'i' tabi'slar, ten'be-ten' ha'm jen'iliwler sanlari', ha'mde raxip da'rwazasi'na kiritilgen ha'm wo'z da'rwazasi'nan wo'tkerip jiberilgen toplar sanlari' menen berilgen. Futbol komandalarinin' turnir kestesini baspadan shig'arilsin. Komandalardi kestedede ta'rtplestiriwde to'mendegi qag'i'ydalarg'a a'mel yetin'sin:

- 1) komandalar toplag'an ochkolarin azayi'wi' boyi'nsha ta'rtplestiriwi kerek;
- 2) yeger komandalar toplag'an ochkolari ten' bolsa, wolardan sali'sti'rg'anda ko'p jen'iske yerisken komanda kestedede joqari' wori'ndi' iyeleydi;

3) yeger yeki komandanin' top lag'an ochkolari ha'm jen'isler sani' ten' bolsa, woldan sali'sti'rg'anda ko'p top kirgizgen komanda kestedede joqari' wori'ndi' iyeleydi.

Komanda haqqi'ndag'i' berilgenler struktura ko'rinishinde, keste bolsa struktura massivi sapasi' ani'qlanadi':

```
struct Jamaat
{
    string Ati;
    int Tabis, Tenbe-ten', Maglub, Urgan_top, Otkezgen_top;
    int Oyin, Ochko;
};
```

Bul yerda Oyin maydani' Tabi's, Ten'be-ten' ha'm Maglub maydanlar ji'yi'ndi'si', komanda top lag'an ochkolar - $Ochko=3*Tabi's+1*Ten'be-ten'$ ko'rinishinde ani'qlanadi'. Komandalar massivi Ochko, Tabi's ha'm Urgan_top maydanlari' boyi'nsha ta'rtiplenedi.

Programma teksti:

```
#include <iostream>
#include <string.h>
#include <conio.h>
using namespace std;
struct Komanda
{
    string Ati;
    int Tabis, Tenbe_ten, Maglub, Urgan_top, Otkezgen_top;
    int Uyin, Ochko;
};
const int at_uzinligi=10;
int komandalar_sani;
Komanda *Komandalar_Kestesi()
{
    char *km=(char*) malloc(at_uzinligi+1) ;
    cout<<" Jamolar sani': ";
    cin>>komandalar_sani;
    Komanda *km=new Komanda[komadalar_sani];
```

```

for(int i=0; i<komadalar_sani; i++) ;
{
cin.ignore() ;
cout<<i+1<<"-Komanda mag'li'wmatlari':\n";
cout<<" Ati': ";
cin.getline(km.Ati, at_uzinligi) ;
while(strlen(km.Ati) <at_uzinligi)
strcat(km, " ") ;
km[i].Ati.assign(km) ;
cout<<" Tabislar sani': ";
cin>> km[i].Tabis;
cout<<" Tenbe_tenler sani': ";
cin>> km[i].Tenbe_ten;
cout<<" Jen'iliwler sani': ";
cin>> km[i].Maglub;
cout<<" Raxip da'rwazasi'na uri'lg'an toplar sani': ";
cin>> km[i].Urgan_top;
cout<<" Wo'z da'rwazasi'na wo'tkergen toplar sani': ";
cin>> km[i].Utkazgan_top;
km[i].Uyin=km[i].Tabis + km[i].Maglub;
km[i].Ochko=km[i].Tabis*3+km[i].Tenbe_ten;
}
free(km) ;
return km;
}
void Utkazish(Komanda &Komanda1, const Komanda &Komanda2)
{
jamoal.Nomi+jamoal2.Ati;
Komanda1.Tabis+jamoal2.Tabis;
Komanda1.Tenbe_ten+jamoal2.Tenbe_ten;
Komanda1.Maglub+jamoal2.Maglub;
jamoal.Urgan_top+jamoal2.Urgan_top;
Komanda1.Otkezgen_top+jamoal2.Otkezgen_top;
jamoal.Uyin=jamoal2.uyin;
Komanda1.Ochko+jamoal2.Ochko;
}
Komanda *Kesteni_Tartiplew(Komanda * km) ;
{

```

```

bool urin_almashdi=true;
for(int i=0; i<komadalar_sani-1 && uri_almashdi; i++) ;
{
Komanda Waqtinsha;
urin_almashdi=false;
for(int j=0; j<komadalar_sani-1; j++) ;
{
// j-Komandanin' ochkosi(j+1) -Komanda ochkosidan u'lken
// bolsa, ta'kirarlawdi'n' keyingi qa'demine o'tilsin.
if(km[j].Ochko>km[j+1].Ochko) continue;
// j ha'm(j+1) -Komandalardin' ochkolari ten' j-Komanda
// jetiskenlikleri(j+1) -jetiskenlikleridam ko'p bolsa,
// ta'kirarlawdi'n' keyingi qa'demine o'tilsin.
if(km[j].Ochko==km[j+1].Ochko) &&
km[j].Tabis>km[j+1].Tabis) continue;
// j ha'm(j+1) -Komandalardin' ochkolari ha'm
jetiskenlikleri sani'
// ten' ha'm j-Komanda o'rgan toplar sani'(j+1) -Komanda
// o'rgan toplardan ko'p bolsa, ta'kirarlawdi'n' keyingi
// qa'demine o'tilsin.
if(km[j].Ochko==km[j+1].Ochko) &&
km[j].Tabis==km[j+1].Tabis) &&
km[j].Urgan_top>km[j+1].Urgan_top) continue;
// joqari'dag'i' sha'rtlerdin' qandayda-biri' da
atqari'lmasa,
// j(j+1) -Komandalar Wori'nlarini' almasti'rilsin.
urin_almashdi=true;
Utkazish(Waqtinsha, km[j]) ;
Utkazish(km[j], km[j+1]) ;
Utkazish(km[j+1], Waqtinsha) ;
}
}
return km;
}
void Kesteni_Chop_YetiW(const Komanda *km)
{
char pr=' ';
cout<<" FUTBOL KOMANDALARI'NIN' TURNIR KESTESI \n ";

```

```

cout<<"-----\n";
cout<<"| JAMOA | O | O' | D | M | UrT| O'T|OCHKO|\n";
cout<<"-----\n";
for(int i=0; i<komadalar_sani; i++) ;
{
cout<<"| "<<km[i].Ati.substr(0, 10) ;cout<<' |';
if(km[i].Uyin<10) cout<<pr;cout<<km[i].Uyin<<" |";
if(km[i].Tabis<10) cout<<pr;cout<<km[i].Tabis<<" |";
if(km[i].Tenbe_ten) cout<<pr;
cout<<km[i].Tenbe_ten<<" |";
if(km[i].Maglub) cout<<pr;cout<<km[i].Maglub<<" |";
if(km[i].Urgan_top<10) cout<<pr;
cout<<km[i].Urgan_top<<" |";
if(km[i].Otkezgen_top<10) cout<<pr;
cout<<km[i].Otkezgen_top<<" |";
if(km[i].Ochko<10) cout<<pr;
cout<<km[i].Ochko<<" |";<<endl;
}
cout<<"-----\n";
}

int main()
{
Komanda *jamoas;
jamoas=Berilgenlerni_kiritish() ;
jamoas=Kesteni_Tartiplew(Komanda) ;
Kesteni_Chop_Yetiw(Komanda) ;
return 0;
}

```

Programma bas funksiya ha'm to'mendegi wazi'ypalardi' wori'nlawshi' to'rtew funksiyadan sho'lkemlesken:

1) Komanda *Komandalar_jadvali() -komandalar haqqi'ndag'i' berilgenlerdi saqlaytug'i'n Komanda strukturalari'nan sho'lkemlesken dinamikali'q massiv jaratadi' ha'm wog'an ag'i'mdan ha'r bir komanda berilgenlerdi woqi'p jaylasti'radi'. Payda bolg'an massivke ko'rsetkishni funksiya na'tiyjesi si'pati'nda qaytaradi';

2) Komanda*jadvalni_Ta'rtiplew(Komanda*jm) -argument arqali ko'rsetilgen massivti ma'sele sha'rti boyi'nsha ta'rtipleydi ha'm sol massivke ko'rsetkishni qaytaradi';

3) void Utkazish(Komanda & komanda1, Komanda & komanda2) - komanda2 strukturasi'ndagi' maydanlardi' komanda1 strukturasi'na wo'tkeredi. Bul funksiya Kesteni_Ta'rtiplew() funksiyasi'nan massivdegi yeki strukturani' wo'z-ara wori'nleri'ndi' almasti'ri'w ushi'n shaqi'ri'ladi';

4) void Kesteni_Chop_YetiW(const Komanda *jm) -argumentda berilgen massivti turnir keستي qa'lipinde baspadan shi'g'aradi'.

Ush komanda haqqi'nda mag'li'wmat berilgenda programma islewinin' na'tiyjesi to'mendegishe boli'wi' mu'mkin:

FUTBOL KOMANDALARI'NIN' TURNIR KESTESI

JAMOA	O	O'	D	M	UrT	O'T	OCHKO
Quri'li'sshi'	20	15	3	2	30	10	48
Paxtakor	20	11	5	4	20	16	38
Neftchi	20	8	5	7	22	20	29

Dinamikali'q strukturalar: Berilgenler u'stinde islewde wolardi'n mug'dari' qansha boli'wi' ha'm wolarg'a yaddan qansha jay aji'ratiw kerekligi aldi'nan belgisiz boli'wi' mu'mkin. Programma islew waqti'nda berilgenler ushi'n za'ru'rat boyi'nsha yaddan jay aji'ratiw ha'm wolardi' ko'rsetkishler menen bo'lew arqali' birden-bir struktura payda yetiw procesi yadni'n' dinamikali'q bo'listiriwi deyiledi. Bul usulda payda bolg'an berilgenler kompleksine byeril-ganlarding dinamikali'q strukturasi' deyiledi, sebebi wolardi'n' wo'lshemi programma atqari'li'wi'nda wo'zgerip turadi'. Programmalisti'ri'wda dinamikali'q strukturalardan si'ziqli' dizimler(shi'nji'rlar), styoklar, gezekler ha'm binar terekler sali'sti'rg'anda ko'p isletiledi. Wolar bir-birinen elementler wortasi'ndagi' baylani'si'wlari' ha'm wolar u'stinde atqari'latug'i'n a'melleri menen pari'q yetedi.

Programma islewinde strukturag'a jan'a elementler qosi'li'wi' yamasa wo'shiriliwi mu'mkin.

Ha'r qanday berilgenlerdin' dinamikali'q strukturasi' maydanlardan tashkil tabadi' ha'm wolardi'n' ayi'ri'mlari' qon'si'las elementler menen baylani'si'w ushi'n xi'zmet yetedi.

Ma'sele. Nolden ayri'qsha pu'tin sanlardan ibarat si'zi'qli' dizim jarati'lsin ha'm wodan ko'rsetilgen sang'a ten' element wo'shirilsin.

Pu'tin sanlardi'n' si'zi'qli' dizim ko'rinishidagi dinamikali'q strukturasi' to'mendegi maydanlardan tashkil tabadi':

```
struct Shi'nji'r
{
    int element;
    Shi'nji'r * keyingi;
};
```

Programma teksti:

```
#include <iostream.h>
struct Shi'nji'r
{
    int element;
    Shi'nji'r * keyingi;
};
Shi'nji'r * Element_Jaylaw(Shi'nji'r * z, int jan'a_elem) ;
{
    Shi'nji'r * jan'a=new Shi'nji'r;
    yangi->element=yangi_elem;
    yangi->keyingi=0;
    if(z) // dizim ba'ntli
    {
        Shi'nji'r * temp=z;
        while(temp->keyingi)
            temp=temp->keyingi;//ro'yxat son'g'i' elementin tabi'w
            temp=temp->yangi;//elementni dizim aqi'ri'na qosi'w
    }
    else z=yangi; //ro'yxat bos
    return z; // dizim basi' adresin qaytari'w
```

```

}
Shi'nji'r * Element_Ushi'ri'w(Shi'nji'r * z, int del_elem)
;
{
if(z)
{
Shi'nji'r * temp=z;
Shi'nji'r * aldi'ngi=0; // usi' elementden aldi'ng'i'
// elementke ko'rsetkish
while(temp)
{
if(temp->element==del_elem)
{
if(aldi'ng'i') //o'chiriladigan element birinshi yemes
{
// wo'shiriletug'i'n elementden aldi'ng'i' elementti
// keyingi elementke jalg'aw
oldingi->keyingi = temp->keyingi;
delete temp; //elementni wo'shiriw
temp=oldingi->keyingi;
}
else
{
//o'chiriladigan element dizim basi'nda
z=z->keyingi;
delete temp;
temp=z;
}
}
else //element ma`nisi wo'shiriletug'i'n sang'a ten yemes
{
oldingi=temp;
temp=temp->keyingi;
}
}
}
return z;
}

```

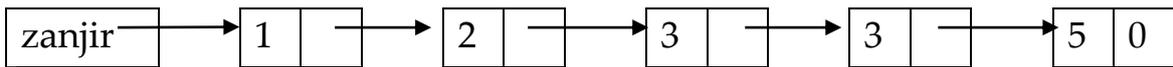
```

void Shi'nji'r_Ekrang'a(Shi'nji'r * z)
{
    cout<<"Shi'nji'r elementleri:"<<endl;
    Shi'nji'r * temp=z;
    while(temp)
    {
        cout<<temp->element<<' ';
        temp=temp->keyingi;
    }
    cout<<endl;
}
Shi'nji'r * Shi'nji'rni_Ushi'ri'w(Shi'nji'r * z) ;
{
    Shi'nji'r * temp=z;
    while(z)
    {
        z=z->keyingi;
        delete temp;
    }
    return z;
}
int main()
{
    Shi'nji'r * shi'nji'r=0;
    int san, del_element;
    do
    {
        cout<<"\n Sandi' kiritin'(0-processni tamamlaw) : ";
        cin>>san;
        if(san) shi'nji'r=Element_Jaylaw(shi'nji'r, san) ;
    }while(san) ;
    Shi'nji'r_Ekrang'a(shi'nji'r) ;
    cout<<"\n Wo'shiriletug'i'n elementti kiritin': ";
    cin>>del_element;
    zanjir=Element_Ushi'ri'w(shi'nji'r, del_element) ;
    Shi'nji'r_Ekrang'a(shi'nji'r) ;
    Shi'nji'r = Shi'nji'rni_Ushi'ri'w(shi'nji'r) ;
    return 0;
}

```

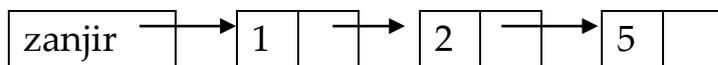
}

Programmanin' bas funksiyasi'nda si'zi'qli' dizim xosil yetiw ushi'n Shi'nji'r tu'rindegi shi'nji'r wo'zgeriwshisi ani'qlang'an boli'p, wog'an 0 ma'nisi berilgen (bos ko'rsetkish ma'nisi, woni'n' ekvivalenti - NULL). Keyin ta'kirarlaw operatori denesinde klaviaturadan pu'tin san turi'ndi'qadi' ha'm Element_Jaylaw) funksiyasi'n shaqi'ri'w arqali' bul san dizimge aqi'ri'na qosi'ladi'. Funksiya jan'a xosil bolg'an dizim basi'ni'n' adresi-ni tag'i' shi'nji'r wo'zgeriwshisine qaytaradi'. Yeger klaviaturadan 0 sani' kiritilse dizim-ni xosil kilish procesi tawsi'ladi'. Shama menen woylayi'q to'mendegi sanlar izbe-izligi kiritil-gan bo'lsin: 1, 2, 3, 3, 5, 0, Wol tu'rde xosil bolg'an dizim to'mendegi ko'riniste boladi'(10.1-su'wret) :



10.1-su'wret. Bes dana sandan sho'lkemlesken si'zi'qli' dizim

Payda bolg'an dizimdi ko'riw ushi'n Shi'nji'r_Ekrang'a() funksiyasi' shaqi'ri'ladi' ha'm ekranda dizim elementleri baspadan shi'g'ari'ladi'. Dizim u'stinde a'mel si'pati'nda berilgen san menen ustma-u'st tu'setug'i'n elementlerdi wo'shiriw ma'selesi qaralgan. Woni'n' ushi'n wo'shiriletug'i'n san del_element wo'zgeriwshige woqi'ladi' ha'm wol Element_Ushi'ri'w() funksiyasi' shaqi'rilishida argument si'pati'nda uzati'ladi'. Funksiya bul san menen ustma-u'st tu'setug'in dizim elementlerini wo'shiredi(yeger bunday element bar bolsa) ha'm wo'zgergen dizim basi'ni'n' adresini shi'nji'r wo'zgeriwshisine qaytarib beredi. Mi'sali', dizimden 3 sani' menen ustma-u'st tu'setug'in elementler wo'shirilgandan keyin wol to'mendegi ko'riniske iye boladi'(10.2-su'wret) :



10.2-su'wret. Dizimden 3 sani'n wo'shirilgandan keyingi ko'rinis

Wo'zgergen dizim elementleri ekrang'a baspadan shi'g'ari'ladi'. Programma aqi'ri'nda, Shi'nji'rni_Ushi'ri'w() funksiyasi'n shaqi'ri'w arqali' dizim ushi'n dinamikali'q tu'rde aji'rati'lg'an yad bosatiladi(garchi bul jumi'sti'n' programma tugashi waqti'nda atqari'li'wi'ni'n' ma'nisi yo'k).

Dinamikali'q strukturalarda wo'zgergiwler (dizimg'e element qosi'w yamasa wo'shiri'w) sali'sti'rg'anda kem a'mellerde atqari'li'wi', wolar jardeminde ma'selelerdi na'tiyjeli sheshiw'din' tiykarlari'nan biri yesaplanadi'.

Birlespeler ha'm wolar u'stinde a'meller: Birlespeler yadni'n' bir ken'isliginde (bir adres boyi'nsha) har-xil tu'rdegi bir qansha berilgenlerdi saqlaw imkaniyati'n beredi.

Birlespe dag'azasi' union gilt so'zi, woda'n keyin identifikator ha'm blok ishinde xar xil tu'rdegi elementler dag'azasi'nan ibarat boladi', mi'sali':

```
union Birlespe
```

```
{
```

```
    int n;
```

```
    unsigned long N;
```

```
    char Qatar[10];
```

```
};
```

Birlespenin' bul dag'azasi'nda kompilyator ta'repinen Birlespe ushi'n woni'n' ishindegi yen' ko'p jay iyelewshi elementinin' - Qatar qatari'ni'n' wo'lsheminde, yag'ni'y 10 bayt jay ajrati'ladi'. Waqti'nin' ha'r bir momentinde birlespede, dag'aza yetilgen maydanlardi'n' tek birewinin' tu'rindegi berilgen bar dep yesaplanadi'. Joqari'dag'i' mi'salda, Birlespe u'stinde a'mel atqari'li'wi'nda woni'n' ushi'n aji'rati'lg'an yadta yamasa int tu'rindegi n yamasa unsigned long tu'rindegi N yamasa Qatar ma'nisi jaylasqan dep yesaplanadi'.

Birlespe maydanlari'na tap struktura maydanlari'na murajat yetkenidey '.' arqali' murajat yetiledi.

Strukturalardan ayri'qsha tu'rde birlespe dag'azasi'nda tek woni'n' birinshi elementine baslang'i'sh ma'nis beriw mu'mkin:

```
union Birlespe
```

```
{
```

```
    int n;
```

```

    unsigned long N;
    char Qatar[10];
} birlespe=[25];

```

Bul mi'salda birlespesinin' n maydani' baslang'i'sh ma'nis alg'an yesaplanadi'.

Birlespe elementi si'pati'nda strukturalar keliwi mu'mkin ha'm wolar a'detde pu'tin berilgendi «bo'leklerge» boli'w yamasa «bo'leklerden» pu'tin berilgendi payda yetiw ushi'n xi'zmet qiladi. Mi'sal ushi'n so'zdi baytlarg'a, baytlardi tetradalarg'a (4 bitge) aji'rati'w (birlestiriw) mu'mkin.

To'mende baytti u'lken ha'm kishi yari'm baytlarg'a aji'rati'wda birlespe ha'm strukturadan paydalani'lg'an programmani teksti keltirilgen.

```

#include <iostream>
using namespace std;
union BCD
{
    unsigned char bayt;
    struct
    {
        unsigned char lo:4;
        unsigned char hi:4;
    } bin;
} bcd;
int main()
{
    bcd.bayt=127;
    cout<<"\n U'lken yari'm bayt : "<<(int) bcd.bin.hi;
    cout<<"\n Kishi yari'm bayt : "<<(int) bcd.bin.lo;
    return 0;
}

```

Programma bas funksiyasi'nda bcd birlespesinin' bayt wo'lsheminde bayt maydani'na 127 ma'nisi beriledi ha'm woni'n' u'lken ha'm kishi yari'm baytlari baspadan shi'g'ari'ladi'.

Programma islewi na'tiyjesinde ekrang'a to'mendegi na'tiyjeler shi'g'adi':

U'lken yari'm bayt: 7

Kishi yari'm bayt: 15

Ma'sele. Haqi'yqi'y tu'rdegi sanni'n' kompyuter yadi'ndag'i' ishki ko'rinishin baspadan shi'g'ari'w. Haqi'yqi'y san float tu'rinde dep yesaplanadi' ha'm wol yadta 4 bayt jay iyeleydi (1-qosi'mshag'a qaran').

Qoyilg'an ma'seleni sheshiw ushi'n birlespe qasiyetlerinen paydalani'ladi', yag'ni'y yadni'n' bir adresine haqi'yqi'y san ha'm belgiler massivi jaylasti'ri'ladi'. Haqi'yqi'y san yadqa o'qilip, belgiler massivinin' ha'r bir elementi - baytlardin' yekilik ko'rinishleri baspadan shi'g'ari'ladi'.

Programma teksti:

```
#include <iostream>
using namespace std;
const unsigned char bitlar_sani=7;
const unsigned char format=sizeof(float) ;
void Belgi_2kodi(unsigned char blg[]) ;
union San_ham_Belgi
{
    float san;
    unsigned char belgi[format];
};
int main()
{
    San_ham_Belgi san_ham_belgi;
    cin>>san_ham_belgi.san;
    for(int b=format-1;b>=0;b--)
        Belgi_2kodi(san_ham_belgi.belgi);
    return 0;
}
void Belgi_2kodi(unsigned char blg[])
{
    unsigned char Bitler_sani=128;
    for(int i=0; i<Bitler_sani; i++)
    {
        if(blg[i] & 10000000)
            cout<<'1';
        else
```

```

        cout<<'0';
        blg[i]=blg[i]<<1;
    }
    cout<<' ';
}

```

Programmada Son_ha'm_Belgi birlespesin ja'riyalaw orkali float tu'rindegi x wo'zge-ruvchisi ha'm 4 dana(float tu'ri formati'ni'n' baytlardag'i' uzi'nli'g'i') belgidan ibarat belgi massivin yadni'n' bir jayi'na jaylasi'wi'na yeriwiladi. Bas funksiyada birlespe tu'rindegi, son_ha'm_belgi Wo'zgeriwshisi dag'aza yetiledi ha'm woni'n' x maydani'na klavia-turadan haqi'yqi'y san woqi'ladi'. Keyin belgiler massivindegi ha'r bir elementinin' yekilik kodi' baspadan shi'g'ari'ladi'. Yekilik kodni' baspadan shi'g'ari'w 8 ret baytti 7-razryadi'ndag'i' sandi' baspadan shi'g'ari'w ha'm bayt razryadlari'n birge shepke ji'lji'ti'w arqali' a'melge asi'ri'ladi'. Sog'an itibar beriw kerekki, belgiler massivindegi elementlerding yekilik kodlari'n baspadan shi'g'ari'w won' ta'repten shep ta'repke wori'nlang'an. Bug'an sebep, san ishki formati'ndag'i' baytlardin' yadta «kichik bayt-kishi adresda» qag'i'ydasi'na ko're jaylasi'wi' boli'p tabi'ladi'.

Programmaga-8.5 sani' kiritilse, ekranda

```
11000001 00001000 00000000 00000000
```

ko'rinishidegi yekilik sanlari' izbe-izligi payda boladi'.

Paydalani'wshi' ta'repinen ani'qlang'an berilgenler tu'ri: C++ tilinde paydalani'wshi' ta'repinen tekg'ana struktura yamasa birlespe tu'rleri, ba'lki usi' waqi'tta bar (ani'qlang'an) tu'rler tiykari'nda jan'a tu'rlerdi jarati'li'wmasi' mu'mkin.

Paydalani'wshi' ta'repinen ani'qlanatug'i'n tu'r typedef gilt so'zi menen baslanadi', wodan keyin bar tu'r ko'rsetiledi ha'm identifikator jazi'ladi'. Aqi'ri'nda jazi'lg'an identifikator - jan'a jarati'lg'an tu'rdin' ati' yesaplanadi'. Mi'sali',

```
typedef unsigned char byte;
```

an'latpasi' byte dep atali'wshi jan'a tu'rdi jaratadi' ha'm wo'z mazmuni'na ko're unsigned char tu'ri menen ekvivalent boladi'. Keyinshelik, programmada yaddan bir bayt jay iyeleytug'i'n ha'm [0..255] aralig'indag'i' ma'nislerdi' qabi'l yetetug'i'n byte tu'rindegi wo'zgeriwshi (turaqli'lardi') ja'riyalaw mu'mkin:

```
byte c=65
byte Byte=0xFF;
```

Massiv ko'rinishidagi paydalani'wshi' ta'repinen ani'qlani'wshi' tu'r dag'azasi' to'mendegishe boladi':

```
typedef char At[30];
At atlar;
```

At tu'rindegi atlar wo'zgeriwshisi dag'azasi' - bul 30 belgidan ibarat massiv (qatar) dag'azasi' boli'p tabi'ladi'.

A'detde sheshilipati'rg'an ma'selenin' predmet oblasti' terminlerinde islew ushi'n strukturalar qayta ataladi'. Na'tiyjede quramali' du'zi'li'wg'a iye bolg'an ha'm za'ru'r ayri'qshali'qlardi' wo'zine jiynag'an jan'a tu'rlerdi jarati'li'wma muwapi'q, bo'linedi.

Mi'sali', komplyeks san haqqi'ndag'a mag'li'wmatlardi' wo'z ishine ali'wshi' Complex tu'ri to'mendegishe ani'qlanadi':

```
typedef struct
{
    double re;
    double im;
} Complex;
```

Yendi kompleks san dag'azasi'n

```
Complex KSan;
```

jazi'w mu'mkin ha'm woni'n' maydanlari'na murajat yetiw mu'mkin:

```
KSAN.re=5.64;
KSAN.im=2.3;
```

Soraw ha'm tapsi'rmalar

1. C++ tilinde strukturalarg'a ko'rsetkish.
2. Strukturag'a ko'rsetkish wo'zgeriwshisi qanday dag'aza yetiledi?
3. Strukturali' ko'rsetkishlerge mi'sal keltiring.
4. Yadni'n' dinamikali'q bo'listiriwi.
5. C++ tilinde dinamikali'q struktura.
6. C++ tilinde birlespeler.
7. Birlespeler qanday e'lin yetiledi?
8. Birlespe maydanlari'na murajat qanday a'melge asi'ri'ladi'?
9. Paydalani'wshi' ta'repinen ani'qlanatug'i'n tu'rler.
10. Birlespeler dag'azasi'na tiyisli mi'sallar keltiring.

Unions

A *union* is comprised of two or more variables that share the same memory location. Thus, a union provides a way of interpreting the same bit pattern in two or more different ways. A union declaration is similar to that of a structure, as shown in this example:

```
union utype {
    short int i;
    char ch;
} ;
```

To access a union element, use the same syntax that you would use for structures: the dot and arrow operators. If you are operating on the union directly (or through a reference), use the dot operator. If the union variable is accessed through a pointer, use the arrow operator. For example, to assign the letter 'A' to element **ch** of **u_var**, you would write the following:

```
u_var.ch = 'A';
```

Another use for a union is shown in the following program, which combines unions with bit-fields to display, in binary, the ASCII code generated when you press a key. This program also shows an alternative method for displaying the individual bits that make up a byte. The union allows the value of the key to be assigned to a character variable, while the bit-field is used to display the individual bits.

Dynamically allocated storage is obtained at run time. Thus, dynamic allocation makes it possible for your program to create variables during its execution, and it can create as many or as few variables as required, depending upon the situation. This makes dynamic allocation especially valuable for data structures such as linked lists and binary trees, which change size as they are used. Dynamic allocation for one purpose or another is an important part of nearly all real-world programs.

Memory to satisfy a dynamic allocation request is taken from the heap. As you might guess, it is possible, under fairly extreme cases, for free memory to become exhausted. Therefore, while dynamic allocation offers greater flexibility, it, too, is finite.

C++ contains two operators, **new** and **delete**, that perform the functions of allocating and freeing memory. Their general forms are shown here:

24-Lekciya: Makroslar

Joba:

1. Makroslardi' ani'qlaw
2. Makroslardi' jaylasti'ri'w

Tayansh so'zler: *Makroslardi' ani'qlaw, makroslardi' jaylasti'ri'w, psevdofunksiyalar, define direktivasi, preprocessorlar, undef direktivasi*

Makroslardi' ani'qlaw: *Makros* -bul programma (kod) bo'legi boli'p, ko'rinisi ha'm islewi tap funksiya si'yaqli'. Biraq wol funksiya yemes. Funksiyalar ha'm makroslar wortasi'nda bir neshe parqlar bar:

- programma tekstinde dus kelgen makros an'latpasi' wo'z ani'qlani'wi'(denesi menen) menen protsessor islew waqti'nda, yag'ni'y programma kompilyatsiyasidan aldi'n almasti'rildi. Usi'ni'n' sebepinen makros funksiyanı shaqi'ri'w menen baylani'sli' qosi'msha waqi't sarfini talap yetpeydi;

- makroslardan paydalani'w programmanin' baslang'i'sh kodi' (tekstin) ken'eyip ketiwine ali'p keledi. Bug'an kerı halda funksiya kodi' bir nusqada boladi' ha'm wol programma kodi'n qi'sqari'wi'na ali'p keledi. Yekinshi ta'repden funksiyanı shaqi'ri'w ushi'n qosi'msha waqi't resursları sarflanadi;

- kompilyator makrosdag'i' tu'rler sa'ykesligin teksermeydi. Usi'ni'n' sebepinen, makrosqa argument ji'beriwde tu'rlerdin' sa'ykesligi yamasa argumentler sani'ni'n' tiwri keliwi yamasa kelmewi haqqi'ndag'i' qa'telik xabarları berilmeydi;

- makros baslang'i'sh kodqa (tekstke) programma bo'legini qoyi'w qurali' bolg'anli'g'i' ushi'n ha'm bunday bo'lekler teksttin' ha'r qi'yli' jayları'na qoyi'w mu'mkinligi sebepli makroslar menen baylani'sli' bir adresler bolmaydi'. Usi'ni'n' sebepinen makroslarda ko'rsetkishler ja'riyalaw yamasa makros adreslerin isletiw mu'mkinshiligi joq.

Makroslardi' ani'qlaw ushi'n `#define` direktivasinan paydalani'ladi'. Funksiyag'a qusap makroslar da parametrlerge i'yelewi mu'mkin. Mi'sal

ushin yeki sandi' ko'beymesin yesaplaytug'in makros tomendegishe ani'qlanadi':

```
#include <iostream>
using namespace std;
#define KOBAYME(x,y)((x)+(y))
int main()
{
    int a=2, b=3,c;
    c=KOBAYME(a,b);
    cout<<c;
    return 0;
}
```

Mi'saldan ko'rinib turi'пти, wolda, si'rtqi' ko'rinishi boyi'nsha makroslardan paydalani'w funksiyalardan paydalani'wg'a uqsaw. Sol sebepli wolardi' ayi'ri'm jag'daylarda wolarg'a psevdofunksiyalar dep ataydi'. Makroslar ani'qlani'wi'ni'n' tag'i' bir ayriqsha ta'repi sonda, C++ tilinde wolardi'n' atlari'n u'lken ha'ripler menen jazi'wg'a kelisilgen.

Joqari'dag'i' mi'saldi'n' ayriqsha ko'rinishlerinen biri bul makros parametrlarin qawi's ishinde jazi'li'wi' boli'p tabi'ladi'. Bolmasa makros ani'qlani'wi'n (denesin) tekstke qosi'wda mazmunan qa'telik ju'zege keliwi mu'mkin.

Mi'sali',

```
#define KVADRAT(x) x*x
```

Programma tekstinde usi' makros isletilingen qatar bar bo'lsin:

```
int y=KVADRAT(2) ;
```

Wol halda, makros ani'qlani'wi'n tekstke qoyi'w na'tiyjesinde programma tekstinde joqari'dag'i' qatar to'mendegi ko'rinishke keledi:

```
int y=2*2;
```

Lekin, programmada makrosdi' isletiw:

```
int y = KVADRAT(x+1) ;
```

ko'rinishinde bulsa, makros ani'qlani'wi'n tekstke paydalaniw na'tiyjesinde usi' qatar

```
int y=x+1*x+1;
```

ko'rsetpesi menen almasti'rili'p, bul a'l'bette ku'tilgen almasti'ri'w yemes. Usi'ni'n' sebepinen, makros ani'qlani'wi'nda uli'wma qag'i'yda si'pati'nda parametrlardi qawi'sqa ali'w usi'ni's yetiledi:

```
#define KVADRAT(x)(x) *(x)
```

Yeger makros shaqi'riliwinda tu'rge keltiriw operatoridan paydalang'an jag'day bolsa, makros denesin toli'qli'g'i'nsha qawi'sqa ali'w talap yetiledi. Mi'sal ushi'n programma tekstinde makrosqa murajat to'mendegishe bo'lsin:

```
double x=(double) KVADRAT(x+1) ;
```

Bul halda makros ani'qlani'wi' :

```
#define KVADRAT(x)((x) *(x) )
```

ko'rinishi tuwri' yesaplanadi'.

Makros ani'qlani'wi'nda son'g'i' yesletpe si'pati'nda soni' atap ko'rsetiw kerekki, arti'qsha probeller makrosdan paydalani'wda aljasi'qlarg'a ali'p keliwi mu'mkin. Mi'sali',

```
#define BASPADAN_SHIGARIW(x) cout<<x
```

makros ani'qlani'wi'nda makros ati' BASPADAN_SHIGARIW ha'm parametrlar dizimi (x) wortasi'nda arti'qsha probel qoyi'lg'an. Preprotessor bul makrosni parametrsiz makros dep qabi'l yetedi, ha'mde (x) cout<<x qatar asti'n makros denesi dep yesaplaydi' ha'm makros almasti'ri'wlarda sol qatardi' programma tekstine qo'yi'ladi'. Na'tiyjede kompilyatsiya qa'tesi ju'z beredi. Qa'teni won'law ushi'n makros ati' ha'm parametrlar dizimi wortasi'ndag'i' probelni ali'p taslaw jeterli:

```
#define BASPADAN_SHIGARIW(x) cout<<x
```

Yeger makros ani'qlani'wi' bir qatarg'a si'y masa, sol qatar aqi'ri'na '\\' belgisin qoyi'w arqali' keyingi qatarda dawam yettiriw mu'mkin:

```
#define MUYESH3(a,b,c)(unsigned int) a+(unsigned int) b\
```

```
>(unsigned int) c &&(unsigned int) a+(unsigned int) c>\
    (unsigned int) b &&(unsigned int) b+(unsigned int) c>\
    (unsigned int) a
```

Makros ani'qlani'wi'nda basqa makroslar qatnasi'wi' da mu'mkin. To'mendegi mi'salda ishpe-ish jaylasqan makros ani'qlani'wi' ko'rsetilgen.

```
#define PI 3.14159
#define KVADRAT(x)((x) *(x) )
#define SHENBER_MAYDANI(r)(PI*KVADRAT(r) )
```

Paydalani'wg'a za'ru'rshiligi qolmagan makrosdi' programma tekstinin' qa'legen jayi'nda #undef direktivasi menen beka'r yetiw mu'mkin, yag'ni'y sol qatardan keyin makros preprotssessor ushi'n ani'q emes yesaplanadi'. To'mende shen'ber maydanin yesaplaytug'i'n programma teksti keltirilgen.

```
#include <iostream>
using namespace std;
#define PI 3.14159
#define KVADRAT(x)((x) *(x) )
#define SHENBER_MAYDANI(r)(PI*KVADRAT(r) )
int main()
{
    double r1=5, r2=10;
    double c1, c2;
    c1=SHENBER_MAYDANI(r1) ;
    #undef SHENBER_MAYDANI
    c2=SHENBER_MAYDANI(r2) ;
    cout<<c1<<endl;
    cout<<c2<<endl;
    return 0;
}
```

Programma kompilyatsiyasiNda c1=SHENBER_MAYDANI(r1) qatar normal qayta islangen halda c2=SHENBER_MAYDANI(r2) qatari' ushi'n SHENBER_MAYDANI funksiyasi' ani'qlanbag'anli'g'i' haqqi'nda qa'telik xabari' baspadan shi'g'ari'ladi'.

Makroslardi' jaylasti'ri'w: Makroslar isletiliwi mu'mkin bolg'an yeki a'mel bar: # - qatardi' jaylasti'ri'w ha'm ## - qatardi' jalg'aw a'melleri.

Yeger makros parametri' aldi'nda # - qatardi' jaylasti'ri'w a'meli qoyi'lg'an bolsa, makros ani'qlani'wi'n tekstke qoyi'w waqti'nda sol o'ring'a saykes argumenttin' (wo'zgeriwshinin') ati' qo'yi'ladi'. Buni' to'mendegi mi'salda ko'riw mu'mkin:

```
#include <iostream>
using namespace std;
#define WOZG_ATI(wozg) cout<<#wozg<<'='<<wozg<<endl;
int main()
{
    int x=10, t=15;
    WOZG_ATI(x) ;
    WOZG_ATI(t) ;
    return 0;
}
```

Programma islewi na'tiyjesinde ekranda

```
x=10
y=15
```

qatari' payda boladi'.

Qatar jalg'aw a'meli yeki qatardi' birlestiriw ushi'n xi'zmet yetedi. Qatarlardi' birlestiriwden aldi'n wolardi' aji'rati'p turg'an probeller wo'shiriledi. Yeger payda bolg'an qatar ati'ndag'i' makros bar bolsa, preprotssessor sol makros ani'qlani'wi'n (denesin) shaqi'ri'w bo'lgan jayg'a jaylasti'radi'.

Mi'sal ushi'n,

```
#include <iostream>
using namespace std;
#define MACRO_BIR cout<<"MACRO_1";
#define MACRO_EKI cout<<"MACRO_2";
#define MACRO_BIRLESPE(n) MACRO_##n
int main(int argc, char* argv[])
{
    int x=10;
    MACRO_BIRLESPE(BIR) ;
}
```

```

    cin>>x;
    return 0;
}

```

programmasi preprotessor ta'repinen qayta ishlangandan keyin woni'n arali'q teksti to'mendegi ko'riniste boladi:

```

int main(int argc, char* argv[])
{
    int x=10;
    cout<<"MACRO_1";
    cin>>x;
    return 0;
}

```

Qatarlardi' jalg'aw a'melinen jan'a wo'zgeriwshilerdi payda yetiw ushi'n paydalani'w mu'mkin.

```

#define WOZG_DAGAZASI(i) int var ## i
...
WOZG_DAGAZASI(1) ;
...

```

Joqari'dag'i' mi'salda makros wo'z ani'qlani'wi' menen almasti'ri'w na'tiyjesinde WOZG_DAGAZASI qatari' worni'nda

```

int var1;

```

ko'rsetpesi payda boladi'.

Soraw ha'm tapsi'rmalar

1. Makros ne?
2. Funksiyalar menen makroslar wortasi'nda qanday parq bar?
3. Makroslarda ko'rsetkishler ja'riyalaw mu'mkinmi?
4. Makroslardi' ani'qlawda qaysi' direktivadan paydalani'ladi'?

5. *Psevdofunksiyalar ne?*
6. *Makroslar u' stinde qanday a' meller wori' nlaw mu' mkin?*
7. *Makroslardi' ja' riyalawg' a tiyisli mi' sallar keltiring.*
8. *Makros parametrleri qanday jazi' ladi'?*

#define

#define is used to define an identifier and a character sequence that will be substituted for the identifier each time it is encountered in the source file. The identifier is called a *macro name* and the replacement process is called *macro substitution*. The general form of the directive is

```
#define macro-name character-sequence
```

Notice that there is no semicolon in this statement. There can be any number of spaces between the identifier and the character sequence, but once the sequence begins, it is terminated only by a newline.

For example, if you want to use the word UP for the value 1 and the word DOWN for the value 0, you could declare these two **#defines**:

TIP: It is important to remember that C++ provides a better way of defining constants. This is to use the **const** specifier. However, many C++ programmers have migrated from C, where **#define** is commonly used for this purpose. Thus, you will likely see it frequently in C++ code, too.

Function-Like Macros

The **#define** directive has another feature: The macro name can have arguments. Each time the macro name is encountered, the arguments associated with it are replaced by the actual arguments found in the program. This creates a *function-like* macro. Here is an example:

```
// Use a function-like macro.
#include <iostream>
using namespace std;

#define MIN(a,b) ((a)<(b)) ? a : b

int main()
{
    int x, y;

    x = 10;
    y = 20;
    cout << "The minimum is " << MIN(x, y);

    return 0;
}
```

When this program is compiled, the expression defined by **MIN(a,b)** will be substituted, except that **x** and **y** will be used as the operands. That is, the **cout** statement will be substituted to look like this:

Macro substitutions are often used to define “magic numbers” that occur in a program. For example, you may have a program that defines an array and has several routines that access that array. Instead of “hard-coding” the array’s size with a constant, it is better to define a name that represents the size, and then use that name whenever the size of the array is needed. Therefore, if the size of the array changes, you have to change it in only one place in the file, and then recompile. For example:

25-Lekciya: Fayllar ha'm ag'i'mlar

Joba:

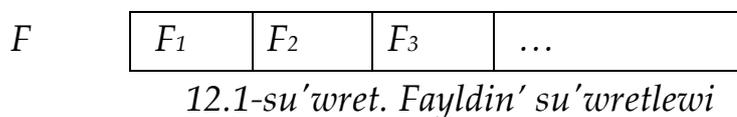
1. Fayl tu'sinigi
2. Tekst ha'm binar fayllar
3. Woqi'w-jazi'w ag'i'mlari'. Standart ag'i'mlar

Tayansh so'zler: *Fayl tu'sinigi, tekst ha'm binar fayllar, woqi'w-jazi'w ag'i'mlari'. standart ag'i'mlar, konvoyeyerli ji'beriw*

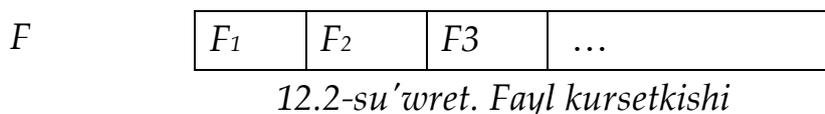
Fayl tu'sinigi: C++ tilidagi standart ha'm paydalani'wshi' ta'repinen ani'qlang'an tu'rlerdin' ayri'qshali'qi' sonnan ibarat, wolardi'n' aldi'nan berilgen shekli elementlerden ibaratli'g'i' boli'p tabi'ladi'. Ha'tte berilgenler dinamikali'q ani'qlang'anda da, operativ yadni'n' (o'rissha: куча) a'melde sheklengenligi sebepli, bul berilgenler mug'dari' joqari'dan shegaralang'an elementlerden ibarat boladi'. Ayi'ri'm bir na'tiyjeni a'melde qollani'wiw ma'seleler ushi'n aldi'nan berilgending komponentalari sani'n ani'qlaw mu'mkinshiligi joq. Wolar ma'seleni sheshiw processinde ani'qlanadi' ha'm yjeterlicha u'lken hajmda boli'wi' mu'mkin. Yekinshi ta'repden, programmada dag'aza yetilgen wo'zgeriwshilerdin' ma'nisleri' si'pati'nda ani'qlang'an berilgenler tek programma islew waqti'dagina bar boladi' ha'm programma wo'z jumi'si'n tamamlag'annan keyin jog'ali'p ketedi. Yeger programma tazadan jumi'sqa tu'sirilse, bul berilgenlerdi tazadan payda yetiw za'ru'r boladi'. Ko'pshilik na'tiyjeni a'melde qollani'wiw ma'seleler bolsa berilgenlerdi doimiy tu'rde saqlap turi'w'di' talap yetedi. Mi'sali', ka'rxana xi'zmetkerlerinin' ayli'q is haqi'si'n yesaplaytug'i'n programmada xi'zmetkerler dizimin, shtat stavkalari' ha'm xi'zmetkerler ta'repinen ali'ng'an is haqi'lar haqqi'ndag'i' mag'li'wmatlardi' doimiy tu'rde saqlap turi'w za'ru'r. Bul talaplarga fayl tu'rindegi obyektlar (wo'zgeriwshiler) juwap beredi.

Fayl - bul birdey tu'rdegi ma'nisler jaylasqan si'rtqi' yaddag'i' oblast boli'p yesaplanadi.

Fayldi', basi'nda izbe-iz tu'rde jaylasqan jazi'wlar (mi'sali', muzi'ka) menen toldi'ri'lg'an ha'm aqi'ri' bos bolg'an jeterlishe uzi'n magnit lentasi'na u'qsati'w mu'mkin.



12.1-suwretde F -fayl ati', F_1 , F_2 , F_3 - fayl elementleri (komponentalari). Tap jan'a muzi'kani' qayi's aqi'ri'na qosi'w mu'mkin bolg'anday, jan'a jazi'wlar fayl aqi'ri'na qosi'li'wi' mu'mkin. Tag'i' bir za'ru'rli tu'siniklerden biri fayl ko'rsetkishi tu'sinigi boli'p tabi'ladi'. Fayl ko'rsetkishi - usi' waqi'tta fayldan o'qilipati'rg'an yamasa wog'an jazi'li'p ati'rg'an jay (jazi'w worni'n) ko'rsetip turadi', yag'ni'y fayl ko'rsetkishi ko'rsetip turg'an wori'ndan bitga jazi'wdi' woqi'w yamasa sol jayg'a jan'a jazi'wdi' jaylasti'ri'w mu'mkin (12.2-su'wret).



Keltirilgen suwretde fayl ko'rsetkishi fayl basi'n ko'rsetip ati'r.

Fayl jazi'wlari'na murajat izbe-iz tu'rde a'melge asi'ri'ladi': p-jazi'wg'a murajat yetiw ushi'n p-1 jazi'wdi' woqi'w za'ru'r boladi'. Soni' ayti'p wo'tiw za'ru'r, fayldan jazi'wlardi' woqi'w procesi «avtamAtlasqan», wol jag'dayda i-jazi'wdi' o'qillip balg'annan keyin, ko'rsetkish na'wbettegi i+1 jazi'w basi'na ko'rsetip turadi' ha'm sol tu'rde woqi'wdi' dawam yettiriw mu'mkin (massivlerdegi dey indyeksti asi'ri'w sha'rt yemes). Fayl - bul berilgenlerdi saqlaw jayi' boli'p esaplanadi ha'm usi'ni'n' sebebini woni'n' jazi'wlari' u'stinde tuwri'dan-tuwri' a'mel atqari'p bolmaydi'. Fayl jazi'wi' u'stinde a'mel wori'nlaw ushi'n jazi'w ma'nisi operativ yadqa saykes tu'rdegi wo'zgeriwshige o'qiliwi kerek. Keyinshelik, za'ru'r a'meller sol wo'zgeriwshi u'stinde atqari'ladi' ha'm kerek bolsa na'tiyjeler tag'i' faylg'a jazi'li'wi' mu'mkin.

Operatsion sistema ko'z qarasi'nan fayl yesaplang'an ha'r qanday fayl C++ tili ushi'n materialli'q fayl yesaplanadi'. MS DOS ushi'n materialli'q fayllar <fayl ati'>.<fayl ken'eytpesi> ko'rinishidagi «8.3» formati'ndagi' qatar (at) orqali' beriledi. Fayl atlari' qatar turaqli'lar yamasa qatar wo'zgeriwshilerinde berilishi mu'mkin. MS DOS qag'i'ydalari'na ko're fayl ati' to'liq boli'wi', yag'ni'y fayl ati'ni'n'n' basi'nda adres bo'legi boli'wi' mu'mkin:

C:\USER\KURS1\Misal.cpp”, ”D:\tekst.txt”.

C++ tilinde logikali'q fayl tu'sinigi boli'p, wol fayl tu'rindegi wo'zgeriwshin an'latadi'. Fayl tu'rindegi wo'zgeriwshilerga basqa tu'rdegi wo'zgeriwshiler si'yaqli' ma'nis beriw operatori orqali' ma'nis berip bolmaydi'. Basqasha aytqanda fayl tu'rindegi wo'zgeriwshiler u'stinde hesh qanday a'mel ani'qlanbag'an. Wolar u'stinde atqari'latug'i'n barli'q a'meller funksiyalar jardeminde atqari'ladi'.

Fayllar menen islew to'mendegi basqi'shlardi' wo'z ishine aladi':

- fayl wo'zgeriwshisi a'l'bette disktag'i' fayl menen baylani'sadi';
- fayl ashi'ladi';
- fayl u'stinde jazi'w yamasa woqi'w a'melleri atqari'ladi';
- fayl jabi'ladi';
- fayl ati'ni'n' wo'zgeriw yamasa fayldi' diskdan wo'shiriw a'mellerin wori'nlawg'a boladi'.

Tekst ha'm binar fayllar: C++ tili C tilinden woqi'w-jazi'w a'melin wori'nlawshi' standart funksiyalar kitapxanasini miyrasli'q boyi'nsha alg'an. Bul funksiyalar <stdio.h> bas bet fayli'nda dag'aza yetilgen. O'qiw-jazi'w a'melleri fayllar menen atqari'ladi'. Fayl tekst yamasa binar (yekilik) boli'wi' mu'mkin.

Tekst fayl - ASCII kodi'ndagi' belgiler menen berilgenler kompleksi. Belgiler izbe-izligi qatarlarga boli'ngan boladi' ha'm qatardi'n' aqi'rg'i' belgisi si'pati'nda CR (karetkani qaytari'w yamasa '\r') LF (qatardi' wo'tkeriw yamasa '\n') belgiler jupli'g'i' yesaplanadi'. Tekst fayldan berilgenlerdi woqi'li'wi'nda bul belgiler jupli'g'i' bir belgige - CR belgisine

almasti'ri'ladi' ha'm kerisinshe, jazi'wda CR belgisi yeki CR ha'm LF belgilerine almasti'ri'ladi'. Fayl aqi'ri' #26(^z) belgisi menen belgilanadi.

Tekst faylg'a basqasha tariyp beriw da mu'mkin. Yeger fayldi' tekst redakciyasi'nda ekrang'a shi'g'ari'w ha'm woqi'w mu'mkin bolsa, bul tekst fayl. Klaviatura da kompyuterge tek tekstlerdi jiberedi. Basqasha aytqanda programma ta'repinen ekrang'a shig'ari'latug'i'n barli'q mag'li'wmatlardi' stdout ati'ndag'i' tekst fayli'na shi'g'ari'li'pati'r dep qaraw mu'mkin. Tap sonday klaviaturadan o'qilipati'rg'an ha'r qanday berilgenlerdi tekst fayli'nan o'qilipati'r dep yesaplanadi'.

Tekst fayllari'ni'n' komponentalari qatarlar dep ataladi'. Qatarlar u'zliksiz jaylasi'p, ha'r qi'yli' uzi'nli'qda ha'm bos boli'wi' mu'mkin. Shama menen woylayi'q, T tekst fayli' 4 qatardan ibarat bo'lsin:

1-atar #13#10	2-qatar uzunroq #13#10	#13#10	4-qatar #13#10#26
------------------	---------------------------	--------	----------------------

12.3-su'wret. To'rtew qatardan sho'lkemlesken tekst fayli'

Tekstti ekrang'a shi'g'ari'wda qatar aqi'ri'ndag'i' #13#10 basqari'w belgileri jupli'g'i' kursordi' keyingi qatarg'a tu'siredi ha'm woni' qatar basi'na ali'p keledi. Bul tekst fayl ekrang'a chop etilsa, woni'n' ko'rinisi to'mendegishe boladi':

```
1-qatar[13][10]
2-qatar uzunroq[13][10]
[13][10]
4-qatar[13][10]
[26]
```

Tekstdegi [n]-n-kodli' basqari'w belgisini an'latadi'. A'detde tekst tahrirlari bul belgilerdi ko'rsetpeydi.

Binar fayllar - bul a'piwayi'g'ana baytlar izbe-izligi. A'detde binar fayllardan berilgenlerdi paydalani'wshi' ta'repinen quralsiz «ko'riw» za'ru'r bolmag'an jag'daylarda isletiledi. Binar fayllardan woqi'w-jazi'wda baytlar u'stinde hesh qanday konvyertatsiya a'melleri atqari'lmaydi'.

Woqi'w-jazi'w ag'i'mlari'. Standart ag'i'mlar: Ag'i'm tu'sinigi berilgenlerdi faylg'a woqi'w-jazi'wda wolardi' belgiler izbe-izligi yamasa ag'i'mi' ko'riniside woyda sawlelendiriwden kelip shi'qqan. Ag'i'm u'stinde to'mendegi a'mellerdi wori'nlaw mu'mkin:

- ag'i'mdan berilgenler bloki'n operativ yadqa woqi'w;
- operativ yaddag'i' berilgenler bloki'n ag'i'mg'a jazi'w(shi'g'ari'w) ;
- ag'i'mdag'i' berilgenler bloki'n jan'ari'w;
- ag'i'mdan jazi'wdi' woqi'w;
- ag'i'mg'a jazi'wdi' shi'g'ari'w.

Ag'i'm menen isleytug'i'n barli'q funksiyalar buferli, formatlawg'an yamasa formatlawmag'an woqi'w-jazi'wdi' ta'miyinleydi.

Programma jumi'sqa tu'skende woqi'w-jazi'wdi'n' to'mendegi standart ag'i'mlar ashi'ladi':

- stdin - woqi'wdi'n' standart qurali';
- stdout - jazi'wdi'n' standart qurali';
- stderr - qa'telik haqqi'nda xabar beriwdin' standart qurali';
- stdprn - qag'azg'a baspadan shi'g'ari'wdi'n' standart qurali';
- stdaux - standart ja'rdemshi qurilma.

Kelisiw boyi'nsha stdin - paydalani'wshi' klaviaturasi', stdout ha'm stderr - terminal (ekran), stdprn - printer menen, ha'mde stdaux-kompyuter ja'rdemshi portlarina baylani'sqan yesaplanadi'. Berilgenlerdi woqi'w-jazi'wda stderr ha'm stdaux ag'i'mi'nan basqa ag'i'mlar buferlanadi, yag'ni'y belgiler izbe-izligi operativ yadni'n' bufer dep atali'wshi ken'isliginde waqti'nsha jamlanadi.

Ha'zirdegi operatsion sistemalarda klaviatura ha'm displeylar tekst fayllari' si'pati'nda qaraladi. Haqi'yqattan da berilgenlerdi klaviaturadan programmaga kirgiziw (woqi'w) mu'mkin, ekrang'a bolsa shi'g'ari'w (jazi'w) mu'mkin. Programma jumi'sqa tu'skende standart woqi'w ha'm jazi'w ag'i'mlari' worni'na tekst fayllardi' belgilew arqali' bul ag'i'mlardi' qayta ani'qlaw mu'mkin. Bul jag'dayni' woqi'wdi' (jazi'wdi') qayta adreslew ju'z beredi deyiledi. Woqi'w ushi'n qayta adreslewda '<' belgisinen, jazi'w ushi'n bolsa '>' belgisinen paydalani'ladi'. Mi'sal ushi'n gauss.exe atqari'li'wshi' programma berilgenlerdi woqi'wdi' klaviaturadan

yemes, ba'lki massiv.txt fayli'nan a'melge asi'ri'w za'ru'r bolsa, wol buyri'q qatari'nda to'mendegi ko'riniste ju'kleniwi za'ru'r boladi':

```
gauss.exe < massiv.txt
```

Yeger programma na'tiyjesin na'tiyje.txt fayli'na chiqaarish za'ru'r bolsa

```
gauss.yexye > na'tiyje.txt
```

qatari' jazi'ladi'.

Ha'm aqi'r-aqi'betde, yeger berilgenlerdi massiv.txt fayli'nan woqi'w ha'm na'tiyjeni na'tiyje.txt fayli'na jazi'w ushi'n

```
gauss.exe < massiv.txt > na'tiyje.txt
```

buyri'q qatari' tyeriladi.

Uluwma alg'anda, bir programmanin' shi'g'i'w ag'i'mi'n yekinshi programmanin' kiriw ag'i'mi' menen bo'lew mu'mkin. Buni' *konvyeyrli ji'beriw* deyiledi. Yeger yeki A.exe programmasi B.exe programmasiga berilgenlerdi ji'beriw kerek bolsa, wol halda wolar wortasi'na '|' belgi qoyi'p jazi'ladi':

```
A.exe | B.exe
```

Bul ko'rinistegi programmalar wortasi'ndag'i' konvyeyrli ji'beriwdi operatsion sistemanin' wo'zi ta'miyinleydi.

Soraw ha'm tapsi'rmalar

1. C++ tilinde fayl tu'sinigi.
2. C++ tilinde logikali'q fayl tu'sinigi.
3. Tekst fayllari'n ani'qlama berin'.
4. Binar fayllar.
5. Fayllar menen islew basqi'shlari'n sanap berin'.
6. Tekst fayllari'ni'n' komponentalri.
7. C++ tilinde ag'i'm tu'sinigi.

8. *Ag'i'm u'stinde atqari'latug'i'n a'meller.*
9. *Qanday standart ag'i'mlardi' bilesiz?*
10. *Konveyerli ji'beriw.*

File I/O

You can use the C++ I/O system to perform file I/O. In order to perform file I/O, you must include the header `<fstream>` in your program. It defines several important classes and values.

Opening and Closing a File

In C++, a file is opened by linking it to a stream. As you know, there are three types of streams: input, output, and input/output. To open an input stream, you must declare the stream to be of class **ifstream**. To open an output stream, it must be declared as class **ofstream**. A stream that will be performing both input and output operations must be declared as class **fstream**. For example, this fragment creates one input stream, one output stream, and one stream capable of both input and output:

26-Lekciya: Woqi'w-jazi'w funksiyalari'

Joba:

1. Belgilerdi woqi'w-jazi'w funktsiyalari'
2. Qatarlardi' woqi'w-jazi'w funktsiyalari'
3. Formatli' woqi'w ha'm jazi'w funktsiyalari'
4. Fayldan woqi'w-jazi'w funktsiyalari'
5. Fayl ko'rsetkishin basqari'w funktsiyalari'

Tayansh so'zler: *Belgilerdi woqi'w-jazi'w funksiyalari', qatarlardi' woqi'w-jazi'w funksiyalari', formatli' woqi'w ha'm jazi'w funksiyalari', komponenta, san tu'rindegi argument, shi'g'i'w formati', fayldan woqi'w-jazi'w funksiyalari', mode ma'nisi, fayl ko'rsetkishin basqari'w funksiyalari'*

Belgilerdi woqi'w-jazi'w funksiyalari': Belgilerdi woqi'w-jazi'w funksiyalari' makros ko'rinishinde a'melge asi'ri'lg'an.

`getc()` makrosi' tayi'nlang'an ag'i'mdan na'wbettegi belgini qaytaradi' ha'm kiriw ag'i'mi' ko'rsetkishin keyingi belgini woqi'wg'a maslasti'ri'lg'an halda asi'radi'. Yeger woqi'w tabi'sli' bolsa `getc()` funksiyasi' belgisiz `int` ko'rinishidagi ma'nisti, bolmasa EOF qaytaradi'. Usi' funksiya prototipi to'mendegishe:

```
int getc(FILE * stream)
EOF identifikator makrosi'
#define EOF(-1)
```

ko'rinishinde ani'qlang'an ha'm woqi'w-jazi'w a'mellerinde fayl aqi'ri'n belgilew ushi'n xi'zmet qiladi. EOF ma'nisi belgili char tu'rinde dep yesaplanadi'. Usi'ni'n' sebebini woqi'w-jazi'w processinde `unsigned char` tu'rindegi belgiler isletilse, EOF makrosi'n isletip bolmaydi'.

Na'wbettegi mi'sal `getc()` makrosi'n isletiwdi ko'rinetug'i'n yetedi.

```
#include <iostream>
using namespace std;
#include <stdio.h>
int main(int argc, char* argv[])
{
```

```

char ch;
cout<<"Belgi kiritin': ";
ch=getc(stdin) ;
cout<<"Siz '"<<ch<<"' belgisin kirgizdin'iz.\n";
return 0;
}

```

getc() makrosi' ko'pshilik jag'daylarda stdin ag'i'mi' menen isletilingenligi sebepli, woni'n' getc(stdin) ko'rinsine ekvivalent bolg'an int getchar() makrosi' ani'qlang'an. Joqari'dag'i' mi'salda ch=getc(stdin) qatari'n ch=getchar() qatari' menen almasti'ri'w mu'mkin.

Belgini ag'i'mg'a shi'g'ari'w ushi'n putc() makrosi' ani'qlang'an ha'm woni'n' prototipi

```
int putc(int c, FILE*stream)
```

ko'rinsinde ani'qlang'an. putc() funksiyasi' stream ati' menen berilgen ag'i'mg'a c belgini shi'g'aradi'. Funksiya qaytaratug'i'n ma'nisi si'pati'nda int tu'rine aylantirilgan c belgi boladi'. Yeger belgini shi'g'ari'wda qa'telik ju'z byersa EOF qaytari'ladi'.

putc() funksiyasi'n standart stdout ag'i'mi' menen baylani'sqan varianti - putc(c, strout) ushi'n putchar(c) makrosi' ani'qlang'an.

Qatarlardi' woqi'w-jazi'w funksiyalari': Ag'i'mdan qatardi' woqi'wg'a mo'lsherlengen getc() funksiyasi'ni'n' prototipi

```
char * gets(char *s);
```

ko'rinsinde ani'qlang'an. gets() funksiyasi' standart ag'i'mdan qatardi' woqi'ydi' ha'm woni' s wo'zgeriwshisine jaylasti'radi'. Jaylasti'ri'w waqti'nda ag'i'mdag'i' '\n' belgisi '\0' belgisi menen almasti'ri'ladi'. Bul funksiyani' isletiwde o'qilipati'rg'an qatardi'n' uzi'nli'g'i' 8 qatar ushi'n ajirati'lg'an jay uzi'nli'g'i'nan asi'p ketmasligini basqariw kerek boladi'. puts() funksiyasi'

```
int puts(const char *s)
```

ko'rinsinde boli'p, wol standart ag'i'mg'a argumentda ko'rsetilgen qatardi' shi'g'aradi'. Bunda qatar aqi'ri'na jan'a qatarg'a wo'tiw belgisi '\n' qosi'ladi'.

Yeger qatardi' ag'i'mg'a shi'g'ari'w tabi'sli' bolsa puts() funksiyasi' keru bolmag'an sandi', bolmasa EOF qaytaradi'.

Qatardi' woqi'w-jazi'w funksiyalari'n isletiwge mi'sal si'pati'nda to'mendegi programmani keltiriw mu'mkin.

```
#include <stdio.h>
using namespace std;
int main(int argc, char* argv[])
{
    char s[80];
    puts("Qatar kiritin': ") ;
    gets(s) ;
    puts("Siz kiritgen qatar: ") ;
    puts(s) ;
    return 0;
}
```

Formatli' woqi'w ha'm jazi'w funksiyalari': Formatli' woqi'w ha'm jazi'w funksiyalari' - scanf() ha'm printf() C tilinden miyrasli'q menen ali'ng'an. Bul funksiyalardi' isletiw ushi'n stdio.h bas bet fayli'n programmaga qosi'w kerek boladi'.

Formatli' woqi'w funksiyasi' scanf() to'mendegi prototipga iye:

```
int scanf(const char *<format>[<adres>, ...])
```

Bul funksiya standart ag'i'mdan berilgenlerdi formatli' woqi'wdi' a'melge asi'radi'. Funksiya, kiriw ag'i'mi'ndag'i' maydanlar izbe-izligi ko'rinishidagi belgilerdi birme-bir woqi'ydi' ha'm ha'r bir maydandi' <format> qatari'nda keltirilgen format parametrine saykes tu'rde formatlaydi'. Ag'i'mdag'i' ha'r bir maydang'a format parametri ha'm na'tiyje jaylasatug'i'n wo'zgeriwshinin' adresi boli'wi' sha'rt. Basqasha aytqanda, ag'i'mdag'i' maydan (aji'rati'lg'an belgiler izbe-izligi) ko'rsetilgen formatdag'i' ma'nisge sa'wlelendiriledi ha'm wo'zgeriwshi menen atalg'an yad bo'legine jaylasti'ri'ladi' (saqlanadi'). Funksiya ag'i'mdan berilgenlerdi woqi'w procesin «to'ldiriwshi belgini» ushratqanda yamasa ag'i'm tamami bolivi na'tiyjesinde toqtati'wi' mu'mkin. Ag'i'mdan berilgenlerdi woqi'w tabi'sli' bolsa, funksiya tabi'sli' aylantirilgan ha'm yadqa saqlang'an maydanlar sani'n qaytaradi'. Yeger hesh bir maydandi' saqlaw

mu'mkinshiligi bolmag'an bolsa, funksiya 0 ma'nisin qaytaradi'. Ag'i'm aqi'ri'na kelip qalg'anda (fayl yamasa qatar aqi'ri'na.) woqi'wg'a xarakat bolsa, funksiya EOF ma'nisin qaytaradi'.

Formatlaw qatari' <format> belgiler qatari' boli'p, wol ush kategoriyag'a bo'linedi:

- toldiriwshi belgiler;
- to'diriwshi belgilerden ayri'qsha belgiler;
- format parametrleri.

Toldiriwshi belgiler - bul probel, '\t', '\n' belgileri. Bul belgiler formatlaw qatari'nan woqi'ladi', lekin saqlanmaydi.

To'ldiriwshi belgilerden ayri'qsha belgiler - bul qalg'an barli'q ASCII belgileri, '%' belgisinen ti'sqari'. Bul belgiler formatlaw qatari'nan woqi'ladi', lekin saqlanmaydi.

Format parametrleri - ag'i'm maydani'ndag'i' belgilerdi ko'rip shi'g'i'w, woqi'w ha'm adresi menen berilgen wo'zgeriwshiler tu'rine saykes tu'rde almasti'ri'w procesin basqaradi. Ha'r bir format parametrine bir wo'zgeriwshi adresi saykes keliwi kerek. Yeger format parametrlar sani' wo'zgeriwshilerden ko'p bolsa, na'tiyje qanday boli'wi'n aldi'nan ayti'p bolmaydi'. Bolmasa, yag'ni'y wo'zgeriwshiler sani' ko'p bolsa, arti'qsha wo'zgeriwshiler inabatqa ali'nbaydi'.

Format parametrleri to'mendegi ko'rinishke iye:

%[<ken'lik>][F|N][h|1|L]<tu'r belgisi>

Format ani'qlawtiruvchisi '%' belgisinen baslanadi' ha'm wodan keyin 12.1-kestede keltirilgen sha'rt yamasa sha'rt bolmag'an komponentalar keledi.

12.1-jadvad. Format parametrleri ha'm wolardi'n' wazi'ypasi'

Komponenta	Boli'wi'sha'rt yamasa joq	Wazi'ypasi'
[*]	yaq	Na'wbettegi ko'rip chiqilayotgan maydan ma'nisin wo'zgeriwshige wo'zlestirmaslik belgisi. Kiriw ag'i'mi'ndag'i' maydan ko'rip shi'g'i'ladi', lekin wo'zgeriwshinde saqlanmaydi.
[<ken'lik>]	yaq	Maydan ken'ligini ani'qlawtiruvchisi. Woqi'latug'i'n

		belgilerding maksimal sani'n ani'qlaydi'. Yeger ag'i'mda toldiruvchi belgi yamasa almasti'ri'lmaytug'i'n belgi uchrasi funktsiya sali'sti'rg'anda kem sandag'i' belgilerdi woqi'wi' mu'mkin.
[F N]	yaq	Wo'zgeriwshi ko'rsetkishinin'(adresinin') modifikatori: F-far pointer; N-near pointer
[h l L]	yaq	Argument tu'rinin' modifikatori. <tur belgisi> menen ani'qlang'an wo'zgeriwshinin' qi'sqa(short-h) yamasa uzi'n(long-1, L) ko'rinishin ani'qlaydi'.
<tur belgisi>	awa	Ag'i'mdag'i' belgilerdi almasti'ri'latug'i'n tu'r belgisi

Ag'i'mdag'i' belgilerdi almasti'ri'latug'i'n tu'r belgisinin' qabi'llawi' mu'mkin bolg'an belgiler to'mendegi kestedeki keltirilgen.

12.2-keste. Almasti'ri'latug'i'n tu'r belgileri

Berilgenler tu'ri	Kutilayotgan ma'nis	Argument tu'ri
San tu'rindegi argument		
d, D	Wonli'q pu'tin	Pu'tinge ko'rsetkish (int * arg yamasa long * arg)
Ye, ye	Ju'ziwshi noqatli' san	float tu'rine ko'rsetkish (float * arg)
f	Ju'ziwshi noqatli' san	float tu'rine ko'rsetkish (float * arg)
G, g	Ju'ziwshi noqatli' san	float tu'rine ko'rsetkish (float * arg)
o	Segizlik san	Pu'tinge ko'rsetkish(int * arg)
O	Segizlik san	Pu'tinge ko'rsetkish(long * arg)
i	Wonli'q, segizlik ha'm won alti'li'q pu'tin san	Pu'tinge ko'rsetkish(int * arg)
I	Wonli'q, segizlik ha'm won alti'li'q pu'tin san	Pu'tinge ko'rsetkish(long * arg)
wol	Belgisiz wonli'q san	unsigned int tu'rine ko'rsetkish (unsigned int * arg)
Wol	Belgisiz wonli'q san	unsigned long tu'rine ko'rsetkish (unsigned long * arg)
x	Won alti'li'q san	Pu'tinge ko'rsetkish(int * arg)
X	Won alti'li'q san	Pu'tinge ko'rsetkish(int * arg)
Belgiler		
s	Qatar	Belgiler massivke ko'rsetkish(char * arg)
c	Belgi	Belgine ko'rsetkish(char * arg). Belgi ushi'n maydan ken'ligi beriliwi mu'mkin(mi'sali', %4). N belgiden sho'lkemlesken belgiler massivine

		ko'rsetkish(char arg[N]).
%	'%' belgisi	Hesh qanday almasti'ri'wlar baja-rilmaydi, '%' belgisi saqlanadi'.
Ko'rsetkishler		
n	int ko'rsetkish(int * arg)	%n argumentigacha muvaffa-qiyatli o'qilgan belgiler sani' nag'i'z wo'zi int ko'rsetkishi boyi'nsha adresde saqlanadi'.
p	YYYY:ZZZZ yamasa ZZZZ ko'rinishidagi won alti'li'q	Obyektga ko'rsetkish(far*yamasa near*)

Formatli' jazi'w funksiyasi' printf() to'mendegi prototipga iye:

```
int printf(const char * <format>[, <argument>, ...])
```

Bul funksiya standart ag'i'mg'a formatlap shi'g'ari'wdi a'melge asi'radi'. Funksiya argumentler izbe-izligidegi ha'r bir argument ma'nisin qabi'l yetedi ha'm wog'an <format> qatari'ndagi' saykes format ani'qlawshisin qollaydi' ha'm ag'i'mg'a shi'g'aradi'. Ha'r bir format ani'qlawshisina bir wo'zgeriwshi adresi saykes keliwi kerek. Yeager format ani'qlawshilari sani' wo'zgeriwshilerden ko'p bolsa, na'tiyjede ne boli'wi'n aldi'nan ayti'p bolmaydi'. Bolmasa, yag'ni'y wo'zgeriwshiler sani' ko'p bolsa, arti'qsha wo'zgeriwshiler inabatqa ali'nbaydi'. Yeager ag'i'mg'a shi'g'ari'w tabi'sli' bolsa, funksiya shi'g'ari'lg'an baytlar sani'n qaytaradi', bolmasa EOF.

printf() funksiyasi'ni'n' <format> qatari' argumentlerdi almasti'ri'w, formatlaw ha'm shi'g'ari'w procesin basqaradi ha'm wol yeki tu'rdegi obyektlerden tashkil tabadi':

- ag'i'mg'a wo'zgeriwsiz shig'ari'latug'i'n a'piwayi' belgiler;
- argumentler dizimidegi tanlanatug'i'n argumentge qollani'latug'i'n

format parametrleri.

Format parametri to'mendegi ko'rinishke iye:

```
%[<bayraq>][<.ken'lik>][<.xona>][F|N|h|1|L]<tu'r belgisi>
```

Format parametri '%' belgisinen baslanadi' ha'm wodan keyin 12.3-kestede keltirilgen sha'rt yamasa sha'rt bolmag'an komponentalar keledi.

12.3-keste. Format ani'qlawtiruvchilari ha'm wolardi'n' wazi'ypasi'

Komponenta	Boli'wi' sha'rt yamasa joq	Wazi'ypasi'
[bayraq]	yaq	Bayraq belgileri. Shi'g'ari'li'p ati'rg'an ma'nis-ni shepke yamasa wog'an tyekislashni, sanni'n' belgisin, wonli'q bo'lshek noqati'n, oxirdagi nollardi, segizlik ha'm won alti'li'q sanlardi'n' belgilerdi baspadan shi'g'ari'wdi' basqaradi. Ma'-salan, '-' bayrag'ina ma'nisti aji'rati'lg'an wori'ng'a sali'sti'rg'anda shep ta'repten baslap chiqa-rishni ha'm kerek bolsa won' ta'repten probel menen tolti'ri'wdi' an'latadi', bolmasa shep tap-mondan probeller menen toldi'radi' ha'm dawami'na ma'nis shig'ari'ladi'.
[<ken'lik>]	yaq	Maydan ken'ligini ani'qlawtiruvchisi. Shig'ari'latug'i'n belgilerding minimal sani'n ani'qlaydi'. Za'ru'r bolsa ma'nis jazi'li'wi'nan artqan jaylar probel menen toldi'ri'ladi'.
[.<xana>]	yaq	Ani'qli'q. Shig'ari'latug'i'n belgilerding mak-simal sani'n ko'rsetedi. Sanlar ushi'n-chi-qariladigan cifrlardi'n' minimal sani'n.
[F N h l L]	yaq	Wo'lshem modifikatori. Argumenttin' qi'sqa(short-h) yamasa uzi'n(long-1, L) ko'rini-shini, adres tu'rin ani'qlaydi'.
<tu'r belgisi>	awa	Argument ma'nisi almasti'ri'latug'i'n tu'r belgisi

Almasti'ri'latug'i'n tu'r belgisinin' qabi'llawi' mu'mkin bolg'an belgiler to'mendegi kestedek keltirilgen.

12.4-keste. *print()* funksiyasi'ni'n' almasti'ri'latug'i'n tu'r belgileri

Berilganlar tu'ri	Kutilayotgan ma'nis	Shi'g'i'w formati'
San ma'nisleri'		
d	Pu'tin san	Belgili wonli'q pu'tin san
i	Pu'tin san	Belgili wonli'q pu'tin san
o	Pu'tin san	Belgisiz segizlik pu'tin san
wol	Pu'tin san	Belgisiz wonli'q pu'tin san
x	Pu'tin san	Belgisiz won alti'li'q pu'tin san (a, b, c, d, e, f belgileri isletiledi)
X	Pu'tin san	Belgisiz won alti'li'q pu'tin san (A, B, C, D, E, F belgileri isletiledi)

f	Ju'ziwshi noqatli' san	[-]dddd.dddd ko'rinishidagi suzuvchi noqatli' san
e	Ju'ziwshi noqatli' san	[-]d.dddd yamasa e[+/-]ddd ko'rinishidagi suzuvchi noqatli' san
g	Ju'ziwshi noqatli' san	Ko'rsetilgan aniqliqqa saykes ye yamasa f formasi'ndag'i' ju'ziwshi noqatli' san
E, G	Ju'ziwshi noqatli' san	Ko'rsetilgan aniqliqqa saykes ye yamasa f formasi'ndag'i' ju'ziwshi noqatli' san. ye format ushi'n 'Ye' belgisi chikariladi.
Belgiler		
s	Qatarg'a ko'rsetkish	o-belgisi uchramaguncha yamasa ko'rsa-tilgen aniqliqqa yeriwilmaguncha bel-gilar ag'i'mg'a shig'ari'ladi'.
c	Belgi	Bir belgi shig'ari'ladi'
%	Hesh narse	'%' belgisi ag'i'mg'a shig'ari'ladi'
Ko'rsetkishler		
n	int ko'rsetkish(int * arg)	%n argumentigacha tabi'sli' shi'g'ari'lg'an belgiler sani', a'yne int ko'rsetkishi boyi'nsha adresda saqlanadi'
p	Ko'rsetkish	Argumentni YYYY:ZZZZ yamasa ZZZZ ko'rinishidagi won alti'li'q sang'a aylanti'ri'p ag'i'mg'a shi'g'aradi'.

Berilganlar ma'nisleri'n ag'i'mdan woqi'w ha'm ag'i'mg'a shi'g'ari'wda scanf() ha'm printf() funksiyalari'nan paydalani'wg'a mi'sal:

```
#include <stdio.h>
using namespace std;
int main(int argc, char* argv[])
{
int a, natiyje;
float b;
char belgi, qatar[81];
printf("\nPu'tin ha'm ju'ziwshi noqatli' sanlardi', ") ;
printf("\nbelgi ha'mde qatardi' kiriting\n") ;
natiyje=scanf("%d %f %c %s", &a, &b, &belgi, qatar) ;
printf("\nAg'i'mdan %d ta ma'nis o'qildi ", natiyje) ;
printf("ha'm wolar to'mendegiler:") ;
printf("\n %d %f %c %s \n", a, b, belgi, qatar) ;
return 0;
}
```

Programma paydalaniwshidan pu'tin ha'm ju'ziwshi noqatli' sanlardi', belgi ha'm qatardi' kirgiziwdi so'raydi. Bug'an juwapan paydalani'wshi ta'repinen

10 12.35 A Qatar

ma'nisleri' kiritilse, ekrang'a

Ag'i'mdan 4 dana ma'nis o'qildi ha'm wolar to'mendegiler:

10 12.35 A Qatar

qatarlari' baspadan shi'g'ari'ladi'.

Fayldan woqi'w-jazi'w funksiyalari': Fayl ag'i'mi' menen woqi'w-jazi'w a'melin wori'nlaw ushi'n fayl ag'i'mi'n ashi'w za'ru'r. Bul jumi'sti' a'melge asi'ri'wshi funksiya prototipi tomendegishe:

FILE * fopen(const char *filename, const char *mode);

ko'rinishinde ani'qlang'an fopen() funksiyasi' arqali' a'melge asi'ri'ladi'.

Funksiya filename ati' menen fayldi' ashadi', wol menen ag'i'mdi' baylani'sti'radi' ha'm ag'i'mdi' identifikasiya yetiwshi ko'rsetkishti juwap si'pati'nda qaytaradi'. Fayldi' ashi'wdin' ilaji bolmag'anli'g'i'n fopen() funksiyasinin' NULL ma'nisli' juwapi' an'latadi'.

Parametrlar dizimidegi yekinshi, mode parametri fayldi' ashi'w rejimin ani'qlaydi'. Wol qabi'llawi' mu'mkin bolg'an ma'nisler 12.5-kestede keltirilgen.

12.5-keste. Fayl ashi'w rejimleri

mode ma`nisi	Fayl ashi'li'w varianti suwretlewi
R	Fayl tek woqi'w ushi'n ashi'ladi'.
W	Fayl jazi'w ushi'n ashi'ladi'. Yeger bunday fayl bar bolsa, wol qaytaldan jazi'ladi' (jan'alanadi')
A	Faylg'a jazi'wdi' qosi'w rejimi. Yeger fayl bar bolsa, fayl woni'n' aqi'ri'na jazi'wdi' jazi'w ushi'n ashi'ladi', bolmasa jan'a fayl jarati'ladi' ha'm jazi'w rejiminde ashi'ladi'.
r+	A'meldegi fayl wo'zgertiw (woqi'w ha'm jazi'w) ushi'n ashi'ladi'.
w+	Jan'a fayl jarati'li'p, wo'zgertiw (woqi'w ha'm jazi'w) ushi'n ashi'ladi'. Yeger fayl bar bolsa, wodag'i' aldi'ng'i' jazi'wlar wo'shiriledi ha'm wol qayta jazi'wg'a tayarlanadi'.
a+	Faylg'a jazi'wdi' qosi'w rejimi. Yeger fayl a'melde bolsa, woni'n' aqi'ri'na (EOF belgiinen keyin) jazi'wdi' jazi'w (woqi'w) ushi'n ashi'ladi', bolmasa jan'a fayl jarati'ladi' ha'm jazi'w rejiminde ashi'ladi'.

Tekst fayli' ashilipati'rg'anli'g'i' haqqi'nda xabar beriw ushi'n fayl ashi'li'w rejimi qatari'na 't' belgisin qosi'p jazi'w za'ru'r boladi'. Mi'sali', tekst fayl wo'zgertiw (woqi'w ha'm jazi'w) ushi'n ashilg'anli'g'i'n xabar beriw ushi'n "rt+" qatari' jazi'w kerek boladi'. Tap sonday binar fayllar u'stinde islew ushi'n 'b' belgisin isletiw kerek. Mi'sal ushi'n fayl ashi'li'wi'ni'n' "wb+" rejimi binar fayl jan'alani'wi'n an'latadi'.

Fayldi wo'zgertiw (woqi'w-jazi'w) ushi'n ashilg'anda, berilgenlerdi ag'i'mdan woqi'w, ha'mde ag'i'mg'a jazi'w mu'mkin. Biraq jazi'w a'melinen keyin tezlik penen woqi'p bolmaydi', woni'n' ushi'n woqi'w a'melinen aldi'n fseek() yamasa rewind() funksiyalari' shaqi'ri'li'wi' sha'rt.

Shama menen woylayi'q "C:\USER\STUDENT\Nodira.txt" atli' tekst fayldi' woqi'w ushi'n ashi'w za'ru'r bo'lsin. Bul talap

```
FILE *f=fopen("C:\USER\STUDENT\Nodira.txt","r+");
```

an'latpasi'n jazi'w arqali' a'melge asi'raladi. Na'tiyjede diskta bar bolg'an fayl programmada f wo'zgeriwshisi ati' menen qandayda bir narse dep tu'siniledi. Basqasha aytqanda, programmada keyinirek u'stinde wori'nlang'an barli'q a'meller, disktag'i' "Nodira.txt" fayli' u'stinde ju'z beredi.

Fayl ag'i'mi' menen islew tawsi'lg'annan keyin wol a'l'bette jabi'li'wi' kerek. Woni'n' ushi'n fclose() funksiyasi'nan paydalani'ladi'. Funksiya prototipi to'mendegi ko'rinishke iye:

```
int fclose(FILE * stream);
```

fclose() funksiyasi' ag'i'm menen bag'li'q buferlerdi tazalaydi (mi'sali', faylg'a jazi'w ko'rsetpesi berilgen, lekin sol waqtqasha buferde saqlani'p ati'rg'an berilgenlerdi disktag'i' faylg'a jazadi') ha'm fayldi' jabadi'. Yeger fayldi' jabi'w qa'telikke ali'p kelsa, funksiya EOF kiymatini, normal jag'dayda 0 ma'nisin qaytaradi'.

fgetc() funksiyasi' prototipi

```
int fgetc(FILE *stream) ;
```

ko'rinishinde ani'qlang'an boli'p, fayl ag'i'mi'nan belgi woqi'w'di' a'melge asi'radi'. Yeger woqi'w a'wmetli bolsa, funksiya o'qilgan belgi int tu'rindagi belgisiz pu'tin sang'a aylanti'radi'. Yeger fayl aqi'ri'n woqi'wg'a

ha'reket yetilse yamasa qa'telik ju'z bersa, funksiya EOF ma'nisin qaytaradi'.

Ko'rinib turi'пти', wolda, `getc()` ha'm `fgetc()` funksiyalari' derlik birdey jumi'sti' atqaradi', ayi'rmashi'li'g'i' sonda, `getc()` funksiyasi' belgi standart ag'i'mdan woqi'ydi'. Basqasha aytqanda, `getc()` funksiyasi', fayl ag'i'mi' si'pati'nda standart operator bolg'an haldag'i' `fgetc()` funksiyasi' menen ani'qlang'an makros boli'p tabi'ladi'.

`fputc()` funksiyasi'

```
int fputc(int c, FILE *stream) ;
```

prototipi menen ani'qlang'an. `fgetc()` funksiyasi' fayl ag'i'mi'na argumentde ko'rsetilgen belgi jazadi' (shi'g'aradi') ha'm wol a'mel yetiwinde `putc()` funksiyasi' menen birdey.

Fayl ag'i'mi'nan qatar woqi'w ushi'n

```
char *fgetc(char *s, int n, FILE *stream)
```

prototipi menen `fgetc()` ani'qlang'an. `fgetc()` funksiyasi' fayl ag'i'mi'nan belgiler izbe-izligin `s` qatari'na woqi'ydi'. Funksiya woqi'w procesin ag'i'mdan `n-1` belgi o'qillip balg'annan keyin yamasa keyingi qatarg'a wo'tiw belgisi '\n' dus kelgende toqtatadi'. Son'g'i' halda '\n' belgisi da `s` qatarg'a qosi'ladi'. Belgilerdi woqi'w tawsi'lg'annan keyin `s` qatar aqi'ri'na, qatar toqtatiw belgisi '\0' belgisi qosi'ladi'. Yeger qatardi' woqi'w tabi'sli' bolsa, funksiya `s` argument ko'rsetetug'i'n qatardi' qaytaradi', bolmasa `NULL`.

Fayl ag'i'mi'na qatardi' `fputs()` funksiyasi' ja'rdeminde shi'g'ari'w mu'mkin. Bul funksiya prototipi

```
int fputs(const char *s, FILE *stream) ;
```

ko'rinishinde ani'qlang'an. Qatar aqi'ri'ndag'i' jan'a qatarg'a wo'tiw belgisi ha'm teminatorlar ag'i'mg'a shi'g'ari'lmaydi'. Ag'i'mg'a shi'g'ari'w tabi'sli' bolsa, funksiya teris san qaytaradi', bolmasa `EOF`.

`feof()` funksiyasi' negizi makros boli'p, fayl u'stinde woqi'w-jazi'w a'melleri orinlani'pati'rg'an fayl aqi'ri' belgisi dus kelgen yamasa joq yekenligin an'latadi'. Funksiya

```
int feof(FILE *stream) ;
```

prototipiga iye boli'p wol fayl aqi'ri' belgisi uchrasa, nolden ayri'qsha sandi' qaytaradi', basqa jag'daylarda 0 ma'nisin qaytaradi'.

To'mende keltirilgen mi'salda faylg'a jazi'w ha'm woqi'wg'a a'meller ko'rsetilgen.

```
#include <stdio.h>
#include <iostream>
using namespace std;
int main(int argc, char* argv[])
{
    char c;
    FILE *in, *out;
    if((in=fopen("D:\\USER\\NODIRA.TXT", "rt") )==NULL)
    {
        cout<<"NODIRA.txt fayli'n ashi'lmadi!!\n";
        return 1;
    }
    if((out=fopen("D:\\USER\\NODIRA.DBL", "wt+") ) ==NULL)
    {
        cout<<"Student.db1 fayli'n ashi'lmadi!!\n";
        return 1;
    }
    while(!feof(in) )
    {
        char c=fgetc(in) ;
        cout<<c;
        fputc(c, out) ;
    }
    fclose(in) ;
    fclose(out) ;
    return 0;
}
```

Programmada “NODIRA.TXT” fayli' tekst fayl si'pati'nda ashi'lg'anli'g'i' ushi'n wol in wo'zgeriwshisi menen baylani'sqan. Tap sonday, “NODIRA.DBL” tekst fayli' jazi'w ushi'n ashi'lg'anli'g'i' wol out menen baylani'sqan. Yeger fayllardi' ashi'w a'wmetsiz bolsa, og'an saykes xabar beriledi ha'm programma wo'z jumi'si'n toqtatadi'. Keyinirek, in

fayli' aqi'ri'na jetpegenshe, wodan belgiler woqi'ladi' ha'm ekrang'a, ha'mde out fayli'na shig'ari'ladi'. Programma aqi'ri'nda yeki fayl da jabi'ladi'.

Ma'sele. Yelekli ta'rtiplew usi'li'. Berilgen x vektori'n ko'bikshe usi'li'nda kemeymey **tug'i'n** yetip ta'rtiplew to'mendegishe a'melge asi'ri'ladi': massivtin' qon'si'las elementleri x_k ha'm x_{k+1} ($k=1,2,3,\dots,n-1$) sali'wti'ri'ladi'. Yeger $x_k > x_{k+1}$ bolsa, wol halda bul elementler wo'z-ara wori'n almasadi'. Sol jol menen birinshi wo'tiwde yen' u'lken element vektordi'n' aqi'ri'na jaylasadi'. Keyingi qa'demde vektor basi'nan $n-1$ wori'ndagi' elementne shekem joqari'da belgilengen jol menen qalg'an elementlerdin' yen' u'lkeni $n-1$ wori'ng'a jaylasti'ri'ladi' ha'm t.b..

Yelekli ta'rtiplew usi'li' ko'biksheli ta'rtiplew usi'li'na uqsaw, lekin x_k ha'm x_{k+1} ($k=1,2,3,\dots,n-1$) elementler wori'n almasg'annan keyin «yelek»ten wo'tkeriw a'meli qollani'ladi': shep ta'repdegi kishi element mu'mkinshiligi bari'nsha shep ta'repke ta'rtiplew saqlang'an halda ko'shiriledi. Bul usi'l a'piwayi' ko'biksheli ta'rtiplew usi'li'na sali'sti'rg'anda tez isleydi.

Programma teksti:

```
#include <stdio.h>
#include <malloc.h>
#include <iostream>
int * Kobikchali_Tartiplew(int*, int) ;
using namespace std;
int main(int argc, char* argv[])
{
    char fati[80];
    printf("Fayl ati'ni'n' kiritin':") ;
    scanf("%s", &fati) ;
    int Wolshem, i=0;
    int * Massiv;
    FILE * f1, *f2;
    if((f1=fopen(fati, "rt") ) ==NULL)
    {
        printf("Qa'te:%s fayli' ashi'lmadi!", fati) ;
        return 1;
    }
    fscanf(f1, "%d", &Wolshem) ;
    Massiv=(int*) malloc(Wolshem*sizeof(int) ) ;
```

```

while(!feof(f1) )
{
    fscanf(f1, "%d", Massiv[i++]) ;
    cout<<Massiv[i-1]<<endl;
}
fclose(f1) ;
Massiv=Kobikchali_Tartiplew(Massiv, Wolshem) ;
f2=fopen("natiyje.txt", "wt+") ;
fprintf(f2, "%d%c", Wolshem, ' ' ) ;
for(i=0; i<Wolshem; i++)
    fprintf(f2, "%d%c", Massiv[i], ' ' ) ;
fclose(f2) ;
return 0;
}
int * Kobikchali_Tartiplew(int M[], int n)
{
    int almasdi=1, waqtinsha;
    for(int i=0; i<n-1 && almasdi;i++)
    {
        almasdi=0;
        for(int j=0; j<n-i-1;j++)
            if(M[j]>M[j+1])
            {
                almasdi=1;
                waqtinsha=M[j];
                M[j]=M[j+1];
                M[j+1]=waqtinsha;
                int k=j;
                if(k)
                    while(k && M[k]>M[k-1])
                    {
                        waqtinsha=M[k-1];
                        M[k-1]=M[k];
                        M[k]=waqtinsha;
                        k--;
                    }
            }
    }
}

```

```
    return M;
}
```

Programmada berilgenlerdi ag'i'mdan woqi'w yamasa ag'i'mg'a shi'g'ari'wda fayldan formatli' woqi'w-fscanf() ha'm jazi'w-fprintf() funksiyalari'nan paydalani'lg'an. Bul funksiyalardi'n' saykes tu'rde scanf() ha'm printf() funksiyalari'nan ayi'rmashi'li'g'i'-wolar berilgenlerdi birinshi argument si'pati'nda beriletug'i'n tekst fayldan woqi'ydi' ha'm jazadi',

Ati' paydalani'wshi' ta'repinen kirgizetug'i'n f1 fayldan pu'tin sanlar massivini' uzi'nli'g'i' ha'm ma'nisleri' woqi'ladi' ha'm wol ta'rtilenip f2 faylg'a jazi'ladi'.

Vektordi' ta'rtilenw Ko'bikchali_Ta'rtilenw() funksiyasi' ta'repinen a'melge asi'ri'ladi'. Wog'an vektor ha'm woni'n' uzi'nli'g'i' kiretug'i'n parametr boladi' ha'm ta'rtilengen vektor funksiya na'tiyjesi si'pati'nda qaytari'ladi'.

Na'wbettegi yeki funksiya fayl ag'i'mi'nan formatlawmag'an woqi'w-jazi'wdi' a'melge asi'ri'wg'a mo'lsherlengen.

fread() funksiyasi' to'mendegi prototipga iye:

```
size_t fread(void*ptr, size_t size,
             size_t n, FILE *stream) ;
```

Bul funksiya ag'i'mdan ptr ko'rsetip turg'an buferge, ha'r biri size bayt bolg'an n ta berilgenler bloki'n woqi'ydi'. Woqi'w tabi'sli' bolsa, funksiya o'qilgan bloklar sani'n qaytaradi'. Yeger woqi'w processinde fayl aqi'ri' ushi'rasi'p qolsa yamasa qa'telik ju'z bersa, funksiya toli'q o'qilgan bloklar sani'n yamasa 0 qaytaradi'.

fwrite() funksiyasi' prototipi

```
size_t fwrite(const void*ptr, size_t size, size_t n,
              FILE *stream) ;
```

ko'rinishinde ani'qlang'an. Bul funksiya ptr ko'rsetip turg'an buferden, ha'r biri size bayt bolg'an n ta berilgenler bloki'n ag'i'mg'a shi'g'aradi'. Jazi'w tabi'sli' bolsa, funksiya jazi'lg'an bloklar sani'n qaytaradi'. Yeger jazi'w processinde qa'telik ju'z bersa, funksiya toli'q jazi'lg'an bloklar sani'n yamasa 0 qaytaradi'.

Fayl ko'rsetkishin basqari'w funksiyalari: Fayl ashi'lg'anda, wol menen `stdio.h` bas bet fayli'nda ani'qlang'an FILE strukturasi' baylani'sadi'. Bul struktura ha'r bir ashi'lg'an fayl ushi'n usi' jazi'w worni'n ko'rsetiwshi yesaplawshisin fayl ko'rsetkishin saykes qo'yadi. A'detde fayl ashi'lg'anda ko'rsetkish ma'nisi 0 boladi'. Fayl u'stinde wori'nlang'an ha'r bir a'melden keyin ko'rsetkish ma'nisi o'qilgan yamasa jazi'lg'an baytlar sani'na asadi. Fayl ko'rsetkishin basqari'w funksiyalari `fseek()`, `ftell()` ha'm `rewind()` funksiyalari' fayl ko'rsetkishin wo'zgertiw, ma'nisin ali'w imkaniyati'n beredi.

`ftell()` funksiyasi'ni'n' prototipi

```
long int ftell(FILE *stream) ;
```

ko'riniside ani'qlang'an boli'p, argumentda ko'rsetilgen fayl menen baylani'sqan fayl ko'rsetkishi ma'nisin qaytaradi'. Yeger qa'telik ju'z bersa funksiya `-1L` ma'nisin qaytaradi'.

```
int fseek(FILE *stream, long offset, int from) ;
```

prototipiga iye bolg'an `fseek()` funksiyasi' stream fayli' ko'rsetkishin from jayi'na sali'sti'rg'anda offset bayt arali'qqa ji'lji'ti'wdi' a'melge asi'radi'. Tekst rejimidegi ag'i'mlar ushi'n offset ma'nisi 0 yamasa `ftell()` funksiyasi' qaytargan ma'nis boli'wi' kerek. from parametri to'mendegi ma'nislerdi' qabi'llawi' mu'mkin:

```
SEEK_SET(=0)-fayl basi';
```

```
SEEK_CUR(=1)-fayl ko'rsetkishinin' usi' waqi'ttag'i'  
ma'nisi;
```

```
SEEK_END(=2) -fayl aqi'ri'.
```

Funksiya fayl ko'rsetkishi ma'nisin wo'zgertiw tabi'sli' bolsa, 0 ma'nisin, bolmasa nolden ayri'qsha ma'nis qaytaradi'.

`rewind()` funksiyasi'

```
void rewind(FILE *stream) ;
```

prototipi menen ani'qlang'an boli'p, fayl ko'rsetkishin fayl baslani'wi'na ali'p keledi.

To'mende keltirilgen programmada binar fayl menen islew ko'rsatilgan.

```
#include <iostream.h>
```

```
#include <stdio.h>
```

```

#include <string.h>
struct Shaxs
{
    char Familiya[20];
    char Ati[15];
};
int main()
{
    int n, k;
    cout<<"Studentler sani'n kiritin': ";
    cin>>n;
    FILE *oqim1, *oqim2;
    Shaxs *shaxs1, *shaxs2, shaxsk;
    Shaxs1=new Shaxs[n];
    Shaxs2=new Shaxs[n];
    if((ag'i'm1=fopen("Student.dat", "wb+"))==NULL)
    {
        cout<< " Student.dat ashi'lmadi!!! " ;
        return 1 ;
    }
    for(int i=0; i<n; i++)
    {
        cout<<i+1<< " -shaxs mag'li'wmatlari'n kiritin':\n " ;
        cout<< " Familyasi: " ;gets(shaxs1[i].Fami'li'ya) ;
        cout<< " Ati': " ; gets(shaxs1[i].I'si'm) ;
    }
    if(n==fwrite(shaxs1, sizeof(Shaxs), n, ag'i'm1))
        cout<<"Berilgenlerdi jazi'w a'melge asi'ri'ldi'!\n ";
    else
    {
        cout<<"Berilgenlerdi jazi'w a'melge asi'rilmadi!\n";
        return 3;
    }
    cout<< " Fayl uzi'nli'g'i': " <<ftell(ag'i'm1) << '\n';
    fclose(ag'i'm1) ;
    if((ag'i'm2=fopen( " Student.dat ", " rb+ " ) ) ==NULL)
    {
        cout<< " Student.dat woqi'wg'a ashi'lmadi!!! " ;
    }
}

```

```

return 2;
}
if(n==fread(shaxs2, sizeof(Shaxs), n, ag'i'm2) )
for(int i=0; i<n;i++)
{
cout<<i+1<< " -shaxs mag'li'wmatlari':\n " ; cout<< "
Familyasi: " <<shaxs2[i].Fami'li'ya<<'\\n';
cout<< " Ati': " <<shaxs2[i].Ism<<'\\n';
cout<<"*****\\n";}
else
{
cout<< " Fayldan woqi'w a'melge asi'rilmadi! \\n " ;
return 4 ;
}
do
{
cout<< " Fayl jazi'wi' nomerini kiritin'(1.. " <<n<< " ) :";
cin>>k;
} while(k<0 && k>n) ;
k--;
cout<< " Aldi'ng'i' Fami'li'ya: "
<<shaxs2[k].Fami'li'ya<<'\\n';
cout<< " Jan'a Fami'li'ya: " ;
gets(shaxs2[k].Fami'li'ya) ;
if(fseek(ag'i'm2, k*sizeof(Shaxs), SEEK_SET) )
{
cout<< " Faylda " <<k+1;
cout<< " -jazi'wg'a wo'tiwde qa'telik ju'z berdi???\n " ;
return 5;
}
fwrite(shaxs2+k, sizeof(Shaxs), 1, ag'i'm2) ;
fseek(ag'i'm2, k*sizeof(Shaxs), SEEK_SET) ;
fread(&shaxsk, sizeof(Shaxs), 1, ag'i'm2) ;
cout<<k+1<< " -shaxs mag'li'wmatlari':\n " ;
cout<< " Familyasi: " <<shaxsk.Fami'li'ya<<'\\n';
cout<< " Ati': " <<shaxsk.Ism<<'\\n';
fclose(ag'i'm2) ;
delete shaxs1;

```

```

delete shaxs2;
return 0;
}

```

Joqari'da keltirilgen programmada, aldi'n "Student.dat" fayli' binar fayl si'pati'nda jazi'w ushi'n ashi'ladi' ha'm wol ag'i'm1 wo'zgeriwshisi menen baylani'sadi'. Shaxs haqqi'ndag'i' mag'li'wmatti' yadi'nda saqlawshi' n wo'lshefli dinamikali'q shaxs1 strukturalar massivi ag'i'm1 fayli'na jazi'ladi', fayl uzi'nli'g'i' chop yetilip fayl jabi'ladi'. Keyin, tap sol fayl ag'i'm2 ati' menen woqi'w ushi'n ashi'ladi' ha'm wodag'i' berilgenler shaxs2 strukturalar massivine woqi'ladi' ha'm ekrang'a baspadan shi'g'ari'ladi'. Programmada fayldag'i' jazi'w di' wo'zgertiw (qayta jazi'w) a'melge asi'ri'lg'an. Wo'zgertiw yetiliwi kerek bolg'an jazi'w ta'rtip nomeri paydalani'wshi' ta'repinen kiritiledi (k wo'zgeriwshisi) ha'm shaxs2 strukturalar massivindegi saykes wori'ndag'i' strukturani'n' Fami'li'ya maydani' klaviaturadan kiritilgen jan'a qatar menen wo'zgartiriledi. ag'i'm2 fayl ko'rsetkishi fayl basi'nan $k * \text{sizeof}(\text{Shaxs})$ baytga ji'lji'ti'ladi' ha'm shaxs2 massivtin' k-strukturasi' (shaxs2+k) sol wori'ndan baslap faylg'a jazi'ladi'. Keyin ag'i'm2 fayli' ko'rsetkishi wo'zgertiw kiritilgen jazi'w basi'na qaytari'ladi' ha'm bul jazi'w shaxsk strukturasi'na woqi'ladi' ha'm de ekrang'a baspadan shi'g'ari'ladi'.

Ma'sele. Haqi'yqi'y sanlar jazi'lg'an f fayli' berilgen. f fayldag'i' elementlerdin' worta arifmetigidan kishi bolg'an elementler mug'dari'n ani'qlansin.

Ma'seleni sheshiw ushi'n f fayli'n jarati'w ha'm qaytaldan woni' woqi'w ushi'n ashi'w za'ru'r boladi'. Jarati'lg'an fayldi'n' barli'q elementlerinin' ji'yi'ndi'si' s wo'zgeriwshisinde payda yetiledi ha'm wol fayl elementleri sani'na bo'linedi. Keyin f fayl ko'rsetkishi fayl basi'na ali'p kelinedi ha'm elementler qayta woqi'ladi' ha'm s ma'nisinen kishi elementler sani'-k sanap boriladi.

Fayldi' jarati'w ha'm wodag'i' worta arifmetikdan kishi sanlar mug'dari'n ani'qlawdi' bo'lek funksiya ko'rinishinde ani'qlaw mu'mkin.

Programma teksti:

```

#include <iostream.h>
#include <stdio.h>
# include <string.h>
int Fayl_Jarati'w()
{
    FILE * f;
    double x;
    // f fayldi' jan'adan payda etiw ushi'n ashi'ladi'
    if((f=fopen( "Sanlar.dbl", "wb+"))==NULL) return 0;
    char *qatar=new char[10];
    int n=1;
    do
    {
        cout<<"Sandi' kiritin'(bos qatar tamamlaw):" ;
        gets(qatar) ;
        if(strlen(qatar) )
        {
            x=atof(qatar) ;
            fwrite(&x, sizeof(double), n, f) ;
        }
    } while(strlen(qatar) ) ;// qatar bos bo'lmasa, ta'kirarlaw
    fclose(f) ;
    return 1;
}
int OAdan_Kishiler_Sani'()
{
    FILE * f; double x;
    f=fopen("Sanlar.dbl ", "rb+");
    double s=0; // s-f fayl elementleri ji'yi'ndi'si'
    while(!feof(f) )
    {
        if(fread(&x, sizeof(double), 1, f) ) s+=x;
    }
    long sanlar_miqdori=ftell(f) /sizeof(double) ;
    s/=sonlar_miqdori; // s-worta arifmetikali'q
    cout<< " Fayldag'i' sanlar worta arifmetiki= " <<s<<'endl';
    fseek(f, SEEK_SET, 0) ; // fayl basi'na kelinsin
    int k=0;
}

```

```

while(fread(&x, sizeof(x), 1, f) )
{
k+=(x<s) ; //o'rta arifmetikdan kishi elementler sani'
}
fclose(f) ;
return k;
}
int main()
{
if(Fayl_Jarati'w() )
{
cout<<"Sanlar.dbl fayli'ndag'i' \n " ;
int OA_kichik=OAdan_Kishiler_Sani'() ;
cout<<"Worta arifmetikdan kishi sanlar mug'dari'= " ;
cout<<OA_kichik;
}
else // f fayli'n jarati'w muvafaqiyatsiz boldi'.
cout<< " Sanlar.dbl fayli'n ashi'w mu'mkinshiligi
bo'lmadi!!!"; return 0;
}

```

Programmada bas funksiyadan ti'sqari' yeki funksiya ani'qlang'an: int Fayl_Jarati'w() - diskta "Sanlar.dbl" atli' fayldi' jaratadi'. Yeger fayldi' jarati'w tabi'sli' bolsa, funksiya 1 ma'nisin, bolmasa 0 ma'nisin qaytaradi'. Fayldi' jarati'wda klaviaturadan sanlardi'n' qatar ko'rinisi woqi'ladi' ha'm sang'a aylantirilib, faylg'a jazi'ladi'. Yeger bos qatar kiritilse, sanlardi' kirgiziw procesi to'xtiladi ha'm fayl jabi'ladi';

int OAdan_Kishiler_Sani'() - disktag'i' "Sanlar.dbl" atli' fayldi' woqi'w ushi'n ashi'ladi' ha'm fayl elementlerinin' s worta arifmetigidan kishi elementleri sani' k tabi'ladi' ha'm funksiya na'tiyjesi si'pati'nda qaytari'ladi'.

Bas funksiyada fayldi' jarati'w tabi'sli' keshgenligi tekseriledi ha'm sog'an saykes xabar beriledi

Soraw ha'm tapsi'rmalar

1. Belgilerdi woqi'w-jazi'w funksiyalari' qanday a'melge asi'ri'ladi'?
2. `getc()` makrosi'ni'n' wazi'ypasi'.
3. Belgi ag'i'mg'a shaqi'ri'wda qnday makroslardan paydalani'ladi'?
4. Qatarlardi' woqi'w-jazi'w funksiyalari'na mi'sallar keltiring.
5. Qaysi' funksiyalar formatli' woqi'w-jazi'w funksiyalari' yesaplanadi'?
6. Formatlaw qatari' kategoriyalari'.
7. Format ani'qlawtiruvchilari ne?
8. `print()` funksiyasi' haqqi'nda mag'li'wmat berin'.
9. Fayl ag'i'mi'n ashi'w qanday a'melge asi'ri'ladi'?
10. Fayldi' ashi'w muvafaqqiyatsiz keshgenda qanday wo'zgeris ju'zege keledi?
11. Fayl ko'rsetkishin basqari'w funksiyalari'na mi'sal keltiring.

Formatted text files (like those used in the preceding examples) are useful for a variety of situations, but they do not have the flexibility of unformatted binary files. For this reason, C++ supports a number of binary (sometimes called “raw”) file I/O functions that can perform unformatted operations.

When performing binary operations on a file, be sure to open it using the **ios::binary** mode specifier. Although the unformatted file functions will work on files opened for text mode, some character translations may occur. Character translations negate the purpose of binary file operations.

In general, there are two ways to write and read unformatted binary data to or from a file. First, you can write a byte by using the member function **put()**, and read a byte by using the member function **get()**. The second way uses C++’s block I/O functions: **read()** and **write()**. Each way is examined here.

Using get() and put()

The **get()** function has many forms, but the most commonly used version is shown next, along with that of **put()**:

```
istream &get(char &ch);  
  
ostream &put(char ch);
```

The **get()** function reads a single character from the associated stream and puts that value in *ch*. It returns a reference to the stream. This value will be null if the end of the file is reached. The **put()** function writes *ch* to the stream and returns a reference to the stream.

The following program will display the contents of any file on the screen. It uses the **get()** function.

There is actually a more compact way to code the loop that reads and displays a file, as shown here:

```
while(in.get(ch))  
    cout << ch;
```

This form works because **get()** returns the stream **in**, and **in** will be false when the end of the file is encountered.

You can detect when the end of the file is reached by using the member function **eof()**, which has this prototype:

```
bool eof( );
```

It returns true when the end of the file has been reached; otherwise, it returns false.

In addition to the form shown earlier, the **get()** function is overloaded several different ways. The prototypes for the three most commonly used overloaded forms are shown here:

```
istream &get(char *buf, streamsize num);
```

```
istream &get(char *buf, streamsize num, char delim );
```

```
int get( );
```

The first form reads characters into the array pointed to by *buf* until either *num*-1 characters have been read, a newline is found, or the end of the file has been encountered. The array pointed to by *buf* will be null-terminated by **get()**. If the newline character is encountered in the input stream, it is *not* extracted. Instead, it remains in the stream until the next input operation.

The second form reads characters into the array pointed to by *buf* until either *num*-1 characters have been read, the character specified by *delim* has been found, or the end of the file has been encountered. The array pointed to by *buf* will be null-terminated by **get()**. If the delimiter character is encountered in the input stream, it is *not* extracted. Instead, it remains in the stream until the next input operation.

27-Leksiya. Obyektke bag'darlang'an programmalastırıw texnologiyaları. İnkapsulyatsiya, Meraslıq, Polimorfizm.

Joba:

1. Ob'ektke bag'ıtlang'an programmalastırıw
2. OBP nın' payda bolıwı ha'm printsipleri
3. Ob'ektke mo'lsherlengen yondoshuvning ush printspi'

Ob'ektke bag'ıtlang'an programmalastırıw: Esaplaw texnologiyasının' rawajlanıwı ha'm sheshilip atırg'an ma'selelerdin' quramalasıp barıwı programmalastırıwdın' turli modellerin' (paradigmalarınin') juzege keliwine sebep bolıp atır. Ob'ektke bag'ıtlang'an programmalastırıw (OBP) bul programmalastırıwg'a jan'adan kirisiu bolıp , protseduralı programmalastırıwda bar bolg'an ma'selelerdi joq etiw mahsetinde juzege kelgen. Birinshiler katarında islep shıg'ılğan kompilyatorlarda (ma'selen , FORTRAN tili ushın) programmanın' funktsiyalarınan paydalanıwga tiykarlangan protsedura modelin qollap quwatlagan. Bul model ja'rdeminde programma jaratıwshı (programmist) bir neshe mın' qatarlı programmalarđ duziwi mumkin edi. Rawajlanı wdın' keyingi basqıshında progmmalardın' sistemalı modeli payda boldı ha'm ol ALGOL , Paskal ha'm C tilleri kompilyatorlarında o'z ko'rinishin taptı. Sistemalı programmalastırıwdın' mag'anası programmanı o'z-ara baylanıstırg'an protseduralar (bloklar) ha'm olar qayta isleytug'in berilgenlerdin' kompleksi dep qarawdan ibarat. Usı model programma blokları ken' qollawg'a goto operatorınan imkan barınsha kem paydalanıwg'a tiykarlang'an ha'm bunda programma duziwshi (programmist) on mın'nan artıq qatarlarg'a iye monolit programmanı jarata alg'an. Duzilgen programmalarđ sazlaw ha'm baqlaw protseduralı modelge salıstırg'anda an'sat bolg'an.

Qollanba C++ tilinde OBP a'melge asırıwda belgili arnawlı at klass tu'sinigi klasstın' ag'zaları ha'm olarg'a mu'ra'ja't turleri, klasstın' dosları, klasslar miyraslıg'ı poliformizmdi a'melge asırıw, statik ha'm dinamik poliformizmler , operatorđ qayta juklew , ayrıqsha

jag'dayda qayta islew shablonlar ha'm standart shablonlar bibliotekası temalarınin' lektsiyası ha'm a'meliy ma'seleleri qaraladı.

OBP nın' payda bolıwı ha'm printsipleri: Struktalı programmaların tiykarında jaratılǵan programmanın' barlıǵının' orınlanıwı a'piwayı ha'm sızıqlı bolıp esaplanadı. A tochkaların programma yadqa yuklenedi ha'm orınlawdı baslaydı. B tochkada jumısın toqtatıp yadı bosatadı. Programma orınlanıw barısında basqa na'rselerden, ma'selen, berilgenlerdi saqlawshı fayldan, videokartalardan paydalanıwı mumkin biraq qayta islew din' tiykarǵı bo'legi bir jerde orınlanadı. A'dette berilgenlerdi qayta islew quramalı emes bolıp ha'r qıylı matematikalıq ha'm logikalıq qurallardan paydalanıladı.

OBP da bunday sızıqlı jag'day kem ushıraydı. Usınday na'tiyjege erisilgendede protsess ko'binese ulıwma basqasha ko'riniste boladı.

OBP da tiykarǵı itibar berilgenlerdin' du'zilisi ha'm mazmunına ha'mde berilgenlerdin' bir biri menen o'z ara ta'sirine qaratıladı.

Yag'nıy buni proektlewde u'lken dıqqat qaratıladı. Sonın' menen birge buni rauajlandırıp barıw imkaniyatı da boladı. Konkret berilgenler tu'rleri haqqında pikirlewde anıq bir qararg'a kelgennen son' usı prezentatsiyanı programmanın' keyingi versiyalarında yamasa ulıwma basqa programma du'zgende paydalanıw imkaniyatı bar boladı. Bul turdegi qolaylıqlar programmalar jaratıwda og'ada a'piwayılastırıp imkaniyatın beredi.

Demek texnologiyanın' atınan belgili bulardın' barlıǵı belgili a'melge asırıladı. Obiektler bul OBP nın' qosımshalarınin' qurılıs blogı bolıp esaplanadı. Ob'ektti o'z ishine ag'za retinde o'zgeriwshiler ha'm funktsiyalardı alg'an struktura dep qaraw mu'mkin. Ob'ektlerden o'zgeriwshiler ma'nisler ob'ektke saqlanıwshı berilgenlerdi , al funktsiyalar bolsa ob'ektin' a'meller imkaniyatına qatnas jasawdı tmiynleydiyu Ob'ekt tu'ri u'shın OBP da arnawlı at klass tusinigi kiritilgen. Klastı anıqlaw ob'ektlerdi yag'nıy klass nusqaların jaratıw imkaniyatın beredi.

Ob'ekt penen qa'siyet ha'm maydan tusinikleri baylanıslı bolıp olar ob'ekttegi berilgenlerge qatnastı ta'miynleydi. Qa'siyet ha'm maydanlardın' ma'nisleri ob'ektin' jag'dayların anıqlaydı. Qa'siyet ha'm maydanlar

turlerge bo'lip u'si sebepli berilgenler string, int ha'm basqada turlerde saqlanıwı mu'mkin.

Qa'siyettin' maydandan parqı ol berilgenlerge tuwrıdan tuwrı qatnas jasaw imkanın bermeydi. Ma'selen adamnın' jası int turinde bolg'an bolsa , onın' ma'nisleri int turi shegarasında bolıwı mu'mkin. Jastın' minus ma'nisin kiritiwde bul qadag'alaw dı qa'siyet o'z wazıypası etip aladı. Qa'siyetten paydalanıw arqalı jas ma'nisleri diapazonın 1 den 120 g'a shekem belgilew mumkin. Metod termini menen ob'ekt ta'repinen usınıs etiliwshi funktsiya tusiniledi. Metodlar ob'ektin' funktsional imkaniyatlarına qatnas jasawda qollanıladı.

Qa'siyet ha'm maydanlar sıyaqlı metodlar za'rur bolganda sırtqı kodlardan mura'jatlarg'a sheklewler qoyıw ushın ashıq yamasa jabıq bolıwı mumkin. OBP tomendegi tiykargı printsiplerge tayanadı.

1. A'mel orinlaytugınlardıń barlıg'ı ob'ekt esaplanadı;

2. Esaplaw ob'ektler ortasında o'z ara ta'sir (berilgenlerdi ob'ekt almastırıw) jolı menen a'melge asırıladi ha'm onda bir ekinshisi ta'repinen qandayda bir jumıstı orinlawdı talap etedi. Ob'ektler xabarlar arqalı bir- biri menen ta'sirde boladı. Xabarlar - bul za'rur bolıwı mumkin bolg'an argumentler menen tolıqtırılǵ'an qanday da bir jumıstı orinlawg'a ruxsat alıw.

3. Ha'r bir ob'ekt basqa ob'ektlerden ibarat g'a'rezsiz yadqa iye boladı.

4. Ha'r bir ob'ekt, onın' ulıwmalıq qa'siyetlerin a'mellerin klasstın' wa'kili esaplanadı.

5. Klasta ob'ektin' ha'reketin yag'nıy funktsional imkaniyatı beriledi. Usının' esabınan klastın' nusqaları bolg'an ob'ektler u'stinde birdey a'meller ko'pligin orinlaw mumkin boladı.

6. Klasslar miyraslıq shejiresi dep atalıwshı jalg'ız terek ko'rinishidegi dizimdi quraydı. Anıq bir klass nusqaları menen baylanıslı yad ha'm a'meller shejire terektin' to'mengi bo'liminde jaylasqan qa'legen miyraslıq klassqa tiyisli boladı.

Ob'ektke mo'lsherlengen yondoshuvning ush printspi': Ob'ektke mo'lsherlengen yondoshuv (OMYO) ni tu'sinip jetiw ha'mde wodan

paydalani'wdi' uzlashtirib ali'w ushi'n, aldi'nam bar puqta bazali'q bilimlerde iyelep ali'w kerek. Bazali'q tu'siniklerde puqta an'lap yetibgina, programmalarde jarati'wda OMYO ni qo'llaw mu'mkin. Inkapsulyalash, miyrasli'q ha'm polimorfizm ob'ektke mo'lsherlengen programmalarde'ri'w (OMD) ninguchta bazali'q iushunchasi yesaplanadi'.

Inkapsulyalash

Inkapsulyalash programmani' qanday da monolit, bo'lindis zat si'pati'nda ali'p qaramay, ko'plegen wo'zbetinshe elementlerge boli'w imkaniyatini beredi. Ha'r bir element wo'z funksiyalari'n basqa elementlerden wo'zbetinshe tu'rde atqara alatug'i'n bo'lek modul si'pati'nda ali'p qaraladi. A'yne inkapsulyalash sebepli g'a'rezsizlik da'rejesi ortadi, sebebi ishki detallar interfeys arti'nda jasi'ri'ng'an boladi'.

Inkapsulyalash modullikning ob'ektke mo'lsherlengen suwretlewi boli'p tabi'ladi'. Inkapsulyalash ja'rdeminde programmali'q ta'miynatti' ma'lum funksiyalardi' wori'nlawshi' modullarga boli'p taslaw mu'mkin. Bul funksiyalardi' a'melge asi'ri'w detallari bolsa si'rtqi' olamdan jasi'ri'n halda boladi'.

Ma'nisan *inkapsulyalash* termini «germetik jasi'ri'lg'an; si'rtqi' ta'sirinlerden qorg'awlang'an programma bo'legi» degen ma'nisdi an'latadi'.

Yeger qandayda-bir programmali'q ob'ektke inkapsulyalash qollang'an bolsa, wol halda bul ob'ekt qara quti' si'pati'nda ali'p qaraladi. Siz qara quti' ne yetip ati'rg'ani'n woni'n' si'rtqi' interfeysini ko'rip turg'ani'n'i'z ushi'ng'ana biliwin'iz mu'mkin. Qara quti' qandayda zat yetiwge majburlash ushi'n, wog'an xabar jiberiw kerek. Qara quti' ishinde ne ju'z boli'p ati'rg'ani' a'hmiyetli yemes, qara quti' jiberilgen xabarg'a adekvat (uyqas tu'rde) munasa'bette boli'wi' za'ru'rlilew boli'p tabi'ladi'.

Interfeys si'rtqi' a'lem menen du'zilgen ayriqsha pitim boli'p, wol jag'dayda si'rtqi' ob'ektlar usi' ob'ektke qanday talaplar jiberiwi mu'mkinligi ko'rsetilgen boladi'. Interfeys-ob'etni basqari'w pulti.

G'alabali'q, jeke ha'm qorg'awlang'an kirisiw.

Qanday da bir elementti g'alabali'q interfeysga kirgiziw yamasa, kerisinshe, wodan shi'g'ari'w ushi'n, gilt so'zden paydalani'w kerek. OMD ning ha'r bir tilinde gilt so'zler kompleksi belgilengen, biraq bul so'zler tiykari'nan birdey funktsiyalardi' atqaradi'.

Ob'ektke mo'lsherlengen tillerdin' ko'bisinde kiriwdin' ush da'rejesi a'meldegi.

-G'alabali'q (public) -barli'q ob'ektler ushi'n kiriw ushi'n ruxsat bar.

-Qorg'awlang'an (protected) -tek usi' ekzemplyarga ha'm ha'r qanday tarmaq klasslarga kiriwge ruxsat bar.

-Jeke (private) -tek usi' ekzemplyarga kiriwge ruxsat bar.

Proekt da kiriw da'rejesin tuwri' tan'lap ali'w wog'ada za'ru'rli boli'p tabi'ladi'. Ko'rinetug'i'n yetiliwi kerek bolg'an barli'q zat g'alabali'q bo'lmog'i kerek. Berkilishi kerek bolg'an ha'r qanday zat qorg'awlang'an yamasa jeke kiriwge iye bo'lmog'i kerek.

Inkapsulyalash ne ushi'n kerek?

Inkapsulyalashdan tuwri' paydalani'w sebepli ob'ektler menen wo'zgartiriletug'i'n komponentler (strukturali'q bo'limler) dek ma'mile yetiw mu'mkin. Basqa ob'ekt sizdn' ob'ektingizdan paydalana ali'wi' ushi'n, wol sizdn' ob'ektingizning g'alabali'q interfeysidan qanday fodalanish kerekligin biliwi jetkilikli. Bunday g'a'rezsizlik ush za'ru'rli arti'qmashi'li'qka iye.

-G'a'rezsizlik sebepli, ob'ektden ta'kiraran paydalani'w mu'mkin. Inkapsulyalash puqta a'melge asi'ri'lg'an bolsa, ob'ektler ma'lum bir programmaga baylanisi'p qalg'an bolmaydi'. Wolardan mu'mkinshiligi bolg'an barli'q jerde paydalani'w mu'mkin boladi'. Ob'ektden basqa qandayda wori'nda paydalani'w ushi'n, woni'n' interfeysidan paydalani'p qo'ya qali'w jetkilikli.

-Inkapsulyalash sebepli, ob'ektte basqa ob'ektler ushi'n ko'rinmas bolg'an wo'zgerislerdi a'melge asi'ri'w mu'mkin. Yeger interfeys wo'zgartirilmasa, barli'q wo'zgerisler ob'ektden paydalanayotganlar ushi'n ko'rinmas boladi'. Inkapsulyalash komponentti jaqsi'law, a'melge asi'ri'w natiyjeliligini ta'miyinlew, qa'telerdi saplasti'ri'w imkaniyati'n beredi, tag'i' bulardi'n' ha'mmesi programmani'n' basqa ob'ektlerine ta'sir ko'rsetpeydi.

Ob'ektden paydalani'wshi'lar wolarda a'melge asi'ri'li'p ati'rg'an barli'q wo'zgerislerden avtomatik ta'rzde yutadilar.

-Qorg'awlang'an ob'ektden paydalani'wda ob'ekt ha'm programmani'n' basqa bo'legi wortasi'nda qandayda-bir na'zerde taza wo'z-ara baylani'slar boli'wi' mu'mkin yemes. Yeger ob'ekt basqalardan aji'rati'lg'an bolsa, bul halda wol programmani'n' basqa bo'legi menen tek wo'z interfeysi arqali' baylani'sqa kirisiwi mu'mkin.

Sunday yetip, inkapsulyalash ja'rdeminde modulli programmalar di' jarati'w mu'mkin. Na'tiyjeli inkapsulyalashning ush ayriqsha belgisi qo'yidagicha:

- abstrakciya;
- yengiziw din' jasi'ri'lg'anli'g'i';
- juwapkershiliktn' boli'ng'anli'g'i'.

Abstrakciya

Garchi ob'ektke mo'lsherlengen tiller inkapsulyalashdan paydalani'wg'a ja'rdem bersa-de, biraq wolar inkapsulyalashni kepilliklamaydi. Ha'kimnin' qol asti'ndag'i' ha'm i'senimsiz kodni' jarati'p qoyi'w an'sat. Na'tiyjeli inkapsulyalash-qi'yqi'mli'q menen islep shi'g'i'w ha'mde abstrakciya ha'm ta'jiriybeden paydalani'w na'tiyjesi. Inkapsulyalashdan na'tiyjeli paydalani'w ushi'n programmani' islep shi'g'i'wda aldi'n abstrakciyadan ja'ne woni'n' menen baylani'sli' kontseptsiyalardan paydalani'w di' u'yreniw kerek.

Abastraktsiya quramali' ma'seleni a'piwayi'lasti'ri'w procesi boli'p tabi'ladi'. Arnawli' bir ma'seleni sheshiwge kiriwar ekansiz, siz barli'q detallarni yesapqa ali'wg'a urinmaysiz, ba'лки sheshimdi an'satlashtiradiganlarini tan'lap alasi'z.

Aytaylik, siz jol ha'reketi modelin du'ziwin'iz kerek. Soni'si' ayan, bul wori'nda siz svetoforlar, mashinalar, shosselar, bir ta'repleme ha'm ikkitomonlama ko'sheler, hawa rayi' sharayatlari' ha'm t.b. klasslari'n jaratasi'z. Usi' elementlerdin' ha'r biri transport ha'reketine ta'sir ko'rsetedi. Biraq bul wori'nda xasharotlar ha'm qus da yo'lda payda boli'wi' mu'mkin bolsa-de, siz wolardi'n' modelin jaratmaysiz.

Inchunin, siz mashinalar markalari'n da aji'rati'p ko'rsetpeysiz. Siz haqi'yqi'y a'lemdi a'piwayi'lashtirasiz ha'mde woni'n' tek tiykarg'i' elementlerinen paydalanasi'z. Mashina-modeldin' za'ru'rli detali, biraq bul Kadillakmi yamasa basqa qandayda markadag'i' mashinama, jol ha'reketi modeli ushi'n bul detallar arti'qsha.

Abstrakciyani'n' yekew a'bzal ta'repi bar. Birinshiden, wol ma'sele sheshimin a'piwayi'lasti'radi'. Keregi tag'i' sonda, abstrakciya sebepli programmali'q ta'miynat komponentlerinen ta'kiraran paydalani'w mu'mkin. Ta'kiraran qollani'latug'i'n komponentlerdi jarati'wda wolar a'detde wog'ada qa'nigeliklesedi. Yag'ni'y komponentler qandayda-bir ma'lum ma'sele sheshimine mo'lsherlengeni, tag'i' wolar kereksiz wo'z-ara baylani'sli'li'qda bolg'ani' sebepli. programma fragmentining basqa qandayda wori'nda ta'kiraran qollani'wi' qi'yi'nlasadi'. Mu'mkinshiligi boricha bir qatar ma'selelerdi sheshiwge qarati'lg'an ob'ektlerdi jarati'li'wma ha'reket yetin'. Abstrakciya bir ma'sele sheshimiden usi' tarawdag'i' basqa ma'selelerdi da sheshiwde paydalani'w imkaniyati'n beredi.

Yekew mi'saldi' ko'rip shi'g'ami'z.

Birinshi mi'sal: bank kassiriga na'wbette turg'an adamlardi' woyda sawlelendiriw yetin'. Kassir bosaganda, woni'n' tu'n'ligine na'wbette turg'an birinshi klient jaqi'nlasadi'. Sonday yetip, na'wbettegi ha'mme adam birin-ketin kassir tu'n'ligi ta'rep ji'lji'ti'ladi'. Na'wbette turg'anlar «birinshi kelgengge birinshi boli'p xi'zmet ko'rsatildi» algoritmi boyi'nsha ji'lji'ti'li'p baradi'.

Yekinshi mi'sal: jarani'n' zaqi'mlang'an jerinin' irin'lesiwixonada gamburgerli konveyerni ko'rip shi'g'ayi'q. Na'wbettegi jan'a gamburger konveyerga kelip tu'skende, wol gamburgerlar qatari'ndag'i' son'g'i' gamburger qasi'ndan jay aladi'. Sol sebepli konveyerdan ali'ng'an gamburger wonda basqalari'nan ko'birek waqi't jatqan boladi'. Restoranlar «birinshi kelgengge birinshi boli'p xi'zmet ko'rsatildi» algoritmi boyi'nsha isleydi.

Garchi bul mi'sallar birotala ha'r qi'yli'sha bolsa-de, wolarda qanday da uli'wma printsp qollang'an boli'p, wodan basqa jag'daylarda da

paydalani'w mu'mkin. Basqasha yetip aytqanda, siz abstrakciyag'a kelesiz.

Bul mi'sallardi'n ha'r yekewinde da «birinshi kelgenge birinshi boli'p xi'zmet ko'rsatildi» algoritmi qollang'an. Bul wori'nda gezek elementi neni bildiriwi za'ru'rli yemes. Rasi'nda usi' element gezek aqi'ri'na kelip qosi'li'wi' ha'mde gezekti woni'n' basi'na jetkende tark yetiwigine za'ru'rli boli'p tabi'ladi'.

Abstrakciya ja'rdeminde bir ret gezekti jarati'p, keyinirek woni' basqa programmalar di' jazi'wda qo'llaw mu'mkin, bul programmalarda elementlerge «birinshi kelgenge birinshi boli'p xi'zmet ko'rsatildi» algoritmi boyi'nsha ishlov beriledi.

Na'tiyjeli abstrakciyani' wori'nlaw ushi'n bir neshe qag'i'ydalardi' an'lati'w mu'mkin.

-Qanday da ani'q jag'dayni' yemes, uli'wma jag'dayni' ali'p qarang.

-Ha'r qi'yli' ma'selelerge ta'n bolg'an uli'wma ta'repti izlep tabi'n'. Shnchaki bo'lek ha'diyseni yemes, tiykarg'i' printspti' ko're biliwge ha'reket yetin'.

-Garchi abstrakciya wog'ada qi'mbatli' bolsa-de, biraq yen' sheshimli ma'seleni yodingizdan shi'g'armang.

-Abstrakciya ha'mme waqi't da ashi'q-oydin yemes. Ma'seleni yechar ekansiz, siz birinshi, yekinshi ha'm, xatto, u'shinshi retine da abstrakciyani' ten'ib ololmasligingiz mu'mkin.

-A'wmetsizlikka tayi'n turing. A'melde ha'r bir jag'day ushi'n tuwri' keletug'i'n abstrakt programmani' jazi'w mu'mkin yemes.

Abstrakciyani' son'g'i' maqset si'pati'nda yemes, ba'lki wog'an yerisiw joli'ndag'i' qural si'pati'nda ali'p qaraw kerek. Arnawli' bir jag'daylarda abstrakciyani' qo'llaw kerek yemes. Yashigina evristik qag'i'ya a'meldegi, boli'p, wog'an ko're, yeger siz qandayda bir ma'seleni wo'z-ara uqsaw usi'llar menen keminde u'sh ret yechgan bolsan'i'z, abstrakciyani' tek sonday ma'selelerge qo'llaw usi'ni's yetiledi.

Abstrakt komponentti ta'kiraran qo'llaw an'satlaw, sebebi wol qandayda-bir bir ayriqsha ma'seleni sheshiwge yemes, ba'lki qatar ma'selelerdi sheshiwge mo'lsherlengen. Biraq bul hal komponentden

jaysha ta'kiraran paydalani'wdan ko're ko'birek inkapsulyalashga tiyisli. Ishki detallarni jasi'ri'wg'a u'yreniw wog'ada za'ru'rli boli'p tabi'ladi'. Mag'li'wmatlardi'n' abstrakt tu'rlerin qo'llaw inkapsulyalashni na'tiyjeli qo'llawg'a mu'mkinshilik beredi.

Yengiziwdi jasi'ri'w ja'rdeminde si'rlardi' jasi'ri'w.

Abstrakciya na'tiyjeli inkapsulyalashning strukturali'q bo'limlerinen biri tek. Si'rtqi' ta'sirinlerden zinhar qorg'awlanbag'an abstrakt programmani' da jazi'w mu'mkin. Nag'i'z wo'zinin' ushi'n ob'ekttin' ishki yengizilishini jasi'ri'w kerek boladi'.

Yengiziwdin' jasi'ri'lg'anli'g'i'

Yengiziwdin' jasi'ri'lg'anli'g'i' yekew arti'qmashi'li'qka iye:

- ob'ektlerdi paydalanuvchilardan qorg'awlaydi';
- paydalanuvchilarni ob'ektlerden qorg'awlaydi'.

Birinshi arti'qmashi'li'q-ob'ektlerdi qorg'awdi' ko'rip shi'g'ami'z.

Tu'p inkapsulyalash til da'rejesinde apparat til konstruktsiyalari ja'rdeminde ta'miyinlenedi.

Mag'li'wmatlardi'n' abstrakt tu'rleri-bul mag'li'wmatlar ha'm wolar u'stinde wo'tkeriladigan operatsiyalar kompleksi.

Mag'li'wmatlardi'n' abstrakt tu'rleri ishki informaciya ha'm jag'dayni' puqta islep shi'g'i'lg'an interfeys arti'nda yashirar yeken, wolar tilde mag'li'wmatlardi'n' jan'a tu'rlerin ani'qlawg'a mu'mkinshilik beredi. Bunday interfeysda mag'li'wmatlardi'n' abstrakt tu'rleri bo'lindis pu'tkillik si'pati'nda usi'ni's yetilgen. Mag'li'wmatlardi'n' abstrakt tu'rleri inkapsulyalashni ko'llashni an'satlasti'radi', sebebi wolar sebepli inkapsulyalashni miyrasli'qsi'z ha'm polimorfizmsiz qo'llaw mu'mkin, bul bolsa inkapsulyalashning a'yne wo'zine di'qqatti' qarati'w imkaniyati'n beredi. Mag'li'wmatlardi'n' abstrakt tu'rleri soni'n' menen birge tu'r tu'siniginin' qollani'wi'n da an'satlasti'radi'. Yeger tu'r ne yekenin an'lap alsak, bul halda ob'ektke mo'lsherlengen yondoshuv qa'nigelestirilgen paydalanuvchilik tu'rleri ja'rdeminde tildi ken'eytiwdin' ta'biy usi'li'n usi'ni's yetip ati'rg'ani'n an'sat sezib ali'w mu'mkin.

Programmatalasti'ri'wda qatar wo'zgeriwshiler jarati'ladi' ha'm wolarg'a bahalar beriledi. Tu'rler ja'rdeminde programma ushi'n qolay bolg'an ha'r qi'yli' ko'rinistegi bahalar ani'qlanadi'. Sonday yetip, tu'rler programma komponentlerinen biri dep bu'ydedew mu'mkin boladi'. A'piwayi' tu'rlerge mi'sal si'pati'nda pu'tkil, uzi'n ha'm ju'ziwshi tu'rlerdi keltiriw mu'mkin. Wo'zgeriwshinin' tu'ri usi' wo'zgeriwshi qanday bahalardi' ali'wi' ja'ne woni'n' u'stinde qanday operatsiyalarni wori'nlaw mu'mkinligin belgilep beredi.

Tu'rler programmada qo'llaw mu'mkin bolg'an wo'zgeriwshiler tu'rin ani'qlap beredi. Usi' tu'rdegi wo'zgeriwshi qanday jol qoyi'latug'i'n bahalarg'a i'yelewi mu'mkinligin tu'r belgilep beredi. Tu'r tekg'ana jol qoyi'latug'i'n bahalar salasi'n, ba'lki usi' wo'zgeriwshi u'stinde qanday operatsiyalarni wori'nlaw mu'mkinligi, soni'n' menen birge ali'natug'i'n na'tiyjeler qanday tu'rde boli'wli'g'i'n da belgilep beredi.

Tu'rler-yesaplarda bir pu'tkillik si'pati'nda a'mel yetetug'i'n zat. Mi'sali', pu'tkil sanni' alayli'q. Yekew pu'tkil sanni' qosar ekansiz, garchi bul sanlar kompyuter yadi'nda bi'ytlar ko'rinisinde ko'rinetug'i'n bolsa-de, siz bi'ytlar u'stindeggi operatsiyalar haqqi'nda bas qotirib o'tirmaysiz,

Yengiziw jasi'ri'lg'an bolg'ani' sebepli, ob'ekt na'zerde taza ha'm destruktiv (strukturani' buzatug'i'n) paydalani'wdan qorg'awlang'an boladi'. Bul yengiziw jasi'ri'lg'anli'g'i'ni'n' abzalli'qlari'nan biri boli'p tabi'ladi'. Biraq yengiziwdin' jasi'ri'lg'anli'g'i' ob'ektlerden paydalani'wshi'lar ushi'n da za'ru'rli boli'p tabi'ladi'.

Yengiziwdin' jasi'ri'lg'anli'g'i' programmani' maslasi'wshi' yetedi, sebebi paydalani'wshi'lar ob'ektin' yengizilishini yesapqa ali'wg'a majbur yemesler. SHunday yetip, yengiziwdin' jasi'ri'lg'anli'g'i' tekg'ana ob'ektti qorg'awlaydi', ba'lki ku'shsiz baylani'sqan kodni' jarati'li'wma ja'rdem berip, usi' ob'ektden paydalani'wshi'lar ushi'n arnawli' bir qolaysi'zli'qlardi' chetlab wo'tiw imkaniyati'n beredi.

Ku'shsiz baylani'sqan kod-bul basqa komponentlerdin' yengizilishiga baylani'sli' bolmag'an kod.

Ku'shli baylani'sqan kod yokibevosita baylani'slarg'a iye kod-bul basqa komponentlerdin' yengiziliwi menen ti'g'i'z baylani'sli' bolg'an kod.

Inkapsulyalash ha'm yengizilishning jasi'ri'lg'anli'g'i'-ka'ramat yemes. Interfeys wo'zgartirilgende, yeski interfeysga baylani'sli' bolg'an yeski kodni' da wo'zgartiw kerek boladi'. Yeger programmani' jazi'wda detallar interfeysda jasi'ri'lg'an bolsa, buni'n' na'tiyjesinde ku'shsiz baylani'sqan programma ju'zege keledi.

Ku'shli baylani'sqan programmada inkapsulyalashning abzalli'qlari' jog'aladi': wo'zbetinshe ha'm ta'kiraran qollani'latug'i'n ob'ektlerdin' jarati'li'wi' mu'mkin bolmaydi'.

Yengiziwdin' jasi'ri'lg'anli'g'i' wo'z kemshiliklerine da iye. Geyde interfeys ja'rdeminde ali'w mu'mkin bolg'ani'nan ko'birek informaciya kerek boli'p qoradi. Programmalar a'leminde ma'lum ani'qli'q menen, yag'ni'y ma'lum bir tuwri' keletug'i'n razryadlar mug'dari' menen isleytug'i'n qara qutilar kerek. Mi'sali', sonday jag'day ju'z beriwi mu'mkin, sizge 64 bitli pu'tkil sanlar kerek boli'p qaladi', sebebi siz ku'ta' u'lken sanlar u'stinde a'meller atqarayapsiz. Interfeysni belgilewde, woni' usi'ni'wg'ana yemes, ba'lki yengiziwde qollang'an tu'rlerdin' ayriqsha ta'replerin hu'jjetlestiriw da wog'ada za'ru'rli boli'p tabi'ladi'. Biraq, g'alabali'q interfeysning ha'r qanday basqa bo'legi si'yaqli', minez-quli'qdi' bellegennen keyin, woni' wo'zgartirip bolmaydi'.

Yengiziwdi jasi'ri'p, wo'zbetinshe, basqa komponentler menen ku'shsiz baylani'sqan programmani' jazi'w mu'mkin. Ku'shsiz baylani'sqan programma bekkemlew boladi', bunnan ti'sqari' woni' modifikatsiya yetiw da an'satlaw. Bular sebepli bolsa woni' ta'kiraran qo'llaw ha'm jetilistiriw an'sat keshedi, sebebi sistemani'n' bir bo'legidegi wo'zgerisler woni'n' basqa wo'zbetinshe bo'limlerine ta'sir yetpeydi.

Juwapkershiliktn' boli'ng'anli'g'i'

Yengiziwdin' jasi'ri'lg'anli'g'i' juwapkershilik tu'sinigi menen baylani'sli'li'g'i' ta'biy boli'p tabi'ladi'. Ku'shsiz baylani'sqan programmani' jarati'w ushi'n, juwapkershilikti tiyisli tu'rde bo'listiriw da za'ru'rli boli'p tabi'ladi'. Juwapkershilik tiyisli tu'rde bo'listirilgende, ha'r bir ob'ekt wo'zi juwapker bolg'an bir funktsiyani' atqaradi' ha'mde bul funktsiyani' jaqsi' atqaradi'. Bul bolsa ob'ekt bir pu'tkillikti sho'lkemlestiriwin da an'latadi'.

Basqasha yetip aytqanda, funktsiyalar ha'm wo'zgeriwshilerdin' tosi'narli' kompleksine mu'tajlik bolmaydi'. Inkapsulalanayotgan ob'ektler wortasi'nda jaqi'n kontseptual baylani's bo'lmog'i kerek. Barli'q funktsiyalar uli'wma wazi'ypani' atqarmog'i kerek.

Yengiziw berkitilmas yeken, juwapkershilik ob'ektden shetke shi'g'i'p ketiwi mu'mkin. Biraq wo'z wazi'ypasi'n qanday sheshiwdi a'yne ob'ekttin' wo'zi biliwi kerek, yag'ni'y a'yne ob'ekt wo'z wazi'ypasi'n wori'nlaw algoritmiga i'yelewi kerek. Yeger yengiziw ashi'q qaldi'rilsa, paydalani'wshi' wodan tuwri'dan-tuwri' paydalani'wi' ha'm soni'n' menen juwapkershilikti boli'wi' mu'mkin.

Yeger yekew ob'ekt birdey wazi'ypani' atqarsa, sonday yeken juwapkershilik tiyislishe bo'liwlenbegen boladi'. Programmada arti'qsha logikali'q sxemalar a'meldegi bolsa, woni' qayta islew kerek boladi'.

Turmi'sda bolg'ani' si'yaqli', bilimler ha'm juwapkershilik jumi'sti' qanday yetip jaqsi' wori'nlaw mu'mkinligin bilgan kisige kepillik yetiliwi kerek. Bir ob'ektke bir (ha'r halda kem mug'dardag'i') wazi'ypa ushi'n juwapkershilikti ju'klew kerek. Yeger bir ob'ektke ko'p mug'dardag'i' wazi'ypalar u'stinen juwapkershilik ju'kletip qoyi'lg'an bolsa, wolardi' wori'nlaw quramali'lasadi', ob'ektti gu'zetip bari'w ha'm jetilistiriw da qi'yi'nlasadi'. Juwapkershilikti wo'zgertiw da havfli, sebebi bunda, yeger ob'ekt bir neshe minez-quli'q liniyalariga iye bolsa, wolardi' da wo'zgertiwge tuwri' keledi. Na'tiyjede ku'ta' u'lken mug'dardag'i' informaciya bir jerge jamlanib qaladi', woni' bolsa ten' bo'listiriw kerek. Ob'ekt ku'ta' u'lkenlashib ketken jag'daylarda, wol a'melde mutaqlil programmag'a shen'berdi ha'mde procedurali' programmalasti'ri'w abzalli'qlari'nan paydalani'w menen birge woni'n' barli'q duzaqlari'na da ilinib qali'wi' mu'mkin boladi'. Na'tiyjede siz inkapsulyalash uluwma qo'llanbag'an programmada ju'zege keletug'i'n barli'q ma'selelerge dus kelip qolasiz.

Ob'ekt bir-yekewden arti'q wazi'ypa ushi'n juwapker yekenin ani'qlagach, juwapkershiliktn' bir bo'legin basqa ob'ektke ali'p wo'tiw kerek.

Yengiziw din' jasi'ri'lg'anli'g'i'-na'tiyjeli inkapsulyalash joli'ndag'i' qa'demlerden biri tek. Juwapkershilikti tiyisli tu'rde bo'liwlamay, na'tiyjede siz proceduralar dizimine iye boli'p qolasiz tek.

Na'tiyjeli inkapsulyalash=abstraktsiya+joriy yetiw din' jasi'ri'lg'anligi+mas'uliyat.

Abstrakciyani' ali'p tastap, programmadan ta'kiraran paydalani'p bolmaydi'. Yengiziw din' jasi'ri'lg'anli'g'i'n ali'p tastap siz ku'shli baylani'sqan programmag'a iye bolasi'z. Aqi'r-aqi'betde, juwapkershilikti ali'p taslaw na'tiyjesinde bolsa siz procedurali', mag'li'wmatlar ishloviga mo'lsherlengen, woraylashmagan ku'shli baylani'sqan programmag'a iye bolasi'z.

Inkapsulyalash: u'lgili qa'teler

Abstrakciyani' wog'ada da'rejede qo'llaw sinfni jazi'wda ma'lum mashqalalardi keltirip shi'g'ari'wi' mu'mkin. Barli'q paydalanuvchilarga ha'mde barli'q jag'daylarda birdey tuwri' keletug'i'n sinfni jazi'w mu'mkin yemes.

Wo'reski abstraktsiyalash da havfli boli'wi' mu'mkin. Ha'tte yeger siz qandayda-bir elementtin' islep shi'g'ili'wi'nda abstrakciyadan paydalang'an bolsan'i'z, wol sol bir elementte da barli'q jag'daylarda isley olmasligi mu'mkin. Paydalani'wshi'ni'n' barli'q mu'tajliklerin qandira alatug'i'n sinfni jarati'w ju'da' qi'yi'n. Abstrakciyag'a o'ralashib qali'w kerek yemes, birinshi na'wbette qoyi'lg'an ma'seleni sheshiw kerek.

Klasqa ma'seleni sheshiw ushi'n kerek bolg'ani'nan ko'birek zatti' kirgiziw usi'ni's yetin'meydi. Birdaniga barli'q ma'selelerdi yechmang, itibari'n'i'zdi' birewinin' sheshimine qarating. Ha'm sonnan son'g'i'na yetip boli'ng'an jumi'sqa sali'sti'rg'anda abstrakciyani' qo'llaw usi'li'n izlep ko'riw mu'mkin.

Mi'sali', bahaybat yesaplar yamasa quramali' modelge uqsaw talay quramali' ma'seleler da ushraydi'. Bul wori'nda ga'p juwapkershilikti bo'listiriy noqati' na'zeriden quramali'li'q haqqi'nda barmaqta. Ob'ekttin' juwapkershilik tarawlari' qansha ko'p bolsa, wol sonsha quramali'law boladi' ja'ne woni' qollap-quwatlaw da talay quramali'li'q tug'diradi.

Ha'm, aqi'r-aqi'betde, programmalasti'ri'wda abtaktsiyalashdan paydalani'wg'a u'yreniw ushi'n waqi't kerek. Haqi'yqi'y abstrakt programma haqi'yqi'y turmi's talaplari'na tiykarlang'an bo'lmog'i kerek. Wol programmashi' jaysha ta'kiraran qollani'latug'i'n ob'ektti jarati'li'wma ti'yanaqli' yetkenligi na'tiyjesinde ju'zege kelmeydi. Aytqanlaridek, woylap tabi'wg'a mu'tajlik tuwi'lganidagina, wol tuwi'ladi'. Tap sol printsp ob'ektlerdi jarati'wda da a'mel yetedi. Birinshi retdayoq rasi'nda abstrakt, ta'kiraran qollani'latug'i'n ob'ektti jazi'w mu'mkin yemes. A'detde ta'kiraran qollani'latug'i'n ob'ektler ishda si'naqdan wo'tip bolg'an ha'mde ko'plegen wo'zgerislerge dus kelgen programmani' jetilistiriw processinde jarati'ladi'.

Ishki wo'zgeriwshilerdi ha'mme waqi't jasi'ri'w kerek: wolar konstantalar bolg'an jag'daylar bunnan ayri'qshali'q. Keregi, wolar tek'g'ana jasi'ri'lg'an boli'wi' kerek, ba'lki wolarg'a tek klassti'n' wo'zi kiritilgan ruxsati'na i'yelewi kerek. Ishki wo'zgeriwshilerga kiritilgan ruxsat berilgende, yengiziw ashi'ladi'.

Ishki mag'li'wmatlari' basqa at asti'nda si'rtqi' paydalani'w ushi'n usi'ni's yetilgan interfeysni jarati'li'wma mu'tajlik joq. Interfeys joqari' da'rejedegi minez-quli'q jolli'qg'a i'yelewi kerek.

Inkapsulyalashning tiykarg'i' abzalli'qlari'

Inkapsulyalash ja'rdeminde juwapkershilikni insan noqati' na'zeriden ta'biy ko'ringen usi'l menen bo'listiriw mu'mkin. Abstrakciyadan paydalani'p, ma'sele sheshimin yengiziw terminlerinde yemes, ba'lki usi' yechilayotgan ma'sele tiyisli bolg'an tarawdi'n' terminlerinde an'lati'w mu'mkin. Abstrakciya ma'seledagi za'ru'rli ta'repti aji'rati'p ko'rsetiw imkaniyati'n beredi.

Kodni'n' za'ru'rli uchastkalarini to'sib ha'm yengizilishni jasi'ri'p, ha'r bir bo'lek komponenttin' tuwri'li'g'i'n tekserip ko'riw mu'mkin. Tekserilgan komponent qollanganda, ha'r bir modulni di'qqat penen tekseriw mu'mkinshiligi tuwi'ladi', bul bolsa pu'tkil programmani'n' isenimli yekenine shubha qaldi'rmaydi. SHunday bolsa-

de, programma tuwri' islep atirg'ani'na isenim boli'w ushi'n, uli'wma tekseriw za'ru'r.

Ta'kiraran qo'llaw mu'mkinshiligi: abstrakciya ja'rdeminde ha'r qi'yli' jag'daylarda qo'llaw ushi'n jaramli' bolg'an an'sat wo'zgartiriletug'i'n programmani' jarati'w mu'mkin.

Gu'zetip bari'wdag'i' qolayli'q: qorg'awlang'an programmani' gu'zetip bari'w an'sat. Ha'kimnin' qol asti'ndag'i' kodni' wo'zgartirmay turi'p, klassti'n' yengizilishiga ha'r qanday kerekli wo'zgerislerdi kirgiziw mu'mkin. Bul wo'zgerisler yengizilishdagi wo'zgerislerdi da, interfeysga jan'a usi'llardi' qosi'wdi' da wo'z ishine ali'wi' mu'mkin. Tek interfeys semantikasi (mazmuni) ning wo'zgerisleri ha'kimnin' qol asti'ndag'i' koddag'i' wo'zgerislerdi talap yetedi.

Jetilistiriw: programmani' buzmay turi'p, yengizilishni wo'zgartiw mu'mkin. Basqasha yetip aytqanda, a'meldegi kodni'n' iskerligin saqlag'an halda funktsional si'pati'n suwretlewlerdi jetilistiriw mu'mkin. Buni'n' u'stine, yengiziw jasi'ri'lg'an yeken, rawajlani'wlasti'ri'lg'an komponentden paydalanayotgan kodni'n' jumi'sqa tu'sirilish si'pati'n suwretlewleri avtomatik ta'rzde jaqsi'lanadi': axir kod, garchi wol wo'zgermegen bolsa-de, rawajlani'wlasti'ri'lg'an komponentlerden paydalanadi-g'oy! Biraq wo'zgerisler kiritilgeninen keyin, tag'i' modulni tekseriw kerek boladi'. Ob'ekttin' wo'zgeriwi usi' ob'ekt paydalanayotgan pu'tkil kodta domino effektin keltirip shi'g'ari'wi' mu'mkin.

Jan'a versiyalarni udayi' ta'kirarlanatug'i'n shi'g'ari'w (baspayeti'w) qolayli'g'i': programmani' wo'zbetinshe modullarga boli'p, kodni' islep shi'g'i'w menen baylani'sli' wazi'ypani' bir neshe islep shi'g'i'wshi'lar wortasi'nda bo'listiriw ha'mde sol jol menen islep shi'g'i'w procesin tezlestirishga yerisiw mu'mkin.

Komponentlerdi islep ha'm tekserip shi'g'i'p, wolardi' tazadan qaytaldan wo'zgartiw kerek bolmaydi'. Sonday yetip, programmashi' bul komponentlerdi ta'kiraran qo'llawi' ha'mde wolardi' tag'i' «nul»den baslap jarati'w ushi'n waqi't sarflamasligi mu'mkin.

Miyrasli'q

Miyrasli'q a'meldegi bolg'an klassti'n' tariypi tiykari'ndayoq jan'a sinfni jarati'w imkaniyati'n beredi. YAngi klass basqasi' tiykari'nda jarati'lgach, woni'n' tariypi avtomatik ta'rzde a'meldegi klassti'n' barli'q qa'siyetleri, turpayi' ha'm yengizilishiga miyrasli'q yetedi. Aldi'n a'meldegi bolg'an klass interfeysining barli'q metodlari ha'm qa'siyetleri avtomatik ta'rzde miyrasxor interfeysida payda boladi'. Miyrasli'q miyrasxor klasi'nda qandayda-bir ta'repten tuwri' kelmegen minez-quli'qdi' aldi'nan ko're biliw imkaniyati'n beredi. Bunday paydali' ayri'qshali'q programmali'q ta'miynatti' talaplardi'n' wo'zgeriwine maslasti'iri'w imkaniyati'n beredi. Yeger wo'zgeritiwler kirgiziwge mu'tajlik tuwi'lsa, bul halda yeski klass funktsiyalari'na miyrasli'q yetiwshi jan'a klass jazi'p qo'ya qolinadi. Keyin wo'zgeritiliwi kerek bolg'an funktsiyalarg'a qaytaldan tariyp beriledi ha'mde jan'a funktsiyalar qosi'ladi'. Bunday worni'na wori'n qoyi'wdi'n' mazmuni' sonnan ibarat, wol da'slepki klass tariypini wo'zgartirmay turi'p, ob'ekt jumi'si'n wo'zgeritiw imkaniyati'n beredi. Axir bul halda qayta test si'naqlari'nan puqta wo'tkerilgen tiykarg'i' klasslarga tegmasa da boladi'-de.

Yeger siz ko'p retlab qo'llaw yamasa basqa qandayda maqsetlerge ko're miyrasli'qdi' qo'llawg'a ahd qilsangiz, aldi'n ha'r gezek qarang-miyrasxor-klass menen miyrasli'qdi' berip ati'rg'an klassti'n' tu'rleri wo'z-ara sa'ykes keledimi. Miyrasli'qda turlaning sa'ykes keliwi ko'binese «Iyis-a» testi dep ataladi'. Yekew klass birdey tu'rge iye bolg'andag'ana, wo'z-ara «Iyis-a» munasa'betinde turi'pti' dep yesaplanadi'.

Birinshi klass wo'zinde yekinshi klassti'n' ekzempliyariga iye bolg'andag'ana yekew klass wo'z-ara «Has-a» munasa'betinde turi'pti' dep yesaplanadi'.

Aytaylik, Canine (I'yt'larniki) bazali'q klasi' a'meldegi. Wol holda i'yt itniki boladi (A dog iyis a canine) . Sol sebepli Dog klasi' Canine klasi'ni'n' howo'kpe keseliasi bo'lmog'i kyerak. Soni'n' bilan birga itning

quyri'g'i' (Tail) bor (A dog has a tail) . Sol sebepli Tail klasi'ni'n' ekzyemplyari (nushasi) ni, masalan, Canine ga kiritip qoyi'w kyerak. Usi' biroz biologik mi'soldan ko'rinib turganidyek, «Ia-a» munosabati sinflarning tabaqalanishida namoyon boladi. Sinflar orasidagi «Has-a» munosabati yesa shunga olib kyeladiki, bir klass yekinshisida mavjud boladi.

Boshqa sinfga myerosxwo'r bolayotgan klass myeros byerayotgan klass bilan shunday munosabatda bo'lmog'i lozimki, bunda natijaviy munosabatlar wo'z ma'nosiga ega bo'lmog'i, yag'ni'y vorislik tabaqalanishiga amal yetiliwi kyerak.

Vorislik-sinflar o'rtasida «Iyis-a» munosabatlari o'rnatil's imkonini byeradigan myexanizm. Myerosxwo'r klass wo'z ajdodi bo'lgan sinfdan xususiyatlar va xulq-atvorni myeros yetip olayotganida, wol soni'n'dyek wo'z ajdodi bo'lgan klass ehtimol boshqa sinflardan myeros yetip olgan xususiyatlar va xulq-atvorga ham ega boladi.

Vorislik tabaqalanishi qanday boli'p tabi'ladi' ma'no kasb yetiwi ushi'n ajdodlar u'stidan qanday amallar bajarilgan bo'lsa, avlodlar u'stidan ham shunday amallar bajarilish imkoniyati boli'wi' lozim. Bul «Iyis-a» tyesti yordamida tyekshiriladi. Myerosxwo'r sinfga funktsiyalarni kyengaytirish va jan'alarini qosi'w ushi'n ruxsat byeriladi. Ammo unga funktsiyalarni chiqarib tashlashga ruxsat joq.

Vorislik yordamida qurilgan klass myetodlar va xususiyatlarning uchta ko'rinishiga ega boli'wi' mu'mkin:

-Worni'ga wori'n qoyi'w (almashtirish) : jan'a klass ajdodlarining myetodi yamasa xususiyatini shunchaki o'zlashtirib olmaydi, balki unga jan'a ta'rif ham byeradi;

-Jan'a: jan'a klass pu'tkillay jan'a myetodlar yamasa xususiyatlarni qosadi;

-Ryekursiv: jan'a klass wo'z ajdodlari myetodlari yamasa xususiyatlarini to'g'ridan-to'g'ri olib qo'ya qoladi.

Ob'yektga mo'ljallangan tillarning ko'bisi ta'rifni ma'lumot uzatilgan ob'yektdan qi'di'radilar. Agar wol yerdan ta'rif topishning iloji bo'lmasa, biron ta'rif topilmaguncha, qi'di'ri'w tabaqalar

boyi'cha yuqoriga ko'tarilavyeradi. Ma'lumotni boshqarish aynan shunday amalga oshiriladi hamda aynan sol tufayli wori'nga wori'n qoyi'w jarayoni jumi's ko'rsatadi.

Voris sinflar himoyalangan kiriw darajasiga ega bo'lgan myetodlar va xususiyatlarga kiriw ruxsati'n olishlari mu'mkin. Bazaviy sinfda faqat avlodlar foydalanishi mu'mkinligi aniq bo'lgan myetodlargagina himoyalangan kiriw darajasini byering. Basqa jag'daylarda jeke yamasa g'alabali'q kiriw da'rejesinden paydalani'w kerek. Bunday yondoshuv barli'q klasslarg'a, atap aytqanda, tarmaq klasslarg'a da kiriw ruxsati' berilgeninen ko're, bekkemlew konstruktsiyani jarati'w imkaniyati'n beredi.

Miyrasli'q tu'rleri

Miyrasli'q u'sh tiykarg'i' jag'daylarda qollani'ladi':

- 1.Ko'p retlab paydalani'wda;
- 2.ajrali'p turi'w ushi'n;
- 3.tu'rlerdi almasti'ri'w ushi'n.

Miyrasli'qdi'n' ayi'ri'm tu'rlerinen paydalani'w basqalari'nan ko're a'bzellew yesaplanadi'. Miyrasli'q jan'a klasqa yeski klassti'n' a'melde qollani'wi'nan ko'p retlab paydalani'w imkaniyati'n beredi. Kodni' qirqib taslaw yamasa kirgiziw worni'na, miyrasli'q kodqa avtomatik ta'rzde kiriwdi ta'miyinleydi, yag'ni'y kodqa kiriwde, wol jan'a klassti'n' bir bo'legi si'yaqli' ali'p qaraladi. Ko'p retlab qo'llaw ushi'n miyrasli'qdan paydalanar ekansiz, siz miyraslar yetip ali'ng'an realizatsiya (yengiziliw) menen baylani'sli' bolasi'z. Miyrasli'qdi'n' bul tu'rin i'qti'yatli'li'q menen qo'llaw kerek. YAxshisi bul wori'nda «Has-a» munasa'betinen paydalani'w kerek.

Pari'qlaw ushi'n miyrasli'q tek a'wlad-klass ha'm a'jdad-klass wortasi'ndag'i' parqlardi' programmalasti'ri'w imkaniyati'n beredi. Parqlardi' programmalasti'ri'w wog'ada qu'diretli qural boli'p tabi'ladi'. Kodlaw koleminin' kishiligi ha'm kodni'n' an'sat basqarilishi proekt islenbesin an'satlasti'radi'. Bul halda kod qatarlari'n kamroq jazi'wg'a tuwri' kelediki, bul qosi'latug'i'n qa'teler mug'dari'n da azaytadi'.

Almasti'ri'w mu'mkinshiligi-OMYO de za'ru'rli tu'siniklerden biri. Miyrasxor klasqa woni'n' a'jdadi bolg'an klasqa jiberiletug'i'n xabarlardi' jiberiw mu'mkin bolg'ani' ushi'n, wolardi'n' ha'r yekewine birdey munasa'bette boli'w mu'mkin. Nag'i'z wo'zinin' ushi'n miyrasxor sinfni jarati'wda minez-quli'qdi' shi'g'ari'p taslaw mu'mkin yemes. Almasti'ri'w mu'mkinshiligin qollap, programmag'a ha'r qanday tarmaq tu'rlerdi qosi'w mu'mkin. Yeger programmada a'jedad qollang'an bolsa, bul halda wol jan'a ob'ektlerden qanday fodalanishni biladi.

Polimorfizm

Yeger inkapsulyalash ha'm miyrasli'qdi' OMYO ning paydali' qurallari' si'pati'nda ali'p qaraw mu'mkin bolsa, polimorfizm -yen' universal ha'm radikal qural boli'p tabi'ladi'. Polimorfizm inkapsulyalash ha'm miyrasli'q menen bekkem baylani'sli', boz u'stine, polimorfizmsiz OMYO na'tiyjeli bo'lolmaydi. Polimorfizm-OMYO paradigmasida worayli'q tu'sinik boli'p tabi'ladi'. Polimorfizmni egallamay turi'p, OMYO den na'tiyjeli paydalani'w mu'mkin yemes.

Polimorfizm sonday jag'dayki, bunda qanday da bir zat ko'p si'rtqi' ko'rinishlarga iye boladi'. Programmalisti'ri'w tilinde «ko'p si'rtqi' ko'rinishler» deyilganda, bir at avtomatik mexanizm ta'repinen tan'lap ali'ng'an ha'r qi'yli' kodlardi'n' ati'nan jumi's ko'riwi tu'siniledi. Sonday yetip, polimorfizm ja'rdeminde bir at ha'r qi'yli' minez-quli'qdi' bildiriwi mu'mkin.

Miyrasli'q polimorfizmning ayi'ri'm tu'rlerinen paydalani'w ushi'n za'ru'r boli'p tabi'ladi'. A'yne wori'ndoshlik mu'mkinshiligi a'meldegi bolg'ani' ushi'n, polimorfizmdan paydalani'w mu'mkin boladi'. Polimorfizm ja'rdeminde sistemag'a tuwri' kelgen waqi'tta qosi'msha funktsiyalardi' qosi'w mu'mkin. Programmani' jazi'w waqi'tnda ha'tte tahmin yetilmegen funktsionallik menen jan'a klasslardi' qosi'w mu'mkin, buni'n' u'stine bulardi'n' ha'mmesin da'slepki programmani' wo'zgartirmay turi'p da a'melge asi'ri'w mu'mkin. Jan'a talaplarg'a an'satg'ana iykemlese alatug'i'n programmali'q qural degende, mine usi'lar tu'siniledi.

Polimorfizmning ush tiykarg'i' tu'ri a'meldegi:

- Qosi'w polimorfizmi
- Parametrik polimorfizm
- Arti'qsha ju'kleniw.

-Qosi'w polimorfizmini geyde sap polimorfizm dep da ataydilar. Qosi'w polimorfizmi soni'n' menen qi'zi'qli'ki, woni'n' sebepli tarmaq klass nushalari wo'zin ha'r qi'yli'sha tuti'wi' mu'mkin. Qosi'w polimorfizmidan paydalani'p, jan'a tarmaq klasslardi' kirgizgen halda, sistemani'n' turpayi'n wo'zgertiw mu'mkin. Woni'n' bas a'bzelligi sonda, da'slepki programmani' wo'zgertirmay turi'p, jan'a minez-quli'qdi' jarati'w mu'mkin.

A'yne polimorfizm sebepli yengiziwden ta'kiraran fodalanishni miyrasli'q menen a'ynelestiriw kerek yemes. Buni'n' warni'na miyrasli'qdan aldi'nam bar wo'z-ara almasi'ni'w munasa'betleri ja'rdeminde polimorf minez-quli'qg'a yerisiw ushi'n paydalani'w kerek. Yeger wo'z-ara almasi'ni'w munosbatlari tuwri' bellensa, buni'n' arti'nan a'l'bette ta'kiraran qo'llaw shi'g'i'p keledi. Qosi'w polimorfizmidan paydalani'p, bazali'q klasstan, ha'r qanday a'wladdan, soni'n' menen birge bazali'q klass qollaytug'i'n metodlardan ta'kiraran paydalani'w mu'mkin.

Parametrik polimorfizmdan paydalani'p, turdosh metodlar ha'm turdosh (universal) tu'rler jarati'w mu'mkin. Turdosh metodlar ha'm tu'rler da'lillerdin' ko'plegen tu'rleri menen isley alatug'i'n programmani' jazi'w imkaniyati'n beredi. Yeger qosi'w polimorfizmidan paydalani'w ob'ektti aqi'l yetiwge ta'sir ko'rsatsa, parametrik polimorfizmdan paydalani'w qollani'lai'p ati'rg'an metodlarga ta'sir ko'rsetedi. Parametrik polimorfizm ja'rdeminde, parametr tu'rin atqari'li'w waqi'ti'i'na shekem dag'aza yetpey turi'p, turdosh metodlar jarati'w mu'mkin. Metodlarning parametrik parametrleri bolg'ani' si'yaqli', tu'rlerdin' wo'zi da parametrik boli'wi' mu'mkin. Biraq polimorfizmning bunday tu'ri barli'q tillerde da uchrayvermaydi (C++da a'meldegi).

Arti'qsha ju'kleniw ja'rdeminde bir at ha'r qi'yli'sha metodlarni bildiriwi mu'mkin. Bunda metodlar tek mug'darlari' ha'm parametr tu'rleri menen pari'q yetedi. Metod wo'z da'lilleri (argumentlari) ga baylani'sli'

bolmag'anda, arti'qsha ju'kleniw paydali' boli'p tabi'ladi'. Metod ayriqsha parametrlar tu'rleri menen sheklenmeydi, ba'lki tu'r-tu'rli tu'rdegi parametrlarga sali'sti'rg'anda da qollani'ladi'. Mi'sali' max metodini ko'rip shi'g'ayi'q. Maksimal-turdosh tu'sinik boli'p, wol yekew arnawli' bir parametrlardi qabi'l yetip, wolardi'n' qay-qaysi'si' u'lkenlew yekenin ma'lum yetedi. Tariyp pu'tkil sanlar yamasa ju'ziwshi noqatli' sanlar sali'sti'ri'wlani'wi'na qaray wo'zgermeydi.

Polimorfizmdan na'tiyjeli paydalani'w ta'repke qoyil'g'an birinshi qa'dem bul inkapsulyalash ha'm miyrasli'qdan na'tiyjeli paydalani'w boli'p tabi'ladi'. Inkapsullashsiz programma an'satg'ana klasslardi'n' yengizilishiga baylani'sli' boli'p qali'wi' mu'mkin. Yeger programma klasslardi'n' yengiziliw aspektrlaridan birine baylani'sli' boli'p qolsa, tarmaq klassta bul usi'n tuwri'law mu'mkin bolmaydi'.

Vorislik-qosi'w polimorfizmining za'ru'rli tarkibiy bo'legi. Hamma vaqt bazaviy sinfga imkon darajada jaqi'nlashtirilgan darajada dasturlashga uringan holda, wori'nbosarlik munosabatlarini o'rnatishga harakat yetiw kyerak. Bunday usi'l dasturda ishlov byerilayotgan ob'yektlar turlari miqdorini osu'tadi.

Puxta o'ylab ishlab chiqilgan tabaqalawo'sindi wori'nbosarlik munosabatlarini o'rnatishga yordam byeradi. Uli'wma bo'leklarni abstrakt sinflarga olib shi'g'i'w kyerak hamda ob'yektlarni shunday dasturlash kyerakki, bunda ob'yektlarning ixtisoslashtirilgan nushalari yemas, balki ularning o'zlari dasturlashtirilsin. Bul kyeyinchalik har qanday voris sinfni dasturda qo'llash imkonini byeradi.

Agar til vositalari bilan intyeryeys va joriy yetin'ishni toli'q ajrati's mu'mkin bo'lsa, wol holda odatda mana sol vositalardan foydalawo'sindi kyerak, vorislikdan yemas. Intyeryeys va joriy yetin'ishni aniq ajramedicina, wori'nbosarlik imkoniyatlarini oshirish va soni'n' bilan polimorfizmdan foydalanishning jan'a imkoniyatlarini ochib byerish mu'mkin.

Biroq ko'p wori'nlarda tajribasiz loyihachilar polimorfizmni ku'shaytirish maqsatalg'ama xulq-atvorni juda baland tabaqaviy darajaga olib shi'g'i'wga urinadilar. Bul holda har qanday avlod ham

bul xulq-atvorni ushlab tu'ra oladi. SHuni esdan chiqarmaslik kyarakki, avlodlar wo'z ajdodlarining funktsiyalarini chiqarib tashlay olmaydilar. Dasturni yanada polimorf yetiw maqsatalg'ama puxta ryejalashtirilgan vorislik tabaqalarini buzi'w yaramaydi.

Hamma narsaning sezimob-kitobi bor. Haqiqiy polimorfizmning kamchiligi shundaki, wol unumdorlikni pasaytiradi. Polimorfizmdan foydalanganda dasturni bajarish paytida tyekshiruvlar o'tkazish talab yetin'adi. Bul tyekshiruvlar turlari statik ravishda byerilgan qiymatlarga ishlov byerishga qaraganda ko'proq vaqtni talab yetadi.

Since object-oriented programming was fundamental to the development of C++, it is important to define precisely what object-oriented programming is. Object-oriented programming has taken the best ideas of structured programming and has combined them with several powerful concepts that allow you to organize your programs more effectively. In general, when programming in an object-oriented fashion, you decompose a problem into its constituent parts. Each component becomes a self-contained object that contains its own instructions and data related to that object. Through this process, complexity is reduced and you can manage larger programs.

All object-oriented programming languages have three things in common: encapsulation, polymorphism, and inheritance. Although we will examine these concepts in detail later in this book, let's take a brief look at them now.

Encapsulation

As you probably know, all programs are composed of two fundamental elements: program statements (code) and data. *Code* is that part of a program that performs actions, and *data* is the information affected by those actions. *Encapsulation* is a programming mechanism that binds together code and the data it manipulates, and that keeps both safe from outside interference and misuse.

In an object-oriented language, code and data may be bound together in such a way that a self-contained *black box* is created. Within the box are all necessary data and code. When code and data are linked together in this fashion, an object is created. In other words, an *object* is the device that supports encapsulation.

Within an object, the code, data, or both may be private to that object or public. *Private* code or data is known to, and accessible only by, another part of the object. That is, private code or data may not be accessed by a piece of the program that exists outside the object. When code or data is *public*, other parts of your program may access it, even though it is defined within an object. Typically, the public parts of an object are used to provide a controlled interface to the private elements of the object.

Polymorphism

Polymorphism (from the Greek, meaning "many forms") is the quality that allows one interface to be used for a general class of actions. The specific action is determined by the exact nature of the situation. A simple example of polymorphism is found in the steering wheel of an automobile. The steering wheel (i.e., the interface) is the same no matter what type of actual steering mechanism is used. That is, the steering wheel works the same whether your car has manual steering, power steering, or rack-and-pinion steering. Therefore, once you know how to operate the steering wheel, you can drive any type of car. The same principle can also apply to programming. For example, consider a stack (which is a first-in, last-out list). You might have a program that requires three different types of stacks. One stack is used for integer values, one for floating-point values, and one for characters. In this case, the algorithm that implements each stack is the same, even though the data being stored differs. In a non-object-oriented language, you would be required to create three different sets of stack routines, calling each set by a different name, with each set having its own interface. However, because of polymorphism, in C++ you can create one general set of stack routines (one interface) that works for all three specific situations. This way, once you know how to use one stack, you can use them all.

Inheritance

Inheritance is the process by which one object can acquire the properties of another object. The reason this is important is that it supports the concept of hierarchical classification. If you think about it, most knowledge is made manageable by hierarchical (i.e., top-down) classifications. For example, a Red Delicious apple is part of the classification *apple*, which in turn is part of the *fruit* class, which is under the larger class *food*. That is, the *food* class possesses certain qualities (edible, nutritious, etc.) that also apply, logically, to its *fruit* subclass. In addition to these qualities, the *fruit* class has specific characteristics (juicy, sweet, etc.) that distinguish it from other food. The *apple* class defines those qualities specific to an apple (grows on trees, not tropical, etc.). A Red Delicious apple would, in turn, inherit all the qualities of all preceding classes, and would define only those qualities that make it unique.

Without the use of hierarchies, each object would have to explicitly define all of its characteristics. However, using inheritance, an object needs to define only those qualities that make it unique within its class. It can inherit its general attributes from its parent. Thus, it is the inheritance mechanism that makes it possible for one object to be a specific instance of a more general case.

C++ Implements OOP

As you will see as you progress through this book, many of the features of C++ exist to provide support for encapsulation, polymorphism, and inheritance. Remember, however, that you can use C++ to write any type of program, using any type of approach. The fact that C++ supports object-oriented programming does not mean that you can only write object-oriented programs. As with its predecessor, C, one of C++'s strongest advantages is its flexibility.

28-Leksiya. Klasslar. Klassni ha'm ob'ektlarni sa'wlelendiriwi. Klass maydanlari ha'm metodlari.

Joba:

1. Klass sintaksisi
2. Nusqalaw konstruktori
3. Jaylastiriladug'un (inline) funktsiya – ag'zalar
4. Klassning konstanta ob'ektlari ham konstanta funktsiya – ag'zalar
5. Klass ob'ektlarining massivi

Klass sintaksisi: Klass tushinigi C++ tilinde eng a'hmiyetli tushiniklardan biri. Klass sintaksisi stuktura(*struct*) sintaksisiga uqas ha'm oning ko'rinisi to'mendegishe:

```
class <klass atı>
{
//klassning jabiq berilgen-ag'zaları ha'm funktsiya-
ag'zaları
public:
//klassning ashıq berilgen-ag'zaları ha'm funktsiya
ag'zaları
}
<ob'ektlar dizimi>
```

A'detde klass usunısında <ob'ektlar dizimi> bo'legi shart emes. Klass ob'ektlarining keyinshelik, zarur boyinsha dag'aza etiw mumkin. Shunday – aq <klass atı> bo'legi majbwriy bolmasada, oning bolganı maqul. Sebebi <klass atı> berilgenlarning jan'a turi bolıp, oning jarideminde usı klass ob'ektlari aniqlanadı.

Klass ishinde dag'aza etilgen funktsiya ha'm berilgen usı klass ag'zalarini esaplanadı. Klass ishinde dag'aza etilgen o'zgeriushiler berilgen – ag'zalar, klass ishinde dag'aza etilgen funktsiyalar funktsiya – ag'zalarini dep ataladı. Kelisim boyinsha klass ishindeki barlıq funktsiya ha'm o'zgeriushiler jabik esaplanadı, yagniy olardı tek g'ana usı klass ag'zalarini isletiwu mumkin.

Klasstın ashıq ag'zaların dag'aza etiw ushın public gılt so'zi ha'm " : " belgisinen paydalanıladı. Klass dag'azasındagı public so'zinen keyın dag'aza etilgen funktsiyalar ha'm o'zgeriushilerge klasstın' basqa ag'zaları ha'm programmanın' usı klasstın' qa'legen jerinen mu'ra'jet etiw mu'mkin.

Klass dag'azasına misal:

```
class Klass {  
    int a; //klasstın' jabıq elementi  
    public:  
    int ma'nis_a();  
    void a_ma'nisi(int san);  
}
```

Sunday – aq int ma'nis_a() ha'm void a_ma'nisi(int san) funktsiyaları Klass klası ishinde dag'aza etilgen bolsada, olar ele anıqlang'an joq. Funktsiyanı anıqlawda "ko'riw oblastına ruxsat beriw"("::") a'melinen paydalanıladı. Funktsiya – ag'zanı anıqlaudın' ulıwma ko'rinisi to'mendegishe:

```
<funktsiya tu'ri><klass atı >::< funktsiya atı>( <parametrler  
dizimi> ) {  
    // funktsiya denesi  
}
```

Joqarıda dag'aza etilgen Klass klasının' int ma'nis_a() ha'm void a_ma'nisi(int san) funktsiya – ag'zaların anıqlaug'a misal keltirilgen.

```
int Klass::ma'nis_a(){  
    return a;  
}  
void Klass::a_ma'nisi(int san){  
    a=san;  
}
```

Tek g'ana Klass klasın dag'aza etiw menen usı klass tu'rindegi ob'ektler ju'zege kelmeydi. Klass ob'ektlerin ju'zege keltiriw ushın klass atın berilgenler spetsifikatorı sıpatıda isletiw za'rwr boladı. Mısalı:

```
Klass obj1,obj2;
```

Klass ob'ekti jaratılǵ'annan keyin '.' ja'rdeminde klasstın' ashıq ag'zalarına mu'ra'jet etiu'ge boladı. Mısal ushın

```
obj1. a_ma'nisi(20);
```

obj2. a_ma'nisi(50); mu'ra'jetler arqalı ob1 ha'm ob2 ob'ektlerinin' a o'zgeriwshilerine ma'nisler beriledi. Ha'r bir ob'ekt klassta dag'aza etilgen o'zgeriwshilerinin' o'z kopyalarına iye boladı. Usı sebepli ob1 ob'ektindegi a o'zgeriwshi ob2 ob'ektindegi a o'zgeriwshiden parq qıladı.

Klass elementlerine mu'ra'jetdi ko'rsetkishler arqalı a'melge asırıw mu'mkin. To'mendegi programma bug'an mısál boladı.

```
class Toshka
{
public:
int x,y;
void Kord_manis_beriw(int _x, int _y);
};
void Toshka :: Kord_manis_beriw(int _x, int _y){x=_x; y=_y;}
int main()
{
Toshka XOY;
Toshka *Kord_korsetkish=&XOY;
// ...
XOY.x=0;
Kord_korsetkish->y=0;
Kord_korsetkish-> Kord_manis_beriw(10,15);
```

```
// ...  
return 0;  
}
```

Sonı aytıp o'tiw kerek klass ob'ektlerine '.' ha'm '->' arqalı mu'ra'jet etiwidin' kompilyatsiya ko'z – qarasınan hesh bir parqı joq. Kompilyator '.' menen mu'ra'jetdi '->' menen almasıradı. Misal:

```
obj1.a_manisi(20); ko'rsetpesi kompilyator ta'repinen
```

```
(&obj1)->a_manisi(20); ko'rinisindegi ko'rsetpe menen almasıradı.
```

C++ tili klasstın berilgen – ag'zalarına belgigi bir sheklewler qoyadı.

– berilgen – ag'zalar **auto**, **extern** yaki **register** modifikatorları menen anıqlanıwı mu'mkin emes.

– klasstın' berilgen – ag'zaları usı klass tu'rindegi ob'ekt bolıwı mu'mkin.

Belgili bir mazmung'a iye klassqa misal tiykarında stek qurılması menen a'mel etetug'ın klass jaratıp ko'reyik.

```
#include <iostream>  
using namespace std;  
#define OLSHEM 100  
class STEK  
{  
int stk[OLSHEM];  
int stk_ushi;  
public:  
void Baslaw();  
void Jaylastiriw(int san);  
bool aliw(int &san);  
};  
void STEK :: Baslaw()
```

```

{
stk_ushi=0;
}
void STEK :: Jaylastiriw(int san)
{
If(stk_ushi== OLSHEM)
{
cout<<"Stek toldi! \n";
return 0;
}
stk[stk_ushi]=san;
stk_ushi++;
};
bool STEK :: aliw(int &san)
{
If(stk_ushi==0)
{
cout<<"Stek bos! \n";
return false;
}
stk_ushi--;
san=stk[stk_ushi];
return true;
}
int main();
{
STEK stek1, stek2;

```

```

int san;
stek1. Baslaw();
stek2. Baslaw();
stek1. Jaylastiriw(1);
stek2. Jaylastiriw(2);
stek1. Jaylastiriw(3);
stek2. Jaylastiriw(4);
if(stek1.aliw(san)) cout<<san<<" ";
if(stek1.aliw(san)) cout<<san<<" ";
if(stek2.aliw(san)) cout<<san<<" ";
if(stek2.aliw(san)) cout<<san<<" ";
if(stek2.aliw(san)) cout<<san<<" ";
return 0;
}

```

Programma isletiliwi na'tiyjesinde ekrang'a to'mendegiler pechatqa shıg'adı.

3 1 4 2 Stek bos!

Klass, onın' ag'zaların paydalanıwdan aldın dag'aza etilgen bolıwı kerek. Biraq ayrım jag'daylarda klassta ele dag'aza etilmegen klassqa ko'rsetkish yaki adres alıw a'melin (&)dag'aza etiwde za'ru'r bolıwı mu'mkin. Bul jag'dayda klasstın' tolıq bolmag'an dagazasınan paydalanıwg'a tuwrı keledi ha'm ol to'mendegi ko'rinsike iye.

```
class<klass atı>;
```

Mısal ko'reyik.

```
class Klass2; //klasstın' tolıq emes dag'azası
```

```
class Klass1
```

```
{
```

```
int x;
```

```

Klass2*klass2; //Klass2 klasına ko'rsetkish
public:
Klass1(int _x){x=_x;}
};
int main()
{
//...
return 0;
}
class Klass2 //Klass2 klasının' toliq dag'azası
{
int a;
public:
Klass2();
};

```

Usını aytıp o'tiw kerek, klass dag'azası struktura (**struct**) dag'azasına uqsas, pariqli twrde:

- Class dag'azasında **public**, **protected** yaki **private** mu'ra'jet modifikatorları isletiledi;
- struct gilt so'zi ornında class yaki union gilt so'zleri isletiliwi mu'mkin;
- a'dette class quramında berilgenlerden tısqarı funktsiya – ag'zalar da kiredi;
- class konstruktor yaki destruktur dep atalıushı arnawlı funktsiya – ag'zalarına iye boladı.

To'mende struct ha'm union gilt so'zleri menen anıqlang'an klasslarga misal keltirilgen.

```

struct Toshka
{
private:

```

```

int x; int y;

public:

int Aliw_X();

int Aliw_Y();

void manis_beriw_X(int _x);

void manis_beriw_Y(int _y);

};

union Bit

{

Bit(unsigned int n);

void Bit_pechat();

unsigned int num;

unsigned char c[sizeof(unsigned int)];

};

```

Mu'ra'jet spetsifikatorı, odan keyin jaylasqan barlıq klass elementlerine qollanılıwı dawam etedi, egerde basqa spetsifikator ushıramag'ansha yaqi klass dag'azası tamam bolmag'ansha.

To'mende klassqa mu'ra'jet spetsifikatorlarına tu'sinikler keltirilgen:

private – berilgen – ag'zalarına ha'm funktsiya – ag'zalarına tek g'ana usı klass ishinde mu'ra'jet etiwı mu'mkin;

protected – berilgen – ag'zalarına ha'm funktsiya – ag'zalarına tek g'ana usı klass ha'm usı klasstan payda bolg'an klasslar funktsiya – ag'zalarına mu'ra'jet etiwı mu'mkin;

public - berilgen – ag'zalarına ha'm funktsiya – ag'zalarına tek g'ana usı klass funktsiya – ag'zaları ha'm usı klass ob'ekti bar bolg'an programma funktsiyaları mu'ra'jet etiwı mu'mkin.

C++ tilinde struktura ha'm birlespeler klass tu'rleri dep qaraladı. Struktura ha'm klasslar bir – birine uqsas, tek kelisim boyınsha mu'ra'jet spetsifikatorı menen parq qıladı: strukturada kelisim boyınsha barlıq elementler **public** mu'ra'jet spetsifikatorına iye bolsa, klassta olar **private** mu'ra'jet spetsifikatorında boladı.

Birlespede de, struktura sıyaqlı kelisim boyınsha elementler public mu'ra'jet spetsifikatorında boladı.

Dizim ha'm birlespeler öz konstruktor ha'm destruktorlarına iye bolıwı mu'mkin. Sol menen birge birlespelerdi klass sıpatında isletiwge ma'lim bir sheklewler bar. Birinshiden, olar qandayda bir klass miyrası bolıwı mu'mkin emes, ezleri de basqa klass ushın tayanış klass bola almaydı. Olar statik atributlı ag'zalarg'a ha'mde olar öz quramına konstruktor ham destruktorı bolg'an ob'ektlerge iye bolıwı mu'mkin emes.

Birlespede klass sıpatında isletiliwine misal koreyik.

```
#include<iostream>

using namespace std;

union Bit
{
    Bit (unsigned short int n);
    void Bit_ekranga _shigariw();
    unsigned short int num;
    unsigned char c[sizeof(unsigned short int)];
};

Bit::Bit(unsigned short int n) { num = n; };

void Bit::Bit_ekranga _shigariw()
{
    for(int j=0; j<sizeof(unsigned short int); j++)
    {
        cout<<j<<" – bayttın' ekilik ko'rinisi "<<" : ";
        for(int i=128; i; i>>=1){ if(i&c[j]) cout<<" 1 "; else
cout<<" 0 ";}

        cout<<endl;
    }
}
```

```

int main()
{
    Bit bit(2000);
    bit Bit_ekranga_shigariw();
    return 0;
}

```

Bul mäsaldä ob'ektke berilgen belgi (on' yamasa teris) siz pu'tin ma'nistin' ekilik k6rinisi ekrang'a shıg'arıladı. Berilgen belgi (on' yamasa teris) siz pu'tin 2000 sanı ushın t6mendegi qatar payda boladı.

0 - bayttın' ekilik ko'rinisi 11010000

1 - bayttın' ekilik ko'rinisi 00000111

Nusqalaw konstrukt6rı: Nusqalaw konstrukt6rı klass ob'ektin jaratıwda sol klastın' bar bolg'an ob'ektlerinen berilgenlerdi (olardin' ma'nisin) nusqasın aladı . Sol sebepli ol klass ob'ektine konstanta ob'ekt (const S&) yaki apiwayı adres alıw 6meli (S&) bolg'an jalg'iz parametrge iye boladı . Parametrlerdin' birinshisin isletiw maqul esaplanadı , ol konstanta ob'ektlerin nusqalaw imkanın' beredi .

T6mendegi mäsaldä nusqalaw konstrukt6rın isletiw k6rsetilgen .

```

#include <iostream>
using namespace std;
class Noqat
{
    int x,y;
public:
    Noqat( const Noqat & koor);
    Noqat( int x0 , int y0 );
    void Abtsissa();
    void Ordinata();
    void Ozgertiw(int delta_x , int delta_y );
};

```

```

Noqat::Noqat( const Noqat & koor ) { x = koor.x ; y = koor.y ; };
Noqat::Noqat( int x0 , int y0 ) { x = x0 ; y = y0 ; };
void Noqat::Abtsissa() { cout<<"x="<<x; };
void Noqat::Ordinata() { cout<<"y="<<y; };
void Noqat::Ozgertiw(int delta_x , int delta_y ) { x+= delta_x ; y+= delta_y ;
};

```

```
int main()
```

```
{
```

```
    Noqat koord1(5,10);
```

```
    Noqat koord2=koord1;
```

```
    Noqat koord3(koord1);
```

```
    koord1.Ozgertiw(3,-2);
```

```
    koord2.Ozgertiw(1,2);
```

```
    cout<<"\n koord1 ob'ekt berilgen ag'zalar ma'nisi:";
```

```
    koord1.Abtsissa();
```

```
    koord1.Ordinata();
```

```
    cout<<"\n koord2 ob'ekt berilgen ag'zalar ma'nisi:";
```

```
    koord2.Abtsissa();
```

```
    koord2.Ordinata();
```

```
    cout<<"\n koord3 ob'ekt berilgen ag'zalar ma'nisi:";
```

```
    koord3.Abtsissa();
```

```
    koord3.Ordinata();
```

```
    return 0;
```

```
}    programma islew natiyjesinde ekrang'a tmendegiler shıg'adı:
```

```
koord1 ob'ekt berilgen ag'zalar ma'nisi: x = 8 y = 8
```

```
    koord1 ob'ekt berilgen ag'zalar ma'nisi: x = 6 y = 12
```

```
    koord1 ob'ekt berilgen ag'zalar ma'nisi: x = 5 y =10
```



```

public:
    Manis_x() { return x ; }
    Manis_y() { return y; }
    void x_Manisi( int x ) ;
    void y_Manisi( int y ) ;
};

```

```

inline void Noqat:: x_Manisi( int _x ) { x = _x; }

```

```

inline void Noqat:: y_Manisi( int _y ) { y = _y; }

```

Klasstın’ konstanta ob’ektləri ham konstanta funksiya - ag’zaları: Klasstın’ funksiya - ag’zaları parametrlər diziminen keyin keliwshi const modifikatorı menen dag’azalanıwı mu’mkin . Bunday funksiya klass berilgen - ag’zalar ma’nislerin özgerite almaydı ha’m klasstın’ konstanta bolmag’an funksiya - ag’zaların shaqırıw mu’mkin emes . Konstanta funksiya - ag’zası bolg’an klassqa mısıl keltiremiz .

```

class Noqat
{
int x,y;
public:
    Noqat ( int _x , int y );
    void Manis_Beriw( int x0 , int y0 );
    void Manis_Aliw( int xx , int yy ); //Konstanta funksiya - ag’zasinin’
dag’azası
};
Noqat::tNoqat ( int _x , int y ) { x=_x ; y=_y ; }
void Noqat::Manis_Beriw( int x0 , int y0 ) { x = x0 ; y = y0 ; }
void Noqat::Manis_Aliw( int &xx , int &yy ) const { xx = x ; yy = y; }

```

Tap sonday , konstanta ob’ektlerin jaratıw mu’mkin . Bunın’ ushın ob’ekt dag’azası aldına const modifikatorın qoyıw jeterli . Mısıl :

```
const Noqat koord( 3 , 6 );
```

til kersetpesindeki const gilt sözi kompilyatorg'a usı ob'ektin' jagdayı özgermewi kerekligin bildiredi . Sol sebepli ob'ekt berilgen - ag'zaları ma'nisin özgeretug'un funksiya - ag'zanı shaqırıwı ushrasa , kompilyator qätelik haqqında xabar beredi . Bul qag'ıydaq'a konstanta funksiya - ag'zalardı shaqırıw boysinbaydı , sebebi öz mazmunın'a kere olar berilgen - ag'zalardıń ma'nisin özgerete almaydı. Joqarıda dag'azalang'an Noqat klası konstanta ob'ekt isletiliwine misal .

```
// Bul jerde Noqat klassı dag'azası ham anıqlanıwı jazıladı
```

```
int main(){  
    Noqat noqat1(3,7);  
    const Noqat noqat2(8,10); //Konstanta ob'ekt  
    int a,b;  
    noqat1.Manis-Aliw( a , b );  
    noqat2.Manis_Beriw(2,3); //qöte  
    noqat2.Manis_Aliw(a,b); //Duris  
    return 0;  
}
```

Konstanta funksiya - ag'zalardıń berilgen - ag'zalar menen islew menen baylanıslı sheklewlerdi " aylanıp ötiw " ushın mutable gilt sözi anıqlang'an . Bul gilt söz klastın' qaysı berilgen - ag'za konstanta funksiya - ag'zalar tarepinen özgeriliwi mu'mkin ekenin kersetedi . Statik ham konstanta berilgen - ag'zalarına mutable gilt sözın isletiw mu'mkin emes , ol berilgenler turinin' modifikatorı sıpatında isletiledi .

Mısal .

```
#include<iostream>  
using namespace std;  
class Klas{  
    mutable int count ;  
    mutable const int *intPtr; //Əmel orınlı , kersetkish bolsa konstanta
```

```
// putin sang'a k rsetse de  zi konstanta emes
```

```
public:
```

```
    int Funktsiya(int i=0) const{
```

```
        count = i++;
```

```
        intPtr = &i;
```

```
        cout<<*&i;
```

```
        return count;
```

```
    }
```

```
};
```

```
int main()
```

```
{
```

```
    Klas S;
```

```
    S.Funktsiya();
```

```
    return 0;
```

```
}
```

Programma islewi n tiyjesinde ekrang'a 1 shig'adı .

Klass ob'ektlarinin' massivi: Klass ob'ektlarinin' massivler qurıw mu'mkin. Ob'ektlar massivni dag'azalaw sintaksisi apiwayı  zgeriwshiler massivni dag'azalawga uqsas . Mısalı , 10 dana S klass ob'ektinen ibarat massiv dag'azası t mendegishe boladı :

```
S obekt_massiv[10];
```

Kompilyator ta'repinen bul massivti jaratıwda klass ob'ektlarinin' kelisimi boyınsha konstruktordan paydalanıw z ru'r . Sol sebepli kompilyator  zi klass ushın konstruktor jaratıwg'a suyenip qalmay , klass ob'ektlarinin' massivni dag'azalawda klass quramına kelisim boyınsha konstruktor kiritiw gerek . Eger paydalanıwshı t repinen konstruktor anıqlanatuǵun bolsa , klass ob'ektlari massivini' har bir elementini' baslang'ısh manisi ayqın tu'rde k rsetiliwi gerek .

Ob'ektlar massivine mu'ra'jet tap q legen tu'rdegi elementlerden ibarat massivke mu'ra'jet etkende y boladı :

```
#include<iostream >
```

```

using namespace std;

class Klas{
int a;
public:
    Klas(int n) { a = n ;}
    int a_Manisi() { return a; }
};

int main(){
Klas ob_Klas[5]={ 12,23,34,56,67};
for( int i=0 ; i<4 ; i++ ){
cout<<ob_Klas[i].a_Manisi()<<" //Massiv elementlerine mura'jet
}
cout<<"\n";
return 0;
}

```

Programma ekrang'a Klass klası har ob'ekti berilgen - ag'zasının' initsializatsiyada qalg'an ma'nisi shig'adı . Initsializatsiyanın' bul usılınan konstruktör bir parametрге iye bolg'an halda isletiw mu'mkin . Egerde konstruktör birden artıq parametрге iye bolsa initsializatsiyanın' tolıq formasınan tuwrı keledi .

```

#include<iostream>

using namespace std;

class Koord
{
int x,y;
public:
Koord(int _x, int _y) { x=_x; y=_y; }
int x_Manisi() { return x; }
int y_Manisi() { return y; }
}

```

```

};

int main(){
    Koord koord_mas[4][2]={ Koord(1,2);Koord(3,4);Koord(5,6);Koord(7,8);
    Koord(9,10);Koord(11,12);Koord(13,14);Koord(15,16);}
    int i,j;
    for( i=0 ; i<4 ; i++ )
    for( j=0 ; j<2 ; j++ ){
        cout<<koord_mas[i][j].x_Manisi()<<" ";
        cout<<koord_mas[i][j].y_Manisi()<<"\n";
    }
    return 0;
}

```

Ushbu programma klass ob'ektlar massivining ha'rbir elementining x,u berilgan - ag'zalarining ma'nisini ekranga shug'aradi.

Klass ob'ektlari massivke ko'rsatkich orqali mu'ra'jet etiw mu'mkin . Bunda ko'rsatkichlar arifmetikasi da orinli boladi . Misali , ko'rsatkich inkrementi massivining keyingi elementine ko'rsatkichine alip kelse , ko'rsatkich dekrementi - ko'rsatkich o'zi ko'rsatip turg'an massiv ob'ektinen alding'1 ob'ektke ko'rsatedi . Joqarida keltirilgen programmaning ko'rsatkich isletilgen variantin ko'reyik .

```

#include<iostream>
using namespace std;
class Koord
{
    int x,y;
public:
    Koord(int _x, int _y) { x=_x; y=_y; }
    int x_Manisi() { return x; }
    int y_Manisi() { return y; }
};

```

```

int main(){
Koord koord_mas[4]={ Koord(1,2);Koord(3,4);Koord(5,6);Koord(7,8);}
int i;
Koord *korset_ob;
korset_ob=koord_mas;
for( i=0 ; i<4 ; i++ ){
cout<<korset_ob->x_Manisi()<<" ";
cout<<korset_ob->y_Manisi()<<"\n";
korset_ob++;
}
return 0;
}

```

Programma islew na'tiyjesinde ekrang'a

1 2

3 4

5 6

7 8

manisleri shig'adı .

Klass ob'ektler massivi dinamik yadta payda bolıw mu'mkin :

```

#include<iostream>
using namespace std;
class Koord
{
int x,y;
public:
Koord(int _x, int _y) { x=_x; y=_y; }
Koord() { x=0; y=0; }
}

```

```

int x_Manisi() { return x; }
int y_Manisi() { return y; }
};
int main(){
int i;
Koord *korset_ob;
korset_ob=new Koord[4];
for( i=0 ; i<4 ; i++ ){
cout<<korset_ob->x_Manisi()<<";
cout<<korset_ob->y_Manisi()<<"\n";
korset_ob++;
}
delete []korset_ob;
return 0;
}

```

Programma Koord klasinin' ob'eklerinin' dinamik massivi payda qiliniwında paydalanıwshı ta'repinen kelisim boyınsha anıqlang'an parametrsiz konstruktor a'mel qıdadı ha'm programma islewi na'tiyjesinde ekrang'a

0 0

0 0

0 0

0 0

ma'nisleri shıg'adı.

The General Form of a class

All classes are declared in a fashion similar to the **queue** class just described. The general form of a **class** declaration is shown here.

```
class class-name {  
    private data and functions  
public:  
    public data and functions  
} object-list;
```

Here *class-name* specifies the name of the class. This name becomes a new type name that can be used to create objects of the class. You can also create objects of the class by specifying them immediately after the class declaration in *object-list*, but this is optional. Once a class has been declared, objects can be created where needed.

Class Fundamentals

Let's begin by defining the terms class and object. A *class* defines a new data type that specifies the form of an *object*. A class includes both data and the code that will operate on that data. Thus, a class *links data with code*. C++ uses a class specification to construct objects. Objects are *instances* of a class. Therefore, a class is essentially a set of plans that specify how to build an object. It is important to be clear on one issue: A class is a logical abstraction. It is not until an object of that class has been created that a physical representation of that class exists in memory.

When you define a class, you declare the data that it contains and the code that operates on that data. Although very simple classes might contain only code or only data, most real-world classes contain both. Within a class, data is contained in variables and code is contained in functions. Collectively, the functions and variables that constitute a class are called *members* of the class. Thus, a variable declared within a class is called a *member variable*, and a function declared within a class is called a *member function*. Sometimes the term *instance variable* is used in place of member variable.

29-Leksiya. Klass konstruktorlari ha'm destruktorelari

Joba:

6. Konstruktor ham destruktorelar
7. `this` ko'rsetkishi
8. Klasi'n' statik ag'zaları
9. Klas funktsiya - ag'zalarına ko'rsetkishlerdi isletiw

Konstruktor ham destruktorelar: Adette klas obiekti jaratıwda onı initsializatsiyalaw kerek. Bul maxsette C++ tilinde konstruktor dep atalıwshı arnawlı funktsiya - ag'za anıqlang'an. Klas konstruktorı har dayım klas obiekti jaratılıwı waqtında avtomatik rawishte shaqırıladı. Konstruktor atı o'zi ag'za bolg'an klas atı menen ustpe - ust tusedi ham qaytarıwshı maniske iye bolmaydı. Misalı,

```
#include<iostream>
using namespace std;
class Klas
{
    int var;
public:
    Klas();// Konstruktor
    void Ekranga_shigariw_var();
};
Klas::Klas()
{
    cout<<"Konstruktor isledi!\n";
    var=0;
}
void Klas::Ekranga_shigariw_var() { cout<<var; }
int main()
{
    Klas ob;
    ob.Ekranga_shigariw_var();
    //...
    return 0;
}
```

```
}
```

Bul misalda Klas konstruktori ekranga xabar shig'aradi ham jabiq *var* o'zgeriwshini initsializatsiya qiladi (baslangish manis beriledi). Soni da aytip o'tiw kerek , programma duziwshisi ayqin rawishte klas konstruktorin shaqiratug'un til ko'rsetpesin jazbawı kerek barlıq zərur jumıstı kompilyator əmelge asıradı. Joqarıda aytilganday konstruktor ol tiyisli bolg'an klas obiekti jaratılıp atırg'an waqıtta avtomatik rawishte shaqırıladi. O'z nawbetinde objekt onı dag'azalawshı operator orınlanıwında jaratıladi. Sonın' ushında C++ tilinde o'zgeriwshi dag'azası orınlanıwshı operator esaplanadı .

Global obiekter ushın konstruktor programma orınlanıwı baslang'anda shaqırıladi .

Konstruktorg'a qarata kerı əmel orınlaytug'un funksiya - ag'zalarg'a *destruktorlar* delinedi . Bul funksiya – ag'za objekt o'shiriliwinde (jog'altılıwında) avtomatik rawishte shaqırıladi . Adette destruktor objekt tərepinen iyelengen diynamik yadtti bosatiw ushın qızmet qiladi . Onın' atı klas atı menen saykes tusedi , tek aldına '~' belgisi qoyıladi.

To'mende destruktor anıqlang'an klasqa misal keltirilgen .

```
#include<iostream>
using namespace std;
class Klas
{
    int var;
    public:
        Klas();// Konstruktor
        ~Klas();// destruktor
        void Ekranga_shigariw_var();
};
Klas::Klas() { cout<<"Konstruktor isledi!\n"; var=0; }
Klas::~Klas() { cout<<"destruktor isledi!\n"; }
void Klas::Ekranga_shigariw_var() { cout<<var<<endl; }
int main()
{
```

```

    Klas ob;
    ob.Ekranga_shigariw_var();
    //...
    return 0;
}

```

Destruktor obiekt o'shiriliwinde shaqiriladi . Global obiektler programma tamamlanivinda o'shiriledi . Lokal obiektler - olardin' ko'riniw oblastinan shig'iwda o'shiriledi.

Soni aniq etip aytip o'tiw kerek , konstruktor ham destruktorga ko'rsetkishler payda qiliw mumkin emes .

Eger klas o'zgeriwshilerin initsializatsiya qiliw zarur bolsa , parametrli konstruktor isletiledi . Joqarida keltirilgen misalg'a o'zgertiw kiritemiz.

```

#include<iostream>
using namespace std;
class Klas
{
    int a,b;
    public:
        Klas(int x, int y);
        ~Klas();
        void Ekranga_shigariw_var();
};
Klas::Klas(int x, int y) { cout<<"Konstruktor isledi!\n"; a=x; b=y; }
Klas::~Klas() { cout<<"Destruktor isledi!\n"; }
void Klas::Ekranga_shigariw_var() { cout<<a<<b<<endl; }
int main()
{
    Klas ob(5,10);
    ob.Ekranga_shigariw_var();
    //...
    return 0;
}

```

Bul jerde ob obiekti dag'azasında konstruktorg'a uzatılǵ'an manisler klas quramındag'ı a ham b jabıq o'zgeriwshilerin initsializatsiya qılıwda isletiledi .

Parametrli konstruktorg'a manis uzatıwda << Klas ob(5,10); >> ham<< Klas(5,10);>> ko'rsetpeleri bir qıylı mazmung'a iye . Ko'pshilik jag'daylarda ekinshi , qısqa forma isletiledi.

Konstruktordan parqlı rawishte destruktur parametrge iye bolıwı mumkin emes , sebebi o'shirilip atırǵ'an obiekt o'zgeriwshilerge manis beriw ahmiyetke iye emes .

Konstruktor ushın anıqlang'an bir - neshe qag'ıydalardı keltiremiz :

- konstruktor funksiyası ushın qaytarılıwshı mənis turi ko'rsetilmeydi;
- konstruktor mənis qaytarmaydı;
- konstruktor miyraslıq penen o'tpeydi;
- konstruktor *const* , *volatile* , *static* yaki *virtual* modifikatorları menen dag'azalanbaydı.

Eger klas anıqlanıwında konstruktor dag'azalanbasa , kompilyator o'zi kelisiw boyınsha parametrsiz konstruktordı jaratadı .

Destruktor ushın to'mendegi qag'ıydalar anıqlang'an :

- destruktur funksiyası parametrlerde bolıwı mumkin emes;
- destruktur manis qaytarmaydı;
- destruktur miyraslıq penen o'tpeydi;
- klas birden artıq konstruktorg'a iye bolıwı mumkin emes;
- destruktur *const* , *volatile* , *static* yaki *virtual* modifikatorları menen dag'azalanbaydı.

Eger klasta destruktur dag'azalanbasa , kompilyator o'zi kelisiw boyınsha destruktordı payda etedi .

Adette klas berilgen - ag'zalar konstruktor denesinde initsializatsiyalanadı.

Biraq initsializatsiyanı basqa usıl menen - elementlerdi initsializatsiyalaw dizimi arqalı əmelge asırıw mumkin . Elementlerdi initsializatsiyalaw dizimi funksiya denesi anıqlangannan keyin eki noqattan (':') keyin jaylasadı ham onda utir menen ajratılǵ'an halda berilgen - ag'zalar ham tayansh klas jazıladı. Har bir element ushın qawsırma ishinde

initsializatsiyada isletiletug'un bir yaki bir neshe parametrlar ko'rsetiledi .
To'mendegi misalda elementlerdi initsializatsiyalaw dizimi arqali klas
o'zgeriwshilerga baslang'ish manis beriw ko'rsetilgen .

```
#include<iostream>
using namespace std;
class Klas
{
    int a,b;
    public:
        Klas( int x , int y);
};
Klas::Klas( int x , int y ):a(x),b(y)
{
    cout<<"Konstruktor isledi\n";
}
//...
int main()
{
    Klas(5,10);
    return 0;
}
```

Keltirilgen programma bo'leginde konstruktor orinlaytug'un jumis
alding'ı misaldag'ı konstruktor jumisi menen bir qiyli mazmunga iye .

Albette , klas elementlerin initsializatsiya qilıwdın' qaysi formasın qollaw
programma duziwshige baylanisli . Biraq , sonday jagdaylar boladı , onda
elementlerdi initsializatsiyalaw diziminen paydalanbawdın' ilaji joq:

- klasın' konstanta berilgenlerine ham adres alıw əmeli qollanılğ'an
o'zgeriwshilerga baslang'ish manis beriwde;
- klas ag'zası basqa klas obiekti bolg'anda ham bul obiekt konstruktorı bir
yaki bir neshe parametrlarğa manis beriwdi talap qilgan hallarda.

Kelisiw boyınsha konstruktorlardı isletiwde kolliziyadan (bir nərseni eki
turli tusiniwden) qashıw kerek , yag'nıy klasta bir neshe konstruktor

bolg'anda komplyator olardin' qaysı birin qashan shaqırılıwın anıq biliwi kerek.

To'mendegi mısalda bul jagday menen baylanıslı qäte ko'rsetilgen.

```
class S
{
    public:
        S(); //Kelisiw boyınsha konstruktor
        S(int i=0); //Kelisiw boyınsha konstruktor ornın'a isletiletugun
konstruktor
};
int main()
{
    S ob1(10); //S::S(int) konstruktorı isletiledi.
    S ob2 ; //Natuwri S::S(int) yaki S::S() konstruktorlarının' qaysı biri?
    //...
    return 0;
}
```

Bul qarama qarsılıqtı sheshiw jolı - usı klas dag' azasınan kelisiw boyınsha konstruktordı o'shiriwden ibarat .

this ko'rsetkishi: C++ tilindegi har bir obiekt kompilyator tärepinen jaratılatur'un ham tap sol obiektke ko'rsetiwshi **this** dep atalıwshı arnawlı ko'rsetkishke iye . **this** ko'rsetkishinin' turi " S " ko'rinisinde bolıp , S - usı obiekt klasının' turi esaplanadı . **this** ko'rsetkishi klasta anıqlang'anlıg'ı sebepli onın' әmel qılıw oblastı o'zi anıqlang'an klas boladı . Basqasha aytqanda , **this** ko'rsetkishin kompilyator tärepinen qosılıwshı klastın' jasırınğ'an parametri dep qaraw mumkin . Klas funktsiya - ag'zası shaqırılğ'anda og'an **this** ko'rsetkishi , tap birinshi argument sıpatında uzatıladı , yag'nıy funktsiya - ag'zanı shaqırıwdın' to'mendegi ko'rinisi

```
Obiekt1.Funktsiya( arg1 , arg2);
```

komplyator tarepinen

```
Obiekt1.Funktsiya( & Obiekt1 , arg1 , arg2);
```

ko'rinisinde beriledi.

Funktsiya shaqirilg'anda onin' qawsirma ishindegi argumentleri stekke on'nan shep tarepke qarap jaylastiriladi . Birinshi argument (this) stekke en' aqirinda jaylastiriladi . Funktsiya - ag'za ishinde obiektler adresi this arqali aniqlanadi .To'mende this ko'rsetkishin isletiw menen baylanisli programma teksti keltirilgen .

```
#include<iostream>
#include<string.h>
using namespace std;
class S
{
    char At[20];
    public:
        S(char*);
        void Salem();
};
S::S(char *_At)
{
    strcpy(At,_At);
    Salem(); //Ush murajat
    this->Salem(); //oz - ara
    (*this).Salem(); //ekvivalent
}
void S::Salem()
{
    cout<<"Salem"<<At<<'\\n';
    cout<<"Salem"<<this->At<<'\\n';
}
int main()
{
    S obiekt("Muqumov Rustam");
    return 0;
}
```

Programma tekstinen ko'rinip turıptı , funksiya - ag'za ishine basqa funksiya - ag'zalarg'a h m berilgen - ag'zalarg'a **bevosita** olardıń atları menen yaki this ko'rsetkishi menen mura'jet etiw mumkin . Sol sebepli  meliyde this ko'rsetkishinen kem paydalanıladı , ol tiykarınan funksiya qaytarıwshı manisi sıpatında (return this; yaki return *this;) h m operatorlardı qayta juklew menen baylanıslı maselelerde ken' qollanıladı.

Klastın' statik ag'zaları: Klas ag'zaları static modifikatorı menen dag'azalaw mumkin . Klas statik ag'zasın klas oblasti shegarasında mura'jet qılıw mumkin bolg'an global o'zgeriwshi yaki funksiya dep qaraw mumkin . Klastın' static dep dag'azalang'an berilgen - ag'zaları klastın' barlıq obiektleri tarepinen birgelikte isletiledi, sebebi bunday o'zgeriwshi jalg'ız nusqada boladı .  melde klastın' statik berilgenleri ushın yadtan orın ,h tteki klastın' birar obiekti dag'azalanbag'an bolsa da ajratıladı . Sol sebepli klas statik berilgeni dag'azalap qalmastan , onı anıqlaw sh rt . Mısalı :

```
class Klas
{
public:
    Klas();
    static int Sanagish; //statik berilgen - ag'za dag'azası
};
```

Bul mısalda Sanagish statik berilgen - ag'zalarga klas atı arqalı mura'jet qılğ'an maqul boladı.

```
Klas::Sanagish++;
```

Bul halat Sanagish++ statik berilgen - ag'za barlıq klas obiektleri ushın jalg'ız ekenligin an'latadı .

Egerde statik berilgenler jabıq dep dag'azalang'an bolsa , olarg'a funksiya - ag'zalar arqalı mura'jet qılıw mumkin .

Ulıwma alg'anda , statik berilgen - ag'zalardı isletiwde to'mendegi esletpelerdi beriw mumkin :

- statik berilgen - ag'zalardı bir neshe klas obiektleri tarepinen birgelikte isletiw ushın anıqlaw kerek ;

- statik berilgen - ag'zalardı *private* , *protected* modifikatorlar menen dag'aza qılıw arqalı olarg'a mura'jetti sheklew kerek .

Klastın' statik berilgen - ag'zanı isletiwge mısál .

```
#include<iostream>
using namespace std;
class S
{
    public:
        S() { ob_sani++; }
        ~S() { ob_sani--; }
        static int ob_sani;
    private:
        int x;
};
int S::ob_sani=0;
int main(){
    S * p_ob = new S[5];
    cout<<" Klastın "<<S::ob_sani<<" obiekti bar!\n ";
    delete []p_ob;
    return 0;
}
```

Programma islew natijesinde ekranda to'mendegi qatar payda boladı Klastın 5 obiekti bar. Klastın' statik funktsiyaları isletiliwinin' o'zgesheligi sonda , olar da jalg'ız nusqada anıqlanadı ham birar klas obiektinin' " jeke " funktsiyası bolmaydı . Sol sebepli , bul funktsiyalarg'a this ko'rsetkishi uzatılmaydı . Statik funktsiyalardıń bunday ayirmashılıg'ınan Windows OC ushın programmalaştırıwda ken' paydalanıladı .

Joqarıda aytlıg'an pikirlerden bir neshe tiykarg'ı natijeler kelip shıg'adı:

- statik funktsiya - ag'zalardı klastın' qandayda bir wəkili (obiekti) bar bolmasa da shıg'arıw mumkin ;

- klastin' statik funksiyası tek klastin' statik berilgenleri qayta islewi mumkin bolg'an ham ol tek funksiyanın' statik funksiya - ag'zaların shaqırırwı mumkin ;

- statik funksiya - ag'za virtual modifikatorı menen dag'aza qılıw mumkin emes .

To'mendegi keltirilgen programma statik funksiya - ag'zanı isletiwge mısal boladı :

```
#include<iostream>
using namespace std;
class S
{
    public:
        S() { sanagish++; }
        ~S() { sanagish--;}
        // ...
        static int Klas_Sanagish() { return sanagish; }
    private:
        int x;
        static int sanagish;
};
int S::sanagish=0;
int main()
{
    S * pOb = new S[10];
    cout<<" S klastin "<<S::Klas_Sanagish()<<" obiekti bar "<<endl;
    delete []pOb;
    return 0;
}
```

Klas funksiya - ag'zalarına ko'rsetkishlerdi isletiw: Klas funksiya - ag'zasına ko'rsetkishti anıqlaw mumkin . Onın' sintaksi to'mendegishe :

<manis turi> (<klas ati>::*<korsetkish ati>) (<parametrler>);

Mısal.

```
void ( S::F_korsetkish ) ( int x ; int y ; );
```

To'mendegi keltirilgen programmada funktsiya - ag'zasina ko'rsetkish isletilgen .

```
class S
{
    int i;
    public:
    S(int l):i(l) {};
    void FunAgza(){ cout<<" Funktsiya - agza isledi " ;}
    void FunShaqiriw(void (S::*F_Korsetkish)()) { this->*F_korsetkish(); }
}
int main()
{
    void (S::*F_Korsetkish)()=&S::FunAgza;
    S obiekt(100);
    obiekt.FunShaqiriw(F_Korsetkish);
    return 0;
}
```

Programmanın' 1-qatarında FunKorsetkish ko'rsetkishine FunAgza funktsiya - ag'za adresi berilgen . 3-qatarda FunShaqiriw funktsiya - ag'za FunKorsetkish ko'rsetkish boyinsha FunAgza funktsiyasın shaqıradı ham ekrang'a

Funktsiya - agza isledi
xabari shigadi.

Klastın' funktsiya - ag'zag'a ko'rsetkishin isletiwde məlim bir sheklewler bar :

- klastın' funktsiya – ag'zag'a ko'rsetkishin klastın' statik funktsiya - ag'zalarına ko'rsete almaydi , sebebi olarg'a this ko'rsetkishi uzatılmaydı ;
- klastın' funktsiya- ag'zag'a ko'rsetkishin qandayda bir klastın' ag'zası bolmag'an apiwayı funktsiya ko'rsetkishine aylandırıw mumkin emes .

Constructors and Destructors

It is very common for some part of an object to require initialization before it can be used. For example, consider the **queue** class, shown earlier in this chapter. Before the queue could be used, the variables **rloc** and **sloc** had to be set to zero. This was performed using the function **init()**. Because the requirement for initialization is so common, C++ allows objects to initialize themselves when they are created. This automatic initialization is performed through the use of a *constructor*.

A constructor is a special function that is a member of a class and that has the same name as the class. For example, here is how the **queue** class looks when it is converted to use a constructor for initialization:

An object's constructor is called when the object is created. This means that it is called when the object's declaration is executed. For global objects, the constructor is called when the program begins execution, prior to the call to **main()**. For local objects, the constructor is called each time the object declaration is encountered.

The complement of the constructor is the *destructor*. In many circumstances, an object will need to perform some action or series of actions when it is destroyed. Local objects are created when their block is entered, and destroyed when the block is left. Global objects are destroyed when the program terminates. There are many reasons why a destructor may be needed. For example, an object may need to deallocate memory that it had previously allocated. In C++, it is the destructor that handles deactivation. The destructor has the same name as the constructor, but the destructor's name is preceded by a **~**. Like constructors, destructors do not have return types.

Parameterized Constructors

A constructor can have parameters. This allows you to give member variables program-defined initial values when an object is created. You do this by passing arguments to an object's constructor. The next example will enhance the **queue** class to accept an argument that will act as the queue's ID number. First, **queue** is changed to look like this:

TIP: Unlike constructors, destructors cannot have parameters. The reason for this is easy to understand: There is no means by which to pass arguments to an object that is being destroyed. Although the situation is rare, if your object needs access to some run-time-defined data when its destructor is called, you will need to create a specific variable for this purpose. Then, just prior to the object's destruction, set that variable to the desired value.

The **this** Keyword

Each time a member function is invoked, it is automatically passed a pointer, called **this**, to the object on which it is called. The **this** pointer is an *implicit* parameter to all member functions. Therefore, inside a member function, **this** may be used to refer to the invoking object.

30-Leksiya. Dos funksiyalar ha'm dos klasslar

Joba:

1. Dos funksiyalar

2. Dos klasslar

Dos funksiyalar:

C++ tili OBP tiykarg'ı konsepsiyalarından biri – berilgenlerin Inkapsulyatsiyalaw konsepsiyasın «doslar» ja'rdeminde «buzıw» mu'mkin. C++ tilinde klasstın' eki tu'rdegi klass dosların dag'aza etiw imkanın beredi: dos funksiya ha'm dos klass.

Bizge belgili, klasstın' jabıq elementlerne funksiya-ag'zalar arqalı mu'rajat etiw mu'mkin. Biraq C++ tili klasstın' elementlerine basqa jol arqalı- dos funksiyalar arqalı mu'rajat qılıwdı qollap-quwatlaydı. Dos funksiyalar klass ag'zaları bolmag'an jag'dayda usı klasstın' jabıq ag'zalarına mu'rajat etiw imkaniyatına iye.

Bazı bir klassqa dos klasstı dag'aza qılıw ushın usı klass quramında aldına *friend* gilt so'zi qoyılǵ'an funksiya prototipi jazıladı.

Mısal ushın:

```
#include <iostream.h>

class Dos_klass
{
    int x,y;
    public:
        Dos_klass(int n, int m)
            {x=n; y=m;}
}
```

```

friend bool Eseli(Dos_klass ob); // Dos funkciya prototipi
};

bool Eseli(Dos_klass ob)
{
    return (!(ob.x%ob.y));
}

main()
{
    Dos_klass obiekt(12,3);
    if(Eseli(obiekt))
        cout<<"Sanlar eseli.\n";
    else
        cout<<"Sanlar o'z -ara eseli emes.\n";
    return 0;
}

```

Sonı aytıp o'tiw kerek, dos funktsiya o'zi dag'aza qiling'an klasstın' ag'zası esaplanbaydı. Usı sebepli, dos funktsiya shaqırıwda obiekt atın ko'rseti yaki klass ag'zalarına mu'rajat a'mellerin ("," yaki "->") isletilmeydi. Dos funktsiyalar klasstın' jabıq ag'zalarına tek klass obiekti arqalı mu'rajat etedi. Bul obiekt o'z na'wbetinde dos funktsiyada dag'aza qiling'an bolıwı yaki og'an argument sıpatında beriliwi mu'mkin. Dos funktsiya (klass) miyraslıq penen o'tpeydi, yag'nıy olar tuwındı klasslarda a'mel qılınbaydı. Basqa ta'repten, funktsiya bir waqıttın' o'zinde bir neshe klassqa dos bolıwı mu'mkin.

Mısalı:

```
#include <iostream>
```

```

using namespace std;
class Klass1;
class Klass2
{
    int d;
public:
    Klass2(int _d){ d=_d;}
    friend bool Eseli_sanlar(Klass1 ob1, Klass2 ob2);
};
class Klass1
{
    int n;
public:
    Klass1(int _n){n=_n; }
    friend bool Eseli_sanlar (Klass1 ob1, Klass2 ob2);
};
bool Eseli_sanlar (Klass1 ob1, Klass2 ob2)
{
    return(!(ob1.n%ob2.d));
}
int main()
{
    Klass1 obiekt1(15);
    Klass2 obiekt2(3);
    cout<<"Sanlar eseli";
    if(!Eseli_sanlar (obeikt1, obeikt2)) cout<<"emes";
}

```

```

        cout<<'!';
        return 0;
    }

```

Bul mısalda klasstın' toliq emes dag'azasınan paydalanıw ko'rsetilgen. Bul qurılmanı paydalanbay ele dag'azası berilmegen klass atın paydalanıp bolmaydı.

Funktsiya bir klasstın' ag'zası bolg'an halda basqa klasstın' dostı bolıwı mu'mkin.

Mısalı:

```

#include <iostream>
using namespace std;
class Klass1; //Klasstın toliq emes dag'azası
class Klass2
{
    int d;
    public:
        Klass2(int _d){ d=_d;}
        bool Eseli_Sanlar(Klass1 ob1);
};
class Klass1
{
    int n;
    public:
        Klass1(int _n){n=_n; }
        friend bool Klass2::Eseli_Sanlar(Klass1 ob1);//Dos funktsiy
protatipi

```

```

};

bool Klass2::Eseli_Sanlar(Klass1 ob1)
{
    return(!(ob1.n%d));
}

int main()
{
    Klass1 obiekt1(15);
    Klass2 obiekt2(3);

    if(obiekt2.Eseli_Sanlar(obiekt1)) cout<<"sanlar eseli";
    else cout<<"Sanlar eseli emes!";
    return 0;
}

```

Dos klasslar:

C++ tilinde tek g'ana dos funktsiyalardı emes, al dos klasslardı da dag'aza qılıw mu'mkin. Dos klassqa onı dos dep dag'aza qılǵ'an klasstın' barlıq ag'zalarına mu'rajat qılıwg'a tolıq ruxsat beredi. Dos klasstı dag'aza qılıw ushın klass dag'azasına dos klasstı *friend* gilt so'zi menen kiritiw jeterili.

Mısalı:

```

Class Klass
{
//...

friend class Dos_klass;

```

```
};
```

Bul jerde Klass klassına dos Dos_klass klassı dag'aza qiling'an.

Klass o'zin basqa bir ekinshi klasstın' dostı dep dag'aza qıla almaydı.

Onın' ushın ol ekinshi klassta dos dep dag'aza qiling'an bolıwı kerek.

Mısalı:

```
#include <iostream>
using namespace std;
class Klass1
{
    friend class Klass2;
    int x;
    void IncX(){x++;}
public:
    Klass1(){ x=0;}
    Klass1(int _x){ x=_x;}
};
class Klass2
{
    Klass1 klass1;
public:
    void x_manis();
};
void Klass2::x_manis(){
    cout<<klass1.x<<endl;
    klass1.IncX();
    cout<<klass1.x<<endl;
```

```

    }
    int main()
    {
        Klass2 klass2;
        klass2.x_manis();
        return 0;
    }

```

Klasslar bir-birin dos dep dag'aza etivi mu'mkin. Biraq bul hal jaqsı bahalanbaydı, ol klasslar shejiresi teren' oylamastan du'zilgenliginin' da'lili esaplanadı. Sog'an qaramastan C++ tili bug'an imkan beredi. O'z-ara dos klasslar to'mendegishe dag'aza qiling'an bolıwı kerek:

```

class Klass2;
class Klass1{
friend class Klass2;
///...
};
class Klass2
{
friend class Klass1;
//...
};

```

Joqarıda keltirilgen programma bo'leginde klasstın' tolıq emes dag'azası tek Klass1 ishinde Klass2 klassına ko'rsetkish bolg'andag'ana za'ru'r boladı.

Dos klasslar ushın to'mendegi qag'ıydalar a'mel qıladı:

-dosliq o'z-ara emes, yag'niy Klass1 klassi Klass2 klasstin' dosti ekenliginen Klass2 klassin Klass1 klass dosti ekenligi kelip shiqpaydi;

- dosliq miyrasliq boyinsha o'tpeydi, eger Klass1 klassi Klass2 klassinin' dosti bolsa, Klass1 klassinin' a'wladlari Klass2 klass doslari bolmaydi;

-dosliq tayansh klass a'wladlarina o'tpeydi, eger Klass1 klassi Klass2 klassinin' dosti bolsa, Klass1 klassin Klass2 a'wladlarina dos bola almaydi.

Friend Functions

It is possible to allow a non-member function access to the private members of a class by declaring it a *friend* of the class. To make a function a friend of a class, include its prototype in the **public** section of the class declaration and precede it with the **friend** keyword. For example, in this fragment **frnd()** is declared to be a friend of the class **cl**:

```
class cl {
    // ...
public:
    friend void frnd(cl ob);
    // ...
};
```

As you can see, the keyword **friend** precedes the rest of the prototype. A function may be a friend of more than one class.

The third reason that friend functions may be desirable is that, in some cases, two or more classes may contain members that are interrelated relative to other parts of your program. For example, imagine two different classes that each display a pop-up message on the screen when some sort of event occurs. Other parts of your program that are designed to write to the screen will need to know whether the pop-up message is active, so that no message is accidentally overwritten. It is possible to create a member function in each class that returns a value indicating whether a message is active or not; however, checking this condition involves additional overhead (i.e., two function calls, not just one). If the status of the pop-up message needs to be checked frequently, the additional overhead may not be acceptable. However, by using a friend function, it is possible to directly check the status of each object by calling only one function that has access to both classes. In situations like this, a friend function helps you write more efficient code. The following program illustrates this concept.

31-Leksiya. Miyrasli'q. Apıwayi miyrasli'q. Koplikli miyrasli'q

Joba:

1. Klasslar ortasındag'ı qatnaslar.
2. Miyraslıq.
3. Apıwayı miyraslıq.
4. Ko'plikli miyraslıq.
5. Virtual tayanış klasslar

Klasslar ortasındag'ı qatnaslar: Ayırım OBP tillerde klasslar ortasındag'ı to'mendegi qatnaslardı qollap-quwatlaydı:

- assotsiatsiya;
- miyraslıq;
- agregatsiya;
- baylanıslıq (paydalanıw).

Assotsiatsiya. Eger eki klass konseptuallıq da'rejede bir-birine ta'sir o'tkizse, ol jag'dayda bunday o'z-ara ta'sirge assotsiatsiya delinedi. Mısalı, sawda orayı jumısın modellestiriwde eki abstraksiya (tu'sinigi) ju'zege keledi: Zatlar (Zatlar klassı) ha'm satıw (satıw klassı). Satıw klassının' ob'ekti-bul qandaydur ha'diyse bolıp, onda 1 den n ge shekem zat klassının' ob'ektleri satılg'anlıg'ın an'latadı.

Assotsiatsiya rejesinin' baslang'ış adımlarında anıqlanatug'ın eki klass ortasındag'ı en' ko'p payda bolg'an semantik baylanıslardı an'latadı. Assosiativ baylanıslar miyraslıq, agregatsiya, baylanıs munasebetleri arqalı konkretlestiriledi.

Agregattsiya. Klasslar ortasındag'ı agregatsiya munasebetı bir klass o'zının' quramı bo'legı sıpatında basqa klasslerin o'z ishine alg'an jag'daylar payda boladı. Basqasha aytqanda bul munasebet "putın/bo'lek" munasebetdur.

Agregatsiyanı eki ko'rınısında a'melge asırıw mumkin.

- qatan' emes agregatsiyalar (apıwayı agregatsiyalar);
- qatan' agregatsiyalar(kompozitsiyalar).

Qatan' emes agregatsiyada dag'aza qiling'an ha'mme bo'limler pu'tinge kırıwı shart emes. Sol sebepli, "bo'limnın' pu'tinge kırıwı" ko'rsetkish

quralı ja'rdemide a'melge asırıladi ha'm olardin' ha'mmesi initsializatsiya qılınıwı shart emes (ayırım ko'rsetkishler 0 ma'nısge iye boladı). Sheshılıp atırg'an ma'selege baylanıslı ta'rizde bunday komponentalar dinamik ta'rizde payda bolıwı yaki joytılwı mumkin.

Qatan' agregatsiya munasebetinde komponenta (bo'lım) putın ob'ekt bar waqtında joq bolıwı mumkin bolmaydı. Ma'selen, ushmu'yesh (u'shmuyesh klassı) u'sh tochkani (tochka klassı ob'ektlerin) o'z ishine aladi ha'm tochkalardan birewi bolmasa, u'shmuyesh jasaw mumkin emes.

Kompozitsiyanı a'melge asırıwdın' apıwayı jolı- kompozitsiya ob'ektler-komponentlardı ma'nısları menen kırıw jolı (apıwayı o'zgerıwshıler sıpatında). Eger komponentlar ko'rsetkishler quralı ja'rdemide a'melge asırılsa, olar to'mendegi talaplarg'a juwap berıwleri kerek: komponentalardın' jasaw dawırı pu'tın ob'ekt jasaw dawırı menen u'stpe-u'st tusıwı kerek.

Baylanıslıq. Eki klass ortasında baylanıslıq (paydalanıw) munasebeti bir klass (tiykarg'ı klassı) basqa klass (ja'rdemshı klassının') xizmetlerinen paydalanıladı.

Ma'selen:

-Tiykarg'ı klass funksiya-ag'zası Ja'rdemshı klassdın' ayırım maydanları ma'nıslarinen paydalanıladı;

-Tiykarg'ı klass -ag'zası Ja'rdemshı klassdın' funksiya-ag'zasın shaqıradı;

-Tiykarg'ı klass funksiya-ag'zasının' prototıpında Ja'rdemshı klass tu'rındağı parametrlar qatnasadı.

Mısal ushın, Ushmu'yesh klassında ushmu'yesh ta'repınin' uzınlıg'ın esaplaytug'ın funksiya- ag'zası dag'azasında Tochka klassı tu'rındağı parametrlar qatnasadı;

double Ushmu'yesh:: Aralıq(Toshka a, Toshka b);

Miyraslıq. Miyraslıq ("is a") klass astınin' (tuwındı klassınin') bir yaki bir neshe *superklasslarında* (*tayanısh klasslarda*) anıqlang'an atribut ha'm a'mellerdı birgelikte isletedı. Ma'selen , Ko'pmu'yeshlik klassınin' miyrası bolg'an Ushmu'yeshlik klassı Ko'pmu'yesh Tayanısh klassı ag'zaların isletiw mumkin (qayta anıqlamasdan)

Apıwayı miyraslıq: C++ tili klassg'a bir yaki birneshe klasstın' berilgen-ag'zaları ha'm funktsiya-ag'zaların miyraslıq penen alıwg'a imkan beredi. Bul jag'dayda jan'a klasstı tuwindı klass (awlad klass) delinedi. Elementlerin miyraslıq penen alatug'ın klassg'a onın' miyrasına salıstırg'anda tayanış klass (awlad klass) delinedi.

Miyraslıq klasslardın' ulıwma qasiyetlerin tek bir tayanış klassda abstraksiyalaw imkanın beredi. Miyras klasslar usı ulıwma qasiyetlerin miyraslıq penen alg'an jag'dayda , ayırım funktsiya-ag'zaların qayta anıqlaw yaki qosımsha berilgen-ag'zalar ha'm funktsiya-ag'zaların kırıtw arqalı tayanış klassga ma'lim bir o'zgeriw kırıtwleri mu'mkin. Uss sebepli tuwindı klasslardı anıqlaw sezirerli da'rejede kemeyedi, sebebi og'an tek g'ana tayanış klassdan farqlı elementler qosıladı.

Tuwindı klassdı dag'aza qılıw sintaksısı to'mendegishe boladı:

```
Klass <tuwindi klass ati>:<mu'rajat qa'siyeti><tayanish klass1>,
<mu'rajat qa'siyeti><tayanish klass2>,
.....
<mu'rajat qa'siyeti><tayanish klassn>,
{
//tuwindi klass qurami
};
```

Bul jerde < Murajet qasiyeti> public, protected yaki private ha'm bolıwı shart emes. Eger < Murajet qasiyeti > bolmasa kelisiw boyınsha klasslar ushın private duzilıs ha'm birlespesı ushın public esaplanadı. <Tayanish klass> aldınnan anıqlangan tayanış klasslardın' atları.

Keste. Klasslardın' miyraslıq variantları

Mu'rajat qa'siyeti	Tayanish klassdag'i mu'rajat turi	Miyras klassdag'i saykes mu'rajat turi
Public	public protected private	Public protected mu'rajat joq
Protected	public protected private	protected protected mu'rajat joq

Private	public protected private	private private mu'rajat joq
---------	--------------------------------	------------------------------------

Mu'rajat qa'siyeti Tayanish klass elementlerinin' miyrasliq boyınsha alıw da'rejesin anıqlaydı. 4.1-sxemada miyrasliq variantları keltirilgen.

4.1-sxema menen tanısıw sonı ko'rsetedi, miyrasliqtın' mu'rajat qa'siyeti tayanish klassda ornatılǵ'an mu'rajetti qanday da'rejede tusgenlıǵın ko'rsetedi(ashıq ag'zalarga salıstırǵ'anda)

Eger tayanish klassdag'ı ag'zalarg'a tuwindı klasslarda mu'rajat qılıw za'ru'r bolsa olardı public dep dag'aza qılıw kerek boladı.

Mısali.

```
class Tayanish
{
int x,y;
public:
Tayanish(int_x=5, int_y=10){x=_x; y=_y;}
int X_manis(){return x;}
int Y_manis(){return y;}
};
class tuwindi: private Tayanish
{
public:
int Tayanish::X_manis();};
int main()
{
int X,Y;
tuwindi ob;
x=ob.X_manis();
y=ob.Y_manis();
cout<<"X="<<X<<" ,Y="<<Y<<endl;
return 0;}
```

Programmada tuwindi klassi tayanish klassdag'ı X_manisi() ha'm Y_manisi() funktsiyalardi private da'rejesindeki mu'rajat penen miyrasliq boyinsha aladi ha'm ekrang'a

X=5, Y=10

An'latpalarin ko'rsetedi.

Soni qayt etiw kerek, tuwindi klass tayanish klassinin' tek public yaki protected ag'zalarina mu'rajat qilw mumkin. Tayanish klasstin' jabiq ag'zalarina ha'r qanday miyrasliqda ha'm jabıqlıg'ınsha qaladı.

Tayanish klasstin' to'mendegi elementleri miyrasliq penen o'tpeydi.

-konstruktorlar;

-kopyalaw konstruktorlari;

-destruktorlar;

-programma du'zıwshısı tarepinen anıqlang'an ma'nis beriw operatorı;

-klasstin' doslari;

Tuwindi klass bir tayanish klassg'a iye bolsa, bunday miyrasliq apıwayı (jeke) miyrasliq delinedi.

To'mendegi misalda apıwayı miyrasliq a'melge asırılğ'an.

```
#include <iostream.h>
```

```
class Toshka
```

```
{
```

```
int x,y;
```

```
public:
```

```
Toshka(int_x, int_y){x=_x; y=_y;}
```

```
Toshka(){x=0; y=0;}
```

```
int X_manis(){return x;}
```

```
int Y_manis(){return y;}
```

```
void manis_X(int_x){x=_x;}
```

```
void manis_Y(int_y){y=_y;}
```

```
};
```

```
class Dekart: public Toshka{
```

```
public:
```

```
Dekart (int_x, int_y): Toshka(2*_x, 2*_y){};
```

```
void manis_XX(){cout<<X_manis_X()<<" "};
```

```

void manis_YY(){cout<<Y_manis_Y()<<" ";}
};
int main()
{
Dekart *dkt;
dkt=new Dekart(10,15); //obiecti uyimda jaratiw
dkt->manis_XX();
dkt->manis_YY();
delete dkt; //obiecti o'shiriw
return 0;}

```

Usı programma Dekart tuwindı klassı bir Tochka tayanış klassga iye. Konstruktor miyraslıq boyınsha o'tpegenlıgı sebepli, miyras klassda konstruktor sintaksisi o'zgergen. Malımki, miyras klass tayanış klassdin' ha'mme berilgenlerin miyraslıq boyınsha aladi (tayanış klassdin' ayırım ag'zalarına tuwrıdan-tuwrı mu'rajat qıla almasa ha'm) ha'm klass ob'ekti anıqlanıwında bul berilgenler initsializatsiya qılınıwı kerek. Sebebı, miyras klassda bul berilgenler isletiwı mumkin. Usı sebepli tuwindı klass konstruktori to'mendegi duzilıs ko'rınısında anıqlanadı.

<tuwindi klass konstruktori>(<parametrler dizimi>):

<tayanish klass konstruktori>(<argumentler dizimi>){ <konstruktor denesi>}

Konstruktor tayanış klass konstruktor'ga ko'rsetiwshı elementlerin initsializatsiyalaw diziminen paydalang'an. Miyras klass konstruktor'ga uzatıl'g'an argumentlerdin' ma'lim bir bo'limı tayanış klass konstruktorna argument sıpatında uzatıladı. Keyin miyras klass konstruktori denesinde tap usı klass ushin za'ru'r initsializatsiya a'mellerin orınlanıwı mumkin. Joqarıda keltirilgen misalda Dekart miyras klassı konstruktori

```
Dekart(int x, int y):Toshka(2*x,2*y){};
```

Ko'rınısında anıqlang'an. Miyras klass ob'ektin initsializatsiya ushin uzatıl'g'an argumentler ekige ko'beytilgen jag'dayda tayanış klass berilgen-ag'zaların initsializatsiyalaw ushin isletilmekde. Konstruktor denesi bos, sebeбі bul klass o'z berilgenlerine iye emes.

Eger miyras klass konstruktoriga parametrsiz aniqlang'an bolsa, yag'nuy kelisw boyinsha konstruktor bolsa, tuwindi klass ob'ektin jaratıwda tayanish klass konstruktoriga avtomatik ta'rizde shaqiriladi.

Tuwindi klass destruktorg'a salıstırg'anda to'mendegi nizamg'a a'mel qiladi: Tuwindi klass destruktoriga tayanish klass destruktorigan aldın islewı kerek. Kerı jag'dayda tayanish klass destruktoriga sol waqıtta islep atırg'an tuwindi klass berilgen-ag'zaların o'shırıwı mu'mkin. Destruktorlar tuwrı islewı kompilyator o'z moyına aladi.

To'mende keltirilgen programmada ha'm miyras konstruktoriga ha'm destruktoriganin' islew izbe-izlıgı ko'rsetilgen.

```
#include <iostream.h>
class Tayanish
{
public:
Tayanish(){cout<<"Tayanish klass konstruktoriga isledi.\n";}
Tayanish(){cout<<"Tayanish klass destruktoriga isledi.\n";}
};
Class Tuwindi: public Tayanish
{
public:
Tuwindi(){cout<<"Tuwindi klass konstruktoriga isledi.\n";}
Tuwind(){cout<<"Tuwindi klass konstruktoriga isledi.\n";}
};
int main()
{
Tuwindi ob;
Return 0;
}
```

Ko'p jag'daylarda tuwindi klassda tayanish klass ag'zaları qatarına jan'a ag'zalar qosiladi. Biraq tayanish klass ag'zaları qayta aniqlaw imkaniyatı bar. A'dette tayanish klass funktsiya-ag'zaların qayta aniqlanadi. Bunun' ushın tayanish klass funktsiya-ag'zasın tuwindi klass dag'azası prototıpın berıw ha'm keyininen onı aniqlaw jetkılıklı. Misal etkende Tochka tayanish

klassda aniqlang'an X_manis() ha'm Y_manis() funktsiyalarin qayta aniqlanivvin qarayıq:

```
#include <iostream.h>
class Toshka
{
protected :
int x,y;
public:
Toshka(int_x, int_y){x=_x; y=_y;}
Toshka(){x=0; y=0;}
int X_manis(){return x;}
int Y_manis(){return y;}
void manis_X(int_x){x=_x;}
void manis_Y(int_y){y=_y;}
};
class Dekart: public Toshka{
public:
Dekart (int_x, int_y): Toshka(_x, _y){};
int X_manis {return ++x;}
int Y_manis {return ++y;}
void manis_XX(){cout<<X_manis_X()<<" ";}
void manis_YY(){cout<<Y_manis_Y()<<" ";}
};
int main()
{
Dekart *dkt;
dkt=new Dekart(10,15); //obiecti uyimda jaratiw
dkt->manis_XX();
dkt->manis_YY();
delete dkt; //obiecti o'shiriw
return 0;}
```

Programma islewı natıyjesinde ekrang'a berilgen koordinatalardın' birge asırılğ'an ma'nislerin dag'aza etiledı. Bul programmada tayanish klassdın'

X_manis() ha'm Y_manis() funktsiyalarin qayta aniqlang'an bolsa ha'm olardin' alding'i variantlarina mu'rajat qilw mumkin. Bunun' ushin "ko'riniw oblastindag'i ruxsat beriw" a'melinen paydalanw za'ru'r boladi. Joqaridag'i misalda keltirilgen programma tekstine

.....

```
void manis_XX(){cout<<Toshka::X_manis_X()<<" "};
```

```
void manis_YY(){cout<<Toshka::Y_manis_Y()<<" "};
```

o'zgartiw arqali qayta aniqlang'an funktsiyanin' tayanish klassdag'i aniqlanwın shaqiriw mumkin.

Tuwındı klass o'z nawbetinde basqa bir klass ushin tayanish bolıwı mumkin. Bunda tayanish klass ushinshi qurallı tayanish klass boladı.

Ko'plıklı miyraslıq: Eger tuwındı klass bir neshe tayanish klasslarga iye bolsa, bunday miyraslıq ko'plıklı miyraslıq delinedı. Ko'plıklı miyraslıq bir klassda bir neshe klass qasiyetlerin ha'm a'mellerin ja'mlew imkanın beredi.

To'mendegi misal ko'plıklı miyraslıqtı payda etedi:

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
#include <string.h>
```

```
class Toshka{
```

```
int x,y;
```

```
public:
```

```
Toshka(int_x, int_y){x=_x; y=_y;}
```

```
Toshka(){x=0; y=0;}
```

```
int X_manis(){return x;}
```

```
int Y_manis(){return y;}
```

```
void manis_X(int_x){x=_x;}
```

```
void manis_Y(int_y){y=_y;}
```

```
};
```

```
class Xabar _qabil_qiliw
```

```
{
```

```
char Xabar[80];
```

```
public:
```

```

Xabar_qabil_qiliw(char* xbr){Xabardi_saqlaw(xbr);}
void Xabardi_saqlaw(char* xabar) {strcpy(Xabar, xabar);}
void Xabar_beriw (){cout<<Xabar;}
};
class Xabardi_dag'aza_qiliw: public Toshka,
public Xabardi_dag'aza_qiliw
{
public:
Xabardi_dag'aza_qiliw(int _x, int _y, char *xbr):
    Toshka(_x,_y),Xabar_qabil_qiliw(xbr){}
void dag'aza_qiliw();
};
void Xabardi_dag'aza_qiliw()::dag'aza_qiliw();
{
gotoxy(X_manis(), Y_manis()); //Kursor xabar shig'ariw toshkasina
Xabar_beriw();
}
Xabardi_dag'aza_qiliw() *xbr_adress;
xbr_adress=new Xabardi_dag'aza_qiliw(10,5, "ko'plikli");
xbr_adress->dag'aza_qiliw();
xbr_adress->manis_X(10);
xbr_adress->manis_Y(6);
xbr_adress->Xabardi_saqlaw("miyrasliq!");
xbr_adress->dag'aza_qiliw();
delete xbr_adress;
return 0;
}

```

Usı mısalda Tochka klass ekrandag'ı belgı orınnı koordinatasın ornatiw ha'm saqlawg'a Xabar_aliw klassı xabardı qabil qiliw ha'm saqlawg'a xizmet qiladı. Olardıń miyrası bolg'an Xabardi_dag'aza_qiliw klassı esa xabardı ekrannın' ko'rsetilgen koordinatada dag'aza qiladı. Mısal tayanış ha'm miyraslıq klasslar konstruktorlardı qanday islewı kerekligin ko'rsetedi. Ko'plikli miyraslıqda, tap apıwayı miyraslıqday, aldın tayanış

klasslar konstruktorlari, keyninen miyras klass konstruktori islew kerek. Tayanish klass konstruktorlarinin' qanday izbe-izlikde islewı, olardin' miyras klassı dag'azanin' initsializatsiya dızımında ornı menen anıqlanadı. Mısalda Xabardi_dag'aza_qiliw klassı miyraslıq haqqında to'mendegı dag'azag'a iye.

```
class Xabardi_dag'aza_qiliw: public Toshka, public Xabardi_qabil_qiliw
```

Bul dag'azag'a to'mendegı konstruktor saykes keledi:

```
Xabardi_dag'aza_qiliw(int _x, int _y, char *xbr):
```

```
    Toshka(_x,_y),Xabar_qabil_qiliw(xbr){}
```

Programmanın' bas funksiyasınan Xabardi_dag'aza_qiliw klassının' dag'azası arqalı konstruktor'ga argumentler uzatıladı:

```
xbr_adress=new Xabardi_dag'aza_qiliw(10,5, "ko'plikli");
```

Keyingi qatardag'ı dag'aza_qiliw() funksiyası ekrannın' (10, 5) koordinatasında "ko'plikli" qatardı dag'aza qıladı. Manis_X(10), Manis_Y(6) ha'm Xabardi_saqlaw("miyraslıq!") funksiyaların shaqırıw arqalı jan'a berilgenler o'zlestiredi. Ekinshi ma'rte dag'aza_qiliw() funksiyasın shaqırıw arqalı ko'rsetilgen koordinatada basqa xabar shıg'adı.

Programma islewı na'tıyjesinde ekrang'a

Ko'plikli

Miyraslıq!

Xabar dag'aza etiledi.

Ko'plikli miyraslıqda konstruktor ha'm destruktordı a'mel qılıw izbe-izligın to'mendegı programma mısasında ko'riw mumkin.

```
#include <iostream.h>
```

```
class Tayanish1
```

```
{
```

```
public:
```

```
Tayanish1(){cout<<"Tayanish1 klass konstruktori isledi.\n"}
```

```
Tayanish1(){cout<<"Tayanish1 klass destruktora isledi.\n"}
```

```
};
```

```
class Tayanish2
```

```
{
```

```

public:
Tayanish2(){cout<<"Tayanish2 klass konstruktori isledi.\n"}
Tayanish2(){cout<<"Tayanish2 klass destruktori isledi.\n"}
};
class: public Tayanish1, public Tayanish2
{
public:
Tuwindi(){cout<<"Tuwindi klass konstruktori isledi.\n";}
Tuwindi(){cout<<"Tuwindi klass destruktori isledi.\n";}
};
int main()
{
Tuwindi ob;
return 0;
}

```

Programma islewı na'tiyjesinde ekrang'a to'mendegı xabarlar dag'aza qılınadı.

```

Tayanish1class konstruktör isledi.
Tayanish2 class konstruktör isledi.
Tuwindi klass konstruktör isledi.
Tuwindi klass destruktör isledi.
Tayanish2 klass destruktör isledi.
Tayanish1 klass destruktör isledi.

```

Mısal sonı ko'rsetedı, konstruktörler olardın' dag'azasındag'ı quramına saykes tarızde, destruktörler esa kerı tarızde isge tu'sedı.

Virtual tayanış klasslar: Ko'plıklı miyraslıqda klasslardın' salıstırmalı quramalı shejeresinde tuwindı klass quralı ta'rizde bir klassdan eki klassqa yaki onnan artıq kopyasın miyraslıq arqalı alıwı mumkin. To'mende keltirilgen mısal bul jag'daydı ashıp beredı.

```

#include <iostream.h>
class Tayanish
{

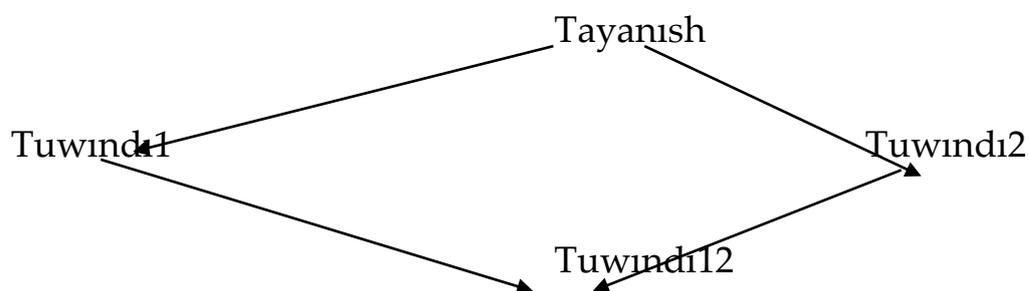
```

```

int x,y;
public:
int X_manis(){return x;}
void manis_X(int _x) {x=_x;}
double y;
};
class Tuwindi1 public Tayanish
{
//...
};
class Tuwindi12 public Tuwindi1, public Tuwindi2;
{//...};
main()
{
Tuwindi12 ob;
ob.y=5.0;
ob.manis_X(10);
int z=ob.X_manis();
return 0;
}

```

Bul misalda Tuwindi12 klassi o'zining tayanish klasslar_ Tuwindi1 ha'm Tuwindi2 klasslar orqali qurali ta'rizde Tayanish klassni miyrasliq boyinsha aladi. Bul jag'daydin' grafik ko'rinishi 4.1-su'wrette keltirilgen.



Miyraslıqtın' bunday ko'rınısında Tuwindi12 klassda Tayanish klassdın' eki kopiyası payda boladı ha'm Tayanish klassdın' y ha'm manis_X ag'zalarına

```
ob.y=5.0;
ob.manis_X(10);
int z=ob.X_manis();
```

ko'rınısında mu'rajetlerge kompilatsiya qa'tesi ju'zege keledi. Sebebi kompilayator bul awlad klasslardın' funktsiya-ag'zalarının' qaysı kopiyasına mu'rajet bolıp atırǵanlıǵın anıqlay almaydı. Bul Orında "ko'riniw oblastına ruxsat beriw" a'meli ja'rdeminde qaysı tayanish klass arqalı awlad klassg'a mu'rajet bolıp atırǵanlıǵın ko'rsetiw mumkin.

```
ob.Tuwindi1::y=5.0;
ob.Tuwindi1::manis_X(10);
int z=ob. Tuwindi1::manis_X();
```

Sondada "::" a'melinen paydalanıw anıqemeslikti joytsa ha'm, Tuwindi12 klassında Tayanish klassının' eki kopiyası payda bolıwın joq qilmaydı. Bul ma'selenı sheshiw ushın kompiliyatorg'a virtual tayanish klassın isletiw haqqında ko'rsetpe beriw kerek boladı. Virtual tayanish klassdı ko'rsetiw ushın miyraslıq ko'rsetiletug'ın Orında mu'rajet qa'siyetinen aldın yaki keyin virtual gılt so'zın jazıw kerek boladı. Joqarıdag'ı mısalda Tuwindi1 ha'm Tuwindi2 klasslar dag'azasına to'mendegi o'zgeriwler qılıw arqalı Tuwindi12 klassında Tayanish klassının' eki kopiyası payda bolıwının' aldı alınadı:

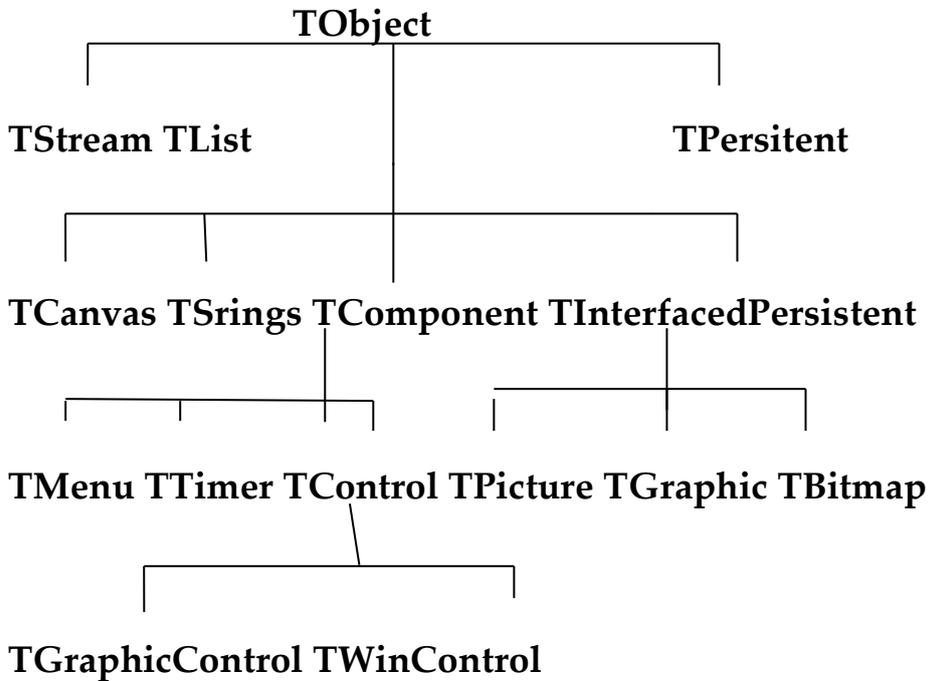
```
.....
class Tuwindi1: virtual public Tayanish
.....
class Tuwindi2: virtual public Tayanish
.....
```

Programanın' bul variantında

```
ob.Tuwindi1::y=5.0;
ob.Tuwindi1::manis_X(10);
int z=ob. Tuwindi1::manis_X();
```

mu'rajetler kompiliyatsiya qa'tesine alıp kelmeydı.

Soni qayd etiw kerek, klasslar shejerasinde virtual klasslar bolg'anda konstruktorlar orinlanıwı izbe-izligı o'zgeredi: aldın virtual klasslar konstruktorları isge tu'sedi, keyninen anıq emes virtual awlad klasslar konstruktorları ha'm aqırında tuwındı klass konstruktorı isleydı. Destruktorlar islewı usı izbe-izliginde kerı ta'rizde a'mel qıladı.



Sorawlar ha'm tapsırmalar

1. Klasslar ortasındag'ı munasebetlerdı aytıp berın'.
2. Miyraslıqdag'ı mu'rajat qasiyetleri neni an'latadı?
3. Apıwayı ha'm ko'plıklı miyraslıqdı sxema ko'rınısında aytıp berın'.
4. Du'zılıs ha'm birlespelerde miyraslıqtı a'melge asırsa bolama?
5. Miyraslıqda konstruktor ha'm destruktordı a'mel qılıw ta'rtıbı qanday?
6. Miyras klass konstruktorının' o'zine ta'n ko'rınısının' sebebı nede?
7. Ko'plıklı miyraslıqda virtual tayanış klasslar isletıwı sebebı ne?

Inheritance is one of the cornerstones of OOP because it allows the creation of hierarchical classifications. With inheritance, it is possible to create a general class that defines traits common to a set of related items. This class may then be inherited by other, more specific classes, each adding only those things that are unique to the inheriting class.

In standard C++ terminology, a class that is inherited is referred to as a *base class*. The class that does the inheriting is called the *derived class*. Further, a derived class can be used as a base class for another derived class. In this way, a multilayered class hierarchy can be achieved.

When one class inherits another, the members of the base class become members of the derived class. The access status of the base class members inside the derived class is determined by the access specifier used for inheriting the base class. The base class access specifier must be **public**, **private**, or **protected**. If the access specifier is not used, then it is **private** by default if the derived class is a **class**. If the derived class is a **struct**, then **public** is the default in the absence of an explicit access specifier. Let's examine the ramifications of using **public** or **private** access. (The **protected** specifier is described in the next section.)

When a base class is inherited as **public**, all public members of the base class become public members of the derived class. In all cases, the private elements of the base class remain private to that class, and are not accessible by members of the derived class. For example, in the following program, the public members of **base** become public members of **derived**. Thus, they are accessible by other parts of the program.

When a class member is declared as **public**, it can be accessed by any other part of a program. When a member is declared as **private**, it can be accessed only by members of its class. Further, derived classes do not have access to private base class members. When a member is declared as **protected**, it can be accessed only by members of its class, or by derived classes. Thus, **protected** allows a member to be inherited, but to remain private within a class hierarchy.

When a base class is inherited by use of **public**, its public members become public members of the derived class, and its protected members become protected members of the derived class.

When a base class is inherited by use of **protected**, its public and protected members become protected members of the derived class.

When a base class is inherited by use of **private**, its public and protected members become private members of the derived class.

In all cases, private members of a base class remain private to the base class, and are not inherited.

As you become more familiar with C++, the meaning of **public**, **protected**, and **private** will become second nature. For now, if you are unsure what precise effect an access specifier has, write a short sample program as an experiment and observe the results.

Inheriting Multiple Base Classes

It is possible for a derived class to inherit two or more base classes. For example, in this short program, **derived** inherits both **base1** and **base2**:

As you can see, the constructor of **base** is executed, followed by the constructor of **derived**. Next (since **ob** is immediately destroyed in this program), the destructor of **derived** is called, followed by that of **base**.

The results of the foregoing experiment can be generalized as follows: When an object of a derived class is created, the base class constructor is called first, followed by the constructor for the derived class. When a derived object is destroyed, its destructor is called first, followed by the destructor for the base class. Put differently, constructors are executed in the order of their derivation. Destructors are executed in reverse order of derivation.

If you think about it, it makes sense that constructor functions are executed in the order of their derivation. Because a base class has no knowledge of any derived class, any initialization it needs to perform is separate from, and possibly prerequisite to, any initialization performed by the derived class. Therefore, it must be executed first.

Likewise, it is quite sensible that destructors be executed in reverse order of derivation. Since the base class underlies a derived class, the destruction of the base class implies the destruction of the derived class. Therefore, the derived destructor must be called before the object is fully destroyed.

In the case of a large class hierarchy (i.e., where a derived class becomes the base class for another derived class), the general rule applies: Constructors are called in order of derivation, destructors in reverse order. For example, this program

When a base class is inherited as private, all members of that class (public, protected, or private) become private members of the derived class. However, in certain circumstances, you may want to restore one or more inherited members to their original access specification. For example, you might want to grant certain public members of the base class public status in the derived class, even though the base class is inherited as private. You have two ways to accomplish this. First, you may use a **using** declaration within the derived class. This is the method recommended by Standard C++ for use in new code. However, a discussion of **using** is deferred until later in this book when namespaces are examined. (The primary reason for **using** is to provide support for namespaces.) The second way to adjust access to an inherited member is to employ an *access declaration*. Access declarations are still supported by Standard C++, but they have recently been deprecated, which means that they should not be used for new code. Since they are still used in existing code, a discussion of access declarations is presented here.

An access declaration takes this general form:

```
base-class::member;
```

The access declaration is put under the appropriate access heading in the derived class. Notice that no type declaration is required (or allowed) in an access declaration.

To see how an access declaration works, let's begin with this short fragment:

```
class base {
public:
    int j; // public in base
};

// Inherit base as private.
class derived: private base {
public:

    // here is access declaration
    base::j; // make j public again
    // ...
};
```


32-Leksiya. Funktsiyalarni qayta yuklew. Klasstin' funktsiya-ag'zalarin qayta yuklew

Joba:

1. Funktsiyalardı qayta yuklew din' abzallıg'ı
2. Klasstin' funktsiya ag'zalarin qayta yuklew
3. Konstruktorlardı qayta yuklew

Funktsiyalardı qayta yuklew din' abzallıg'ı:

Funktsiyalardı qayta yuklew- bul C++ tili ta'repinen qollap kuwatlanatug'in polimorfizim ko'rinislerinen biri. C++ tilinde birneshe funktsiyalar bir turdegi atqa iye bolıwı mumkin. Bul attag'ı funktsiyag'a qayta yuklenushi funktsiya delinendi. Argumentlerdin' turleri yaki sanı menen parqlanatug'in funktsiyalar g'ana qayta yuklenedi. Qaytarushı ma'nis turi menen parq qılatug'in funktsiyalardı qayta yuklep bolmaydı. Funktsiyalardı qayta yuklew arqalı mazmun jag'ınan ha'r turli a'mel orınlaushı funktsiyalardı birdey at penen da'sturdi a'puaylastırıw mumkin.

Funktsiyalardı qayta yuklew ushın onı dag'aza etiw ha'm keyin za'rur bolg'an barlıq shaqırıu variantların anıqlau kerek boladı. Kompilyatordın' o'zi argumentler sanı ha'm turine mas ra'uishte funktsiyanın' kerekli variantın tan'laydı. Mısal:

```
#include<iostream>
//abs() funktsiyanı qayta yuklew
int abs1(int n);
long abs1(long l);
double abs1(double d);
main(){
    int n=125; long l=-26; double d=-15.0;
    cout<<n<<" sannın' obsalyut manisi="<<abs1(n)<<endl;
        cout<<l<<" sannın' obsalyut manisi ="<<abs1(l)<<endl;
    cout<<d<<" sannın' obsalyut manisi ="<<abs1(d)<<endl;
    return 0;
}
```

```
int abs1(int n){return n<0?-n:n;};
long abs1(long l){return l<0?-l:l;};
double abs1(double d){return d<0?-d:d;};
```

C++ tili kitapxanasında sannın' absolyut ma'nisi esaplaw ushın argument turine mas ra'wishte abs(), labs(), ha'm fabs(), funktsiyaları anıqlanadı. Joqarda keltirilgen misaldı sannın' kaday turde bolıunan katti na'zer bir abs() funktsiyası shaqırılǵ'an ha'm sonın' esabına programma tekstı a'puaylaskan.

Klasstın' funktsiya ag'zaların qayta juklew:

Funktsiyalardı qayta juklewde to'mendegilerge itibar beriw kerek:

- 1) Argumentleri ha'r turli insalizatorlarga iye funktsiyalar ha'r turli funktsiya esaplanadı ha'm olardı qayta juklew mumkin;
- 2) Eki funktsiya parametrları diziminde kadayda bir turdegi parametr bolıp, ekinshisinde mass orında usı turdegi adres olıushı parametr bolsa, bunday funktsiyalar qayta juklew ko'z-karasınan usı bir turdegi ko'rinis dep esaplanadı ha'm olar qayta juklenbeydi. Mısalı, dag'azaları

```
int func(int,int);
```

ham

```
int func(int&,int&);
```

kompilatsiya ka'tesine alıp keledi, sebebi olar qayta juklew sha'rtleri boyınsha birdey esaplanadı.

- 3) Funktsiyalardı kadayda bir turge tiysli argumentleri bir-birinen const ha'm volatile modifikatorları menen parqlansa, qayta juklew ko'z qarasınan olar birdey dag'aza esaplanadı ha'm bul parq penen funktsiyalardı qayta juklew mumkin emes;
- 4) Funktsiyalardı parametrleri dizimindegi adres aliushı parametr bolǵ'an xalda ekinshi funktsiyadag'ı mas orındag'ı adres aliushı parametr const ha'm volatile modifikatorı menen berilgen bolsa,

qayta juklew ko'z qarasinan bul funktsiyalar parqlanadı ha'm olardı qayta juklew mumkin. Ma'selen, to'mendegi qayta juklew orinlı.

```
#include <iostream>
using namespace std;
class klass{
    klass(){cout<<"klass ushin kelisiw boyinsha konstruktor\n";}
    klass(klass& ob){cout<<"klass ushin nusqalaw konstruktori\n";}
    klass(const klass& ob){cout<<"klass ushin nusqalaw konstruktori\n";}
};
int main(){
    klass ob1; // kelisiw boyinsha konstruktor
    klass ob2(ob1); //nusqalaw konstruktori
    const klass ob3;
    klass ob4(ob3); // konstanta obiektin nusqalaw konstruktori isledi
    return 0;
}
```

- 5) Ha'r kaday eki qayta juklenwshi funktsiyalar ha'r turli parametrler dizimine iye bolıu kerek;
- 6) Funktsiya ag'zaları tek olardıń biri static, ekinshisinde joqlıg'ı tiykarında qayta juklew mumkin emes;
- 7) Typedef-anıqlaular qayta juklew mixanizimine ta'sir etpeydi, sebebi olar jana turdi jaratpaydı, olar bar turdegi sinonimii esaplanadı. Misalı to'mendegi
Typedef char *PSTR;
anıqlau kompilyatorg'a <<void Satr(char * s); >> ha'm <<void Satr(PSTR * s);>> ko'rinisindegi funktsiya dag'azaların ha'r turli dep esaplaug'a imkaniyat bermeydi ha'm olar qayta juklenbeydi;
- 8) Barlıq enum turleri ha'r turli esaplanadı ham olardan funktsiyalardı qayta juklewde paydalanıu mumkin;

- 9) Massiv ha'm ko'rsetkishler qayta juklew ko'z qarasına birdey dag'aza esaplanadı. Birak bul pikir tek bir o'lsheimli massivler ushin ornalı. Sol sebepli, «void Satr(char*s);» ha'm «void Satr(char s[]);» ko'rinisindegi funktsiyalardı qayta juklewge urınıu kompilatsiya qa'tesine alıp keledi;
- 10) Ko'p o'lsheimli massivlarda ekinshi ha'm onnan keiyngi o'lsheimler berilgenler turinin' ayrıqsha bir bo'legi dep qaraladı ha'm bul jag'day qayta juklew ushin tiykar boladı. To'mendegi funktsiyalar qayta jukleniwi munkin:

```
void Satr(char s[ ]);
```

```
void Satr(char s[ ][5]);
```

Qayta juklenwshi funktsiyalar dag'aza qiling'anda kompilyator bul funktsiyalardı ko'rinu oblastların ko'rip shıg'adı. Mısalı, twındı klassta tayanısh klasstag'ı funktsiya ag'za menen birdey attag'ı funktsiya dag'aza etilgen bolsa, ol tayanısh klass funktsiyasın "jasıradı" ha'm kopilyator bul funktsiyalardı qayta juklewdir' ornına qayta anıqlaydı, sebebi olar turli ko'rinu oblastlarına iye. Tap sonday kompilyator basqa ko'rinu oblastlarında anıqlap baradı. Eger funktsiya fayl (baslang'ısh tekst) ko'rinui oblastında dag'aza qiling'an ha'm ol blokta dag'aza etilgen funktsiya menen bir ko'rinu oblastında bolsa ham olar kompilyator ta'repinen qayta juklenbeydi. Funktsiyanın' lokal dag'azası onın' global dag'azasın qaplaydı (jasıradı) mısalı:

```
#include <iostream>
```

```
using namespace std;
```

```
void func(int i) {cout<<"Global dag'aza qiling'an funktsiya\n";}
```

```
void func(char*s) {cout<<"Lokal dag'aza qiling'an funktsiya\n";}
```

```
int main(){
```

```
void func(char*);//Funksiyanin' lokal dag'azasi
```

```
// func(100);// Kompilyatsiya qa'tes, sebebi blokta funktsiya qayta anıqlang'an
```

```

func("\n Qayta aniqlaw ju'z berdi"); //orinli
    return 0;
}

```

Qayta yuklenwshi funktsiya ag'zalar ha'r turli mura'jet spetsifikatori menen dag'aza etilgen boluu mumkin. Olar bir ko'rinw oblastina iye (klass ishinde) ha'm sol sebepli olar kompilyator ta'repinen bir funktsiya dep qaraladi ha'm qayta yuklenedi.

```

#include <iostream>
using namespace std;
class Klass{
    public:

        void klass_Agzasi(double,char*);
        void Qayta_Juklew1();
        void Qayta_Juklew2();
    private:
        void klass_Agzasi(int);
};
void
Klass::klass_Agzasi(double,char*s){cout<<"\ndouble="<<d<<"\t"<<"qatar="<<s;}
void Klass::klass_Agzasi(int i){cout<<"\n int="<<i<<endl;}
void Klass::Qayta_Juklew1(){ klass_Agzasi(10.5,"Qayta Juklew!");}
void Klass::Qayta_Juklew2(){ klass_Agzasi(10);}

int main(){
    Klass klass;
    //sinf.Sinf_Azosi(100); //qate, private-funksiya shaqirilib atir
    Klass.klass_Agzasi(12.5,"qatar manisi"); //O'rinli qayta Juklew
    Klass.Qayta_Juklew1();
    Klass.Qayta_Juklew2();
}

```

```
        return 0;
    }
```

A'melde klasstin' statik funktsiya ag'zalarında qayta juklew mumkin, biraq bul isten mazmunlı jetiskenlikke erisiw mumkin emes.

Konstruktorlardı qayta juklew:

Bazı bir waqıtları qayta juklew mexaniziminde konstruktorlardı qayta juklewde paydalanıladı (destruktordı qayta juklew mumkin emes). Qayta juklew qollanıwdan maqset paydalanıwshıg'a klass wakilinin' (obiektin') turli variantları menen islewge imkanyat beriwdir. Itibar berilgen bolsa, aldında konstruktorlardın' qayta juklew ushrag'an edi, biraq onda qayta juklew haqqında ga'p bolmag'an edi.

Konstruktordı qayta juklewge za'ruryat paydalanıwshıgıa konstruktor ja'rdeminde turlerdi almastırıw imkanın beriw bolıp esaplanadı.

Mısalı, obyektinde Klass klass bolg'an funktsiyanı dag'azası berilgen bolsın:

```
void funk(Klass*ob_k);
```

Bul funktsiyanı basqa turdegi argument penen shaqırıw kompilattsiya qa'tesine alıp keledi, sebebi kompilyator turlerdi qanday turde maslastırıwdı anıqlay almaydı. Kompilyatorg'a qandayda bir T turdi klass turine alıp keliw za'rurligi haqqında ko'rsetpeni beriw ushın klass klasında

```
Klass (T t);
```

Konstruktordı dag'aza qılıw kerek. Bul jagdayda kompilyator

```
funk(t);
```

shaqırıwdı

```
func(Klass(t));
```

mazmunında qabil qiladı ha'm qa'telik juz bermeydi.

```
Class Klass
```

```
{  
    int x;  
    public:  
    Klass();  
    Klass(int); // int turin klass turine o'zgeritiw  
    Klass(long); // long turin klass turine o'zgeritiw  
    Klass(double); // double turin klass turine o'zgeritiw  
    int x_manis();  
};  
Klass::Klass(){x=0;}  
Klass::Klass(int _x){x=_x;}  
Klass::Klass(long _x){x=int(_x);}  
Klass::Klass(double _x){x=_x;}  
int Klass::x_manis(){ return x;}  
void func(Klass & ob_k) { cout<<ob_k.x_manis()<<' \t';}  
  
int main()  
{  
    func(10); func(11L); func(12.15);  
    Klass klass;  
    func(klass);  
    return 0;  
}
```

Da'stur islewi na'tijesinde ekrang'a

10 11 12 0

ma'nisleri shig'adi.

Bul usıldan birar bir berilgendi bir klass turinen ekinshi klass turine otkiziwde paydalanıw mumkin. To'mendegi keltirilgen misal buni ko'rsetedi.

```

#include<iostream>
class Klass1
{
    int x;
public:
    Klass1() {x=5;};
    int manis(){return x;};
};
class Klass2
{
    int z;
public:
    Klass2(){z=10;};
    Klass2(Klass1 & ob_k); // o'zgartiw konstruktor
    int manis(){return z;}
};
Klass2::Klass2(Klass1 & ob1){z=ob1.manis();}
void func(Klass2 & ob2){cout<<ob2.manis()<<endl;}
int main()
{
    Klass1 klass1; func(klass1); // Klass1 turi Klass2 turine o'zgartiledi.
    Klass2 klass2; func(klass2); // Tur o'zgartiwge qa'jet joq.
    return 0; }

```

Sonı esapqa alıw kerek yag'ıy, konstruktorlardı qayta juklew din turli variantların isletiw mumkin, biraq olardıń sapası programma duziwshinin' qolında boladı.

Ulıwma aytqanda, sonı esapqa alıw kerek yag'nıy, paydalanıwshı ta'repinen ko'rsetiletug'ın turlerdi ozgertiw kompilyator ta'repinen to'menegi jag'daylarda g'ana a'mlge asırıladı:

- Obyektlerdi insalizatsiya qilg'anda;
- Funktsiyalardı shaqırıwda;
- Funktsiya manis qaytarıwda;

Turdi o'zgartiw konstruktoridan turlerdi ashıq ra'wishte keltiriwde paydalanıw mumkin:

```
...
void main()
{
    Klass1 klass1;
    Klass2 klass2=(Klass2) klass; // Ashıq ravishte turge keltiriw
    ...
}
```

To'mendegi nusqalaw konstruktorın qayta juklewge misal ketirilgen. Da'sturde to'rtmuyeshlik penen islew ushın klasslar isletiledi.

```
#include<iostream >
Using namespace std;
class Tortmuyeshlik
{
    pubic:
    Tortmuyeshlik();
    Tortmuyeshlik(int,int);
    Tortmuyeshlik(int,int,int,int);
    Tortmuyeshlik(const Tortmuyeshlik &);
    Tortmuyeshlik(const Tortmuyeshlik &,int,int);
    private:
    int x,y,w,h;
};
Tortmuyeshlik:: Tortmuyeshlik(){x=y=w=h=0;}
Tortmuyeshlik:: Tortmuyeshlik(int _x,int _y){x= _x; y=_y;w=h=100;}
Tortmuyeshlik:: Tortmuyeshlik(int _x,int _w,int
_h){x=_x;y=_y;w=_w;h=_h;}
Tortmuyeshlik:: Tortmuyeshlik(const Tortmuyeshlik & tb) {x=tb.x; y=tb.y;
w=tb.w; h=tb.h;}
Tortmuyeshlik:: Tortmuyeshlik(const Tortmuyeshlik &tb, int _x, int _y)
{
    x=_x; y=_y; w=tb.w; h=tb.h;
```

```

}
int main()
{
Tortmuyeshlik tbmuyesh1(5,10,10,100);
Tortmuyeshlik tbmuyesh2(tbmuyesh1,15,200);
Tortmuyeshlik tbmuyesh3(50,50);
Tortmuyeshlik tbmuyesh4(tbmuyesh3);
return 0;
}

```

Bul programmada tortmuyeshlik klassı bir neshe konstrukg'a iye bolıp, to'rtmuyeshlik oblasti jaratıwda ol paydalanıwshı talabına maslasıwshanlıq qa'siyetine iye. Qayta yukleniwshi konstruktorlar variantları ishindegi Tortmuyeshlik(const Tortmuyeshlik &) ha'm Tortmuyeshlik(const Tortmuyeshlik &, int,int) konstrukları **nusqalaw konstruktorları** esaplanadı. Bul imkanyat, qoyılǵ'an misal mazmunınan kelip shıqqan halda kelip shıqqan xalda xa'r turli nusqalaw konstruktorlardı jaratıwda qol keledi.

Sorawlar ha'm tapsırmalar

1. Funktsiyalardı qayta yuklew abzallıǵ'ı nede?
2. Klasstın' funktsiya ag'zaların qayta yuklew sha'rtlerin sanap o'tin'.
3. Funktsiyalardı qayta yuklew ha'm qayta anıqlaw ortasındag'ı parıqtı aytın'.
4. Klass konstruktorların qayta yuklew za'rurligi nede?

Overloading Constructors

Although they perform a unique service, constructors are not much different from other types of functions, and they too can be overloaded. To overload a class's constructor, simply declare the various forms it will take and define each action relative to these forms. For example, the following program declares a class called **timer**, which acts as a countdown timer (such as a darkroom timer). When an object of type **timer** is created, it is given an initial time value. When the **run()** function is called, the timer counts down to zero and then rings the bell. In this example, the constructor has been overloaded to allow the time to be specified in seconds as either an integer or a string, or in minutes and seconds by specifying two integers. This program makes use of the standard library function **clock()**, which returns the number of system clock ticks since the program began running. Its prototype is shown here:

In C++, both local and global variables can be initialized at run time. This process is sometimes referred to as *dynamic initialization*. So far, most initializations that you have seen in this book have used constants. However, under dynamic initialization, a variable can be initialized at run time using any C++ expression valid at the time the variable is declared. This means that you can initialize a variable by using other variables and/or function calls, so long as the overall expression has meaning when the declaration is encountered. For example, the following are all perfectly valid variable initializations in C++:

33-Leksiya. Operatorlardı qayta juklew

Joba:

1. Operatorlardı qayta ju'klew tusinigi
2. Binar operatorlardı qayta ju'klew
3. Salıstırıw ha'm logikalıq operatorlardı qayta ju'klew
4. Ma'nis beriw operatorın qayta ju'klew

Operatorlardı qayta ju'klew tusinigi; C++ tilinde ornatılğan operatorlardı qayta ju'klew imkanıyatı bar.

Operatorlar global ta'rizde yamasa klass shegarasında qayta ju'kleniwı mumkin. Qayta ju'klew operator gılt so'zi menen funktsiya korinisinde a'melge asırıladı. Qayta ju'kleniwshi funktsiya operator-funktsiya dep ataladı ha'm atı operatorX ko'rinisinde bolıwı kerek, bul jerde X-qayta ju'kleniwshi operator. C++ tilinde qayta ju'kleniwı operatorın qayta ju'klew ushın operator+ atlı funktsiyasın anıqlaw kerek boladı. Eger qosıw ma'nis beriw a'meli menen kelgen jagdayın qayta ju'klew ushın operator+=ko'rinisinde funktsiya anıqlaw za'rur boladı. A'dette kompilyator da'stur kodında qayta ju'klengen operatorlar ushrag'anda olardı ayqın emes ta'rizde qollaydı. Za'ru'r bolganda ayqın shaqırıw mumkin:

```
Noqat noqat1, noqat2, noqat3;
```

```
//Qayta ju'klengen qosıw operatorın ayqın ta'rizde shaqırıw  
noqat3= noqat1.operator( noqat2) ;
```

Keste. Qayta ju'kleniwshi operatorlar

Operator	Sıpatlama	Da'rejesi
,	u'tir	binar
!	logikalıq biykar	Unar
!=	ten' emes	binar
%	qaldıqlı bo'liw	binar
%=	modullı bo'liw ma'nis beriw menen	binar
&	razryadlı ha'm	binar
&	adresti alıw	unar
&&	logikalıq ha'm	binar
&=	razryadlı ha'm ma'nis beriw menen	binar
()	funksiyanı ıñakırıw	-
*	ko'beytiw	binar
*	mu'rajat quralı	binar
*=	ko'beytiw ma'nis beriw menen	binar

+	qosiw	binar
+	unar qosiw	unar
++	inkrement	unar
+=	qosiw ma'nis beriw menen	binar
-	ayırma	binar
-	unar minus	unar
--	dekrement	unar
-=	ayırma ma'nis beriw menen	binar
->	elementti tan'law	binar
->*	elementti ko'rsetkish arqalı tan'law	binar
/	bo'liw	binar
/=	bo'liw ma'nis beriw menen	binar
<	kishi	binar
<=	kishi yamasa ten'	binar
<<	razryad boyınsha shepke jiljitiw	binar
<<=	on'g'a jiljitiw ma'nis beriw menen	binar
=	ma'nis beriw	binar
==	ten'	binar
>	u'lken	binar
>=	u'lken yamasa ten'	binar
>>	razryad boyınsha on'g'a jiljitiw	binar
>>=	on'g'a jiljitiw ma'nis beriw menen	binar
[]	massiv indeksi	-
^	razryadlı ayırıqsha jag'dayı yamasa	binar
^=	razryadlı ayırıqsha jag'dayı yamasa ma'nis beriw menen	binar
	razryadlı yamasa	binar
=	razryadlı yamasa ma'nis beriw menen	binar
	logikalıq yamasa	binar
~	bitli logikalıq biykar	binar
Delete	dinamik o'bektti joq qılıw	-
New	dinamik o'bektti jaratıw	-

6.2-kestesinde qayta ju'klenbeytug'in operatorlar dizimi keltirilgen.

6.2-keste. Qayta ju'klenbeytug'in operatorlar

Operatorlar	sıpatlaması
.	ag'zanı tan'law
::	ko'riniw oblastına ruqsat beriw operatori
.*	ko'rsetkish boyınsha ag'zanı tan'law

?:	sha'rt a'meli
#,##	preprotssessor belgileri

Qayta ju'klenetug'in operatorlardin' operator funktsiyaları,new xa'm delete operatorlarinan,to'mendegilerge boysınıwı kerek:

1) operator-funktsiya klasının' statik bolmag'an funktsiya-ag'zası bolıwı kerek yamasa operator-funktsiya klass yamasa sanap o'tiletug'in tu'rge kiriwshi(int) tu'rdegi argument qabil qılınıwı mumkin yamasa operator-funktsiya klass sonday sanap o'tiletug'in tu'rge kiriwshi yamasa adres alıw a'meli (&)bolg'an argumentlerdi qabil qılınıw mumkin.

Mısalı,

```
class Noqat
```

```
{
```

```
public:
```

```
Noqat operator<(Noqat&);//<'operatori ushin opertator –funktsiya
dag'aza qiliw
```

```
...
```

```
//Qosiw opertatorlarin dag'aza qiliw
```

```
friend Noqat operator+(Noqat&,int);
```

```
friend Noqat operator+(int,Noqat&);};
```

Bul mısalda '<'operator klasının' funktsiya-ag'zası sıpatında dag'aza qiling'an, qosıw operatorı bolsa klastın' dostı sıpatında dag'aza qiling'an xa'm ol bir operatordı qayta ju'klewden bir neshe variantı mumkinligin ko'rsetedi;

2)operator–funktsiya operatorının' argumentler (operandlar) sanın,

olardın' ustinligi xa'm orınlanıw ta'rtibin o'zgerge almaydı;

3)klass funktsiya ag'zası sıpatında dag'aza qiling'an unar operatorının' operator-funktsiyası parametrge iye bolmawı kerek.Eger operator-funktsiya global funktsiya bolsa,ol tek bir parametrge iye boladı;

4) klass funktsiya ag'zasi sıpatında dag'aza qiling'an binar operatorının' operator-funktsiyası bir parametrge iye bolıwı kerek. Eger operator-funktsiya global funktsiya bolsa,ol tek eki parametrge iye boladı;

5) operator-funktsiya kelisiw boyınsha ma'nis qabıl qılıwshı parametrge iye bolmawı kerek;

6) klass funktsiya ag'zasi sıpatında dag'aza qiling'an operator funktsiyasının' birinshi parametri (eger ol bar bolsa)klass tu'rinde bolıwı kerek.Sebebi,bul funktsiya o'bektı ushın usı operator shaqırıladı.Birinshi argument ustinde hesh qanday tu'rge keltiriw a'meli orınlanbawı kerek;

7) ma'nis beriw operatorının' operator-funktsiyasınan tısqarı barlıq funktsiyalar miyraslıq penen o'tedi;

8)'=',"()","[]", ha'm''->' operatorının' operator-funktsiyaları klastın' static bolmag'an funktsiya ag'zaları bolıwı kerek(ha'm olar global funktsiya bola almaydı).

Operatorlardı qayta ju'klew arqalı,klass shegarasında operatordın' ahmiyeti tupten o'zgerip jiberiwi mumkin.Biraq bul jumıstı zaru'r bolg'anda g'ana a'melge asırg'an maqul.Bolmasa orınlanatug'in a'mellerdi mazmunlıq qatelikler juzege keliwi mumkin.

Binar operatorlardı qayta ju'klew: Binar operatorının' operator-funktsiyası klastın' statik bolmag'an funktsiya ag'zasi dag'aza qiling'anda to'mendegi sintaksiske iye bolıwı kerek:

< qaytarılatug'in ma'nis tu'rge > operator X(<parametr tu'rge >< parametr >);

Bul jerde -< qaytarılatug'in ma'nis tu'rge >funktsiya qaytaratug'in ma'nis tu'rge,X- qayta ju'klenetug'in operator, < parametr tu'rge >- funktsiya parametri. Funktsiya parametri operatordın' on' ta'repindegi o'bekt uzatıladı, operatordın' shep ta'repindegi o'bekt bolsa ayqın emes ta'rizde this ko'rsetkishi menen uzatıladı. Eger operator-funktsiya global dep dag'aza qılınsa,ol to'mendegi ko'riniske iye boladı.

< qaytarılatug'in ma'nis tu'rge > operator X(<parametr tu'rge 1> < parametr tu'rge 1>< parametr tu'rge 2>,< parametr tu'rge 2>);

Bul jerde funktsiya parametrlerinin' keminde birewi operator qayta ju'klenetug'in klass tu'rgende bolıwı kerek. Sonda da , qaytaratug'in ma'nis tu'rine hesh qanday sheklew bolmasada, a'dette ol klass tu'rgede yamasa klasqa ko'rsetkish boladı.

Operator funktsiyalardı jazıwdın' bir neshe mısalların keltiremiz. Bul mısallar operatorlardı qayta ju'klewdin tolıq imkaniyatların ashıp bermese de, onın' ahıyetli ta'replerin ko'rsetedi. Birinshi na'wbette operator funktsiyalarınin' klastın' funktsiya-ag'zası korinisinde anıqlanıwın ko'remiz. To'mendegi da'sturde noqat klası ushın (+) ha'm ayırma (-) operatorların qayta ju'klew a'melge asırılǵ'an.

```
#include <iostream>

using namespace std;

class Noqat{

int x,y;

public:

Noqat (){

x=0;y=0;}

Noqat(int _x,int _y){x=_x; y=_y;}

void Noqat_Manisi(int&_x,int&_y){_x=x;_y=y;}

Noqat operator+(Noqat& ob);

Noqat operator-(Noqat& ob);};

Noqat Noqat ::operator+(Noqat& ob){

Noqat AraliqOb;

AraliqOb.x=x+ob.x;

AraliqOb.y=y+ob.y;

return AraliqOb;}

Noqat Noqat ::operator-(Noqat& ob){
```

```

Noqat AraliqOb;

AraliqOb.x=x-ob.x;

AraliqOb.y=y-ob.y;

return AraliqOb;}

int main(){

int x,y;

Noqat A(100,200),B(50,100),C;

C=A+B; //Qayta yuklengen qosiw operatori amel qiladi.

C.Noqat_Manisi(x,y);

cout<<"C=A+B amel natijyesi"<<"C.x="<<x<<"C.y="<<y<<endl;

A=A-B;// Qayta yuklengen ayiriw operatori amel qiladi.

A.Noqat_Manisi(x,y);

cout<<"A=A-B amel natijyesi " <<"A.x="<<x<<"A.y="<<y<<endl;

return 0;}

```

Da'stur islewi natijesinde ekrang'a to'mendegi korinistegi natijeler shig'adi

C=A+B amel natijyesi:C.x=150 C.y=300

A=A-B amel natijyesi:A.x=50 A.y=100

Da'sturdegi sol na'rsege itibar beriw kerek, operator-funktsiya parametri '

klass o'bekti adresti aliw ko'rinisinde aniqlang'an.Uliwma alg'anda,argument sypatında o'bektin' o'zında shaqırıw mumkin,biraq funktsiyadan shig'ıwda bul o'bekt destruktur ja'rdeminde joq qılınadı.Funktsiya parametri klass o'bektne adresti aliw ko'rinisinde bolıwının' abzalıg'ı sonda,funktsiya shaqırılğ'anda og'an o'bekt emes,ba'lkim o'bektke ko'rsetkish uzatıladı ha'm klass nusqası ushın shaqırılaturğ'ın destruktur og'an ta'sir qılmaydı.Operotor-funktsiyalardıń qaytarıwshı ma'nis usı klass twrinde bolıp,bul jag'day o'bektlerdi

salıstırg'anda quramalı an'latpalarda kollaw imkanın beredi. Misalı, to'mendegi a'meller da'stur ushın ruxsat etilgen til ko'rsetpesi esaplanadı:

$S=A+V-S;$

Ekinshi ta'repten, to'mendegi an'latpa da orınlı:

$(A+V).Noqat_Manisi(x,y);$

Bul an'lanpa qosıw operatorının' operator-funttsiyasındag'ı waqtınsha ob'ektin Noqat_Manisi() funttsiyası qollanıladı.

Keyingi misal operator-funttsiyası parametri sıpatında barlıq turdegi berilgen jag'dayın ko'rsetedi. Bul berilgen operatorдын' on' ta'repinde kelgenine itibar beriw kerek.

```
#include <iostream>

using namespace std;

class Noqat{

int x,y;

public:

Noqat(){x=0; y=0;}

Noqat(int _x,int _y){x=_x; y=_y;}

void Noqat_Manisi(int&_x,int&_y){_x=x; _y=y;}

Noqat operator+(Noqat& ob);

Noqat operator+(int n);};

Noqat Noqat::operator+(Noqat& ob){

Noqat AraliqOb;

AraliqOb.x=x+ob.x;

AraliqOb.y=y+ob.y;
```

```

return AraliqOb;}

Noqat Noqat::operator+(int n){

Noqat AraliqOb;

AraliqOb.x=x+n;

AraliqOb.y=y+n;

return AraliqOb;}

int main(){

int x,y;

Noqat A(100,200),B(50,100),C;

C=A+B;//parametric klass turidagi obyekt bolgan qayta yuklengen qosiw
operatori a'mel qiladi,

C.Noqat_Manisi(x,y);

cout<<" C=A+B a'meli natijesi:"<<"C.x="<<x<<"C.y="<<y<<endl;

C=A+30; //parametric sanap o'qilgan turidagi obyekt bolgan qayta
//yuklengen qosiw operator a'mel qiladi.

C.Noqat_Manisi(x,y);

cout<<"C=A+30 a'mel natijesi:"<<"C.x="<<x<<"C.y="<<y<<endl;

return 0;} Da'stur islewi natijesinde ekrang'a to'mendegi ko'rinistegi
to'mendegi natijeler shig'ariladi.

C=A+B a'meli natijesi: S.x=150 S.u=300

C=A+30 a'meli natijesi: S.x=130 S.u=230

Operator-funktsiya parametri operatorin' on' ta'repidagi operand
ekenligi sebepli kompilyator to'mendegi ko'rsetpelerdi duris "tusinedi":

C=A+30;

Biraq kompilyator

```

$C=30+A;$

qo'rsatpesin qabil qilmaydi.

Bul mashqalani operator-funktsiyanin' "ishki" imkaniyatlarini menen sheship bolmaydi. Mashqalados operator-funktsiyalardan paydalanish arqali sheshish mumkin. Belgili, dos funktsiyalarga jasiring'an this ko'rsatkishi uzatilmaydi. Sonin' ushyn binar operator-funktsiyasi eki argumentke iye bolishi kerek-birinshisi shep operand ushyn, ekinshisi on' operand ushyn.

```
#include <iostream.h>

class Noqat{

int x,y;

public:

Noqat(){x=0; y=0;}

Noqat(int _x,int _y){x=_x; y=_y;}

void Noqat_Manisi(int& _x,int& _y){_x=x; _y=y;}

friend class Noqat operator+(Noqat& ob1, Noqat& ob2);

friend class Noqat operator+(Noqat& ob1, int n);

friend class Noqat operator+(int n, Noqat& ob);

};

Noqat operator+(Noqat& ob1,Noqat& ob2){

Noqat AraliqOb;

AraliqOb.x=ob1.x+ob2.x;

AraliqOb.y=ob1.y+ob2.y;

return AraliqOb;}

Noqat operator+(Noqat& ob,int n){
```

```

Noqat AraliqOb;

AraliqOb.x=ob.x+n;

AraliqOb.y=ob.y+n;

return AraliqOb;}

Noqat operator+(int n,Noqat& ob){

Noqat AraliqOb;

AraliqOb.x=ob.x+n;

AraliqOb.y=ob.y+n;

return AraliqOb;}

int main(){

int x,y;

Noqat A(100,200),B(50,100),C;

C=A+B;

C.Noqat_Manisi(x,y);

cout<<" C=A+B a'meli natijjesi:"<<"C.x="<<x<<"C.y="<<y<<endl;

C=A+30;

C.Noqat_Manisi(x,y);

cout<<"C=A+30:"<<"C.x="<<x<<"C.y="<<y<<endl;

C=30+A;

C.Noqat_Manisi(x,y);

cout<<"C=30+A:"<<"C.x="<<x<<"C.y="<<y<<endl;

return 0;}

```

dos funkttsilardı qayta ju'klew esabına

C=A+30;

C=30+A;

til ko'rsatpelerin orinlaw imkaniyatı juze keledi.

Salıstırıw ha'm logikalıq operatorlardı qayta ju'klew: Salıstırıw ha'm logikalıq operatorları,sonda da binar operatorlar bolsada olar jeke turde qaraladı.Sebebi olarg'a saykes keliwshi operator-funktsiyalar o'zleri anıqlang'an klass tu'rgendegi ma'nisin emes,ba'lki logikalıq ma'nislerdi qaytarıwı kerek(yamasa ,true ha'm false sıpatında qabıl qılınıwshı putin san ma'nisin).Mısalı "==" ha'm "&&" operatorların qayta ju'kleweniwin ko'remiz.

```
#include <iostream>
```

```
using namespace std;
```

```
class Noqat{
```

```
int x,y;
```

```
public:
```

```
Noqat(int _x,int _y){x=_x;y=_y;}
```

```
Noqat(){x=0; y=0;}
```

```
void Manis_xy(int&_x,int&_y){_x=x; _y=y;}
```

```
bool operator==(Noqat ob);
```

```
bool operator&&(Noqat ob);
```

```
};
```

```
bool Noqat::operator==(Noqat ob){return(x==ob.x && y==ob.y);}
```

```
bool Noqat::operator&&(Noqat ob){return(x&&ob.x) && (y&&ob.y);}
```

```
int main(){
```

```
Noqat noqat1(10,20),noqat2(10,25),noqat3(10,20),noqat4;
```

```
if(noqat1==noqat2) cout<<"noqat1 ham noqat2 oz ara ten\n";
```

```

else cout<<"noqat1 ham noqat2 oz ara ten emes\n";

if(noqat1==noqat3) cout<<"noqat1 ham noqat3 oz ara ten\n";

else cout<<"noqat1 ham noqat3 oz ara ten emes\n";

if(noqat1&&noqat2) cout<<"noqa1&& noqat2 ras\n";

else cout<<"noqat1 && noqat2 jalgan\n";

if(noqat1 &&noqat4) cout<<"noqat1 && noqat4 ras\n";

else cout<<"noqat1 && noqat4 jalgan\n";

return 0;

} Da'stur islewi natiyjesinde ekrang'a to'mendegi korinistegi natiyjeler
shig'adi

```

noqat1 ham noqat2 oz ara ten emes

noqat1 ham noqat3 oz ara ten

noqa1&& noqat2 ras

noqat1 && noqat4 jalgan

Operatorlardı qayta ju'klew arqalı koordinata noqatları arasındag'ı jan'a mazmundag'ı munasebetler anıqlanadı.

Ma'nis beriw operatorın qayta ju'klew: Ma'nis beriw operatorı da binar operator esaplanadı, biraq onı qayta ju'klew o'zine ta'n qa'siyetlerge iye:

-ma'nis beriw operatorının' operator-funktsiyası global ta'rizde dag'aza qılınıwı mumkin emes, yagnıy ol tek klastın' funktsiya ag'zası bolıwı kerek;

- ma'nis beriw operatorının' operator-funktsiyası tuwindı klasqa miyraslıq penen o'tpeydi;

-kompilyator ma'nis beriw operatorının' operator-funktsiyasın payda qılıw mumkin, eger ol klasta anıqlanbag'an bolsa.

Kompilyator ta'repinen kelisiw boyınsha payda qiling'an ma'nis beriw operatorı klastın' ha'r bir statik bolmag'an ag'zaları ma'nis beriw a'melin

orinlaydi. Biraq, eger klass ko'rsetkishleri bolatug'in bolsa, olarg'a bunday ma'nis beriw operatori qollanilmaydi. Soni aytip o'tiw kerek, ma'nis beriw operatori orinlang'annan keyin shep ta'repindegi openrand o'zgeredi, sebebi og'an jan'a ma'nis beriledi. Sol sebepli

ma'nis beriw operatorinin' operator-funktsiyasi oni shaqirg'an o'bektke ko'rsetkishin qaytarıwı shart. Bunun' ushin funktsiya ayqın emes ta'rizde klass funktsiyalarına birinshi parametr sıpatında uzatılğ'an this ko'rsetkishin qaytarıwı jeterli. O'z nawbetinde funktsiyanın' this ko'rsetkishin qaytarıwı to'mendegi ko'rinistegi ma'nis beriw a'mellerin "tusiniw" imkanın beredi:

To'mendegi misal ma'nis beriw operatorın qayta ju'klewdi ko'rsetedi:

```
#include <iostream>

using namespace std;

class Noqat {
    int x,y;
    public:
        Noqat (int _x,int _y){x=_x; y=_y; }
        Noqat(){x=0;y=0;}
        void manis_xy(int &_x, int &_y){_x=x; _y=y;}
        bool operator==( Noqat ob);
        Noqat &operator=(Noqat &ob);
};

bool Noqat::operator==(Noqat ob){return (x==ob.x && y==ob.y);
}

Noqat & Noqat::operator  =(Noqat &ob){
    if(this==&ob)    return *this;
    x=ob.x;
```

```

        y=ob.y;
        return *this;
    }
int main(){
    int a,b;
    Noqat noqat1(10,20),noqat2(20,25),noqat3;
    noqat3=noqat2;
    if(noqat2==noqat3)
        cout<<"noqat2 noqat3 ke oz ara ten \n";
    else
        cout<<"noqat2 noqat3 ke oz ara ten emes. \n";
    noqat3=noqat2=noqat1;
    if(noqat1==noqat3)
        cout<<"noqat1 noqat3 oz ara ten. \n";
    else
        cout<<"noqat1 noqat3 ke ten emes. \n";
    noqat3.manis_xy(a,b);
    cout<<"noqat3.x="<<a<<"noqat3.y="<<b<<endl;
    return 0;}

```

Da'stur islewi natijesinde ekrang'a

noqat2 noqat3 ke oz ara ten

noqat1 noqat3 oz ara ten

noqat3.x=10 noqat3.y=20

xabarları shıg'ariladı.

Ma'nis beriw a'melin qayta ju'klewde mazmunlıq qateliğe alıp kelinetug'ın noqat1= noqat1;

ko'rinistegi o'zin o'zine ju'klew jag'dayı ayrıqsha bahalaw kerek.

Sebebi ma'nis beriw operatori orinlanıwında aldın shep ta'reptegi operand yadı tazalanadı ha'm keyin sol oring'a on' ta'reptegi operandın' haqıyqatqa tuwrı kelmeytug'ın ma'nisin jaylastırıwjuzege keledi. Sol sebepli, joqarıdag'ı mısalda operator=() funksiyası

```
if(this==&ob) return *this;
```

bahalaw ko'rsetpesine iye ha'm ol qa'telik bolıwına jol qoymaydı.

Funksiyalardı shaqırıw operatorı qayta ju'klew: Qawsırmalar arqalı a'melge asırilatug'ın funksiyanı shaqırıw operatorı sintaksiske iye bolg'an binar operator esaplanadı:

<anlatpa>(< an'latpalar dizimi >).

Bul jerde <anlatpa>-birinshi operand ha'mde < an'latpalar dizimi > majburiy bolmag'an ekinshi operand esaplanadı. Funksiyanı shaqırıw operatorın' operator-funksiyası klasstın' statik bolmag'an funksiya-ag'za ko'rinisinde dag'aza qılınıwı kerek. Funksiyanı shaqırıw operatorın qayta ju'klewge za'rurlik, a'dette ko'p parametrdi talap qılatug'ın a'mellerdi orinlag'anda juzege keledi. Funksiyanı qayta ju'klew operatorı qayta ju'klengende, ol tek qawısishindegi o'zi dag'aza qılın' klass obiyektlerine bolg'an murajetti o'zgerterdi, biraq funksiya shaqırılıw protsesine ta'sir qılmaydı. Funksiyanı shaqırıw operatorı qayta ju'klewge mısal ko'reyik:

```
#include <iostream.h>
```

```
class Noqat{
```

```
int x,y;
```

```
public:
```

```
Noqat(int _x,int _y){x=_x; y=_y;}
```

```

Noqat(){x=0; y=0;}

Manis_xy(int&_x, int&_y){_x=x; _y=y;}

Noqat &operator()(int dx,dy)
{
x+=dx; y+=dy;
return *this;}};

int main(){
int x,y;
Noqat n1,n2;
n2=n1(5,10);//operator funktsiyani shaqiriw
n2.Manisi_xy(x,y);
cout<<"1-shaqiriwda:N2.x="<<x<<"N2.y="<<y<<endl;
n2=n2(1,2);//qayta ju'klengen funktsiyani shaqiriw operatori
n2.Manis_xy(x,y);
cout<<"2-shaqiriwda :N2.x="<<x<<"N2.y="<<y<<endl;
return 0;}

```

da'stur islewi natijesinde ekrang'a to'mendegi qatarlar shıg'adı:

1-shaqiriwda: n2.x=5 n2.y=10

2-shaqiriwda:n2.x=6 n2.y=12

Keltirilgen misal ushın << Noqat n1(5,10);>>ha'm <<n1(5,10);>> an'latpalardı o'z ara almasırmaw kerek.Birinshi an'latpa klastın' parametrli konstruktorına murajetti an'latsa ,ekinshisi-qayta ju'klengen operator funktsiyanı shaqırıw kerekligin bildiredi.Misal sonni ko'rsetedi, funktsiyanı shaqırıw operatori qayta ju'klew a'melge asırılǵ'an klass obyektine funktsiya arqalı murajet qılıw mumkin.

Klas ag'zalarina murajet operatorlarin qayta ju'klew: Klas ag'zalarina murajet operatorlarin qayta ju'klew ag'zalarina murajet operatorlarin("->") qayta ju'klew arqali a'melge asiriladi('.' Operatorlari qayta ju'klenbeydi). Bul operator unar esaplanadi ha'm onin' operator funktsiyasi klastin' statik bolmag'an ag'zasi qilip dag'aza qilinivi kerek. Saykes operator funktsiyasinin' ko'rinisi to'mendegishe:

```
<klass atı> * operator ->{<til ko'rsetpeleri>;
```

Bul jerde <klass atı>- operator tiyisli klass atı.Klas ag'zalarin tan'law operatorin qayta ju'klew,a'dette qollonilivi usi waqıtta orinli yamasa orinli emesligin baqlawshı "ongli" ko'rsetkishlerdi a'melge asiriwda qollaniladi. Klas ag'zalarin tannaw operatorı qayta ju'klewge misal keltiremiz:

```
#include <iostream.h>
```

```
class Noqat{
```

```
int x,y;
```

```
public:
```

```
Noqat(int _x,int _y){x=_x; y=_y;}
```

```
Noqat(){x=0; y=0;}
```

```
Manisi_X(){return x;}
```

```
Manisi_Y(){return y;}
```

```
Noqat &operator->();};
```

```
Noqat *Noqat::operator->(){
```

```
cout<<"Obyekt elementine murajet:"
```

```
return this;}
```

```
int main(){
```

```
Noqat n(5,10);
```

```
//Murajet operatorinin operator funktsiyasin shaqiriw
```

```
cout<<"n->x="<<n->Manisi_X()<<endl;
```

```
cout<<"n->y="<<n->Manisi_Y()<<endl;
return 0;}
```

Da'stur islewi natiyjesinde ekrang'a

Obyekt elementine murajet: n->x=5

Obyekt elementine murajet: n->y=10

Kompilyator ta'repinen<<n->Manisi_X()>>ko'rsetpesin

(n.operator->())-> Manisi_X()

korinisinde talqın qılınıwshı operator funktsiyanı qanday orınlawın ishki moxiyatın ashıp beredi.

new ha'm delete operatorların qayta ju'klew: Yadıdı dinamik ta'rizde ajratıw ha'm tazalaw (kaytarıw) wazıypasın a'melge asırıwshı new ha'm delete operatorların orınlang'anda saykes ta'rizde standart anıqlang'an operator new() (yamasa operator new[]()-massiv ushın qollanılg'anda) ha'm delete() (yamasa operator delete []()-massiv ushın qollanılg'anda) arnawlı funktsiyalardı da shaqırıladı. Bunnan keyin ayırıqsha za'rurlik bolmasa, bul funktsiyalardıń massiv variantı qaralmaydı ha'm birewi ushın aytılg'an fikirler ekinshisi ushın da orınlı dep esaplanadı. Ulıwma alg'anda, new ha'm delete operatorları eki turli variantı da qayta ju'kleniwı mumkin:

```
::operator new() // Global (standart);
```

```
::operator delete() // Global (standart);
```

```
<klass ati>::operator new() //klass funktsiyası
```

```
<klass ati>::operator delete() // klass funktsiyası
```

Klasta anıqlang'an operator new()operator- funktsiyası klastın' statik ag'za sı bolıp, ol klass obiyektleri ushın global:: operator new() funktsiyasın jasıradı.

Global :: operator new()-operator funktsiyasının' ozi de qayta ju'kleniwshi (klastan tısqarı) ha'm qayta ju'kleniwshi funktsiyalardıń turli prototipi ne iye bir neshe variantları bolıwı mumkin. Paydalanıwshı ta'repinen anıqlanatuǵın new operatorının' operator-

Funktsiyası `void*` ma'nisin kaytarıw kerek ha'm birinshi parametr sıpatında `size_t` tu'rgendegi parametrge iye bolıwı kerek. Aqırǵı parametr `<stddef.h>` sarlavxa faylında `unsigned int` ko'rinisinde anıqlang'an. `new` operatorın

qayta ju'klew prototiptan paydalanıladı:

```
void * operator new(size_t size); //bir obyekt
```

```
void * operator new[](size_t size); //obyektler massivi
```

Bul prototiplar `<new.h>` sarlavxa faylında jaylasqan. Sol sebepli, eger

`new` ha'm `delete` operatorların qayta ju'klew za'rur bolg'anda bul fayıldı da'stu'rge qosıw kerek boladı.

Sog'an itibar beriw kerek, `new` operatorlar-funktsiyadag'ı `size` arametrinin' baytlardag'ı o'lishemin(`sizeof(size)`), yag'nıy yadta ajratıw kerek soxanın baytlardag'ı o'lishemin operatorlar-funktsiya ushın kompilyatordın' o'zi esaplap beredi. Operator funktsiyag'a bul parametrıdın' qoyılıw sebebi sonnda, tuwındı klaslar operator `new()` ha'm operator `new []()` funktsiyaların miyraslıq penen o'te aladı ha'm tuwındı klass obyektlerinin' o'lishemi tayansh klass obyektleri o'lisheminen parq qılıw mumkin.

Eger yadtag'ı aldın ajratılǵan orındı qaytadan "taqsimlash" za'rur bolsa, `new` operatorının' qosımsha parametrge iye bolg'an jaylastırıw formasınan paydalanıw mumkin. Saykes operator-funktsiya sintaksisi to'mendegi ko'riniske iye:

```
void operator new(size_t size, void*p); // obyektler ushin
```

```
void* operator new(size_t Type_size, void*p); // obyektler massivi ushin
```

A'dette `new` operatorının' jaylastırıw formasınan global obyektler ushın ,tu'rge keltiriw a'meli orınlang'an halda qollanıladı. Bul variantı absolyut adres boyınsha aldınnan ajratılǵan orıng'a obyektı jaylastırıw a'melge asırılıdı. Ko'rsetilgen adres uyumdan bolıwı shart emes. Jaylastırılǵan obyekt o'lishemi ko'rsetilgen oblastta o'lisheminen kishi bolg'an jag'daylarda ajratılǵan oblasttı korrekt ta'rizde tazalaw paydalanıwshı zimmasiga yuklatılıdı ,sebebi `delete` operatorı durıs islewine isenim joq.

Qayta ju'klengen new operatorin shaqiriw sintaksisi to'mendegishe

```
<:>new<tur uzunlig'1 ><tur atı><(<insalizattsiyalaw>)>.
```

yaki

```
<:>new<tur uzunlig'1 >(<tur atı>)<(<insalizattsiyalaw>)>.
```

Bul jerde <:>-shart bolmag'an,ko'rinis oblastina ruxsat beriw operatori;

<tur uzunlig'1 >-operator new() funktsiyasının' size parametri,
ko'rsetilmegen-

ligi mumkin; < tur atı >-yadatan ajratilg'an berilgennin' twri;

<(< insalizattsiyalaw >)>-< tur atı > twrinin' konstruktori ushin
uzatilatug'in baslang'ish ma'nisler dizimi(dizim ko'rsetilmegen bolıwı
mumkin).

Sintaksis sonni ko'rsetedi,new operatori obyektke baslang'ish ma'nis beriw
beriw menen shaqiriw mumkin.Biraq bul operatorlar massivı ushin
shaqirg'anda insalizattsiyanı qollanıp bolman'dı ha'm obyektlerdin'
baslang'ish ma'nisleri belgisiz boladı.

Global new ha'm delete operatorların qayta anıqlaw ha'm qayta ju'klewge
mısal keltiremiz:

```
void* operator new(size_t size)//Global new operatorin qayta anıqlaw{
```

```
cout<<"yadatan"<<size<<"bayt ajratiwga soraw boldi\n";
```

```
return malloc(size);}
```

```
void operator delete(void *p){
```

```
cout<<"yadti bosatiw!\n";
```

```
free(p);}
```

```
// Global new operatorini qayta juklew
```

```
void*operator new(size_t size,char * fname,int line)
```

```
{
```

```

cout<<fname<<"fayling"<<line<<"-qatarinda\n";

cout<<size<<"bayt ajratiwga soraw boldi!\n";

return malloc(size);}

int main(){
char fayl_ati[]="new_quyk.cpp";
int qatar=5;
long*plnt=new long;//Global new operatorin shaqiriw
delete plnt;//Global delete operatorin shaqiriw
plnt=new(fayl_ati,qatar) long;//qayta juklengen new operatorin shaqiriw
delete plnt;//Global delete operatorin shaqiriw
return 0;}

```

Da'stur islewi natijesinde ekrang'a to'mendegi qatarlar shig'adi
Yadta 4 bayt ajartiwga soraw boldi
yad bosatildi!
new_quyk.cpp faylinin 5-qatarinda
4 bayt ajratiwga so'rov boldi!
Yad bosatildi!

Nawbettegi misalda global new operatorinin' jaylasıwı formasınan
paydalanıw mumkin.

```

// Global new operatorin qayta aniqlaw
void*operatori new(size_t size,void*krst){
cout<<"obyekt korsetilgen adreske jaylastirildi\n";
return krst;}
void Sestemani_tekseriw(){ cout<<"Sestema normal islep atir\n";
}
class Noqat{
int x,y;
public:
Noqat(int_x, int_y){
x=_x; y=_y; cout<<" obyektler berilgen manisler menen
insalizatsiyalandi\n";
cout<<"x="<<x<<"y="<<y<<\n;}
Noqat(){
{ x=0; y=0;
cout<< "Obyektler kelisiw boyinsha insalizatsiyalandi\n";
cout<<"x="<<x<<"y="<<\n;}

```

```

~Noqat(){cout<<"noqat ::~Noqat() isledi\n";}
};
Noqat noqat;//Global obyekt
int main(){
Sestemani_tekseriw();// qandayda bir jumisti orinlaw
Noqat *krst=new(noqat)Noqat(10,20);
// Endi obyekttni insalizatsiyalaw mumkin!
krst->Noqat::~~Noqat();
return 0;}

```

Da'stur natijesi to'mendegi xabarlar boladi:
Obyektler kelisiw boyinsha insalizatsiyalandi
x=0 y=0
Sistema normal islep atir
Obyekt korsetilgen adreske jaylastirildi

obyektler berilgen manisler menen insalizatsiyalandi

x=10 y=20

Noqat::~~Noqat() isledi

Noqat::~~Noqat() isledi

Keyingi da'stur lokal new operatorin qayta ju'klewdi klass shinde
Jaylastiriw formasinan paydalang'an halda a'melge asiriwg'a misal:

```

class Qatar{
union{char ch; char buf[81];};
public:
Qatar(char c='\0'): ch(c){cout<<"qatar klasstin belgili konstruktori\n";}
Qatar(char * s){cout<<"Qatar klasstin qatarli konstruktori\n";
strcpy(buf,s);}
~Qatar(){ cout<<"qatar::~~Qatar()"<<endl;}
//new operatorinin' jaylasiwshi sintaksisi
void* operator new(size_t,void*buffer){return buffer;}};

```

```

char qatar_buffer[sizeof(Qatar)]; //yad bufferi
int main(){
Qatar*krst=new(qatar_buffer) Qatar("C++"); // qatardi bufferga jaylastiriw
cout<<"qatar_buffer"<<qatar_buffer<<endl;
krst->Qatar::~Qatar(); //Destruktordi ashiq tarzide shaqiriw
krst=new(qatar_buffer)Qatar('c'); //'c' belgisi qatar basina jaylastiriw
cout<<"qatar_buffer[0]="<<qatar_buffer[0]<<endl;
cout<<"bufferdin' jan'a manisi"<<endl;
cout<<"qatar_buffer="<<qatar_buffer<<endl;
krst->Qatar::Qatar(); //Destruktordi ayqin tarzide shaqiriw
return 0;}

```

Da'sturde global obyekt qatar_buffer qatarı jaratıladı ha'm sol yad oblastına Qatar klası obyektı "C++" ma'nisi menen jaylastırıladı. Ashıq ta'rizde klass destruktordı shaqırıw arqalı klass obyektı jog'aladı. Keyin, usı adreste Qatar klasının' ekinshi obyektı 'c' belgisi menen jaratıladı ,yadtın' qatar_buffer adresli oblastında qatar ma'nisi ko'rsetiledi ha'm klass obyektı joq qılınadı. Da'stur islewi natiyjesinde ekrang'a Qatar klasstin qatarli konstruktordı

```
qatar_buffer=C++
```

```
Qatar::~Qatar()
```

Qatar klasinin' belgili konstruktordı

```
Qatar_buffer[0]=c
```

Bufferdin' jana ma'nisi

```
Qatar_buffer=c++
```

```
Qatar::~Qatar()
```

xabarları ko'rinedi.

Sorawlar ha'm tapsırmalar

1. Operatorlardı qayta ju'klew ahmiyeti nede?
2. Ne ushın ayırım operatorlar qayta ju'klenbeydi?
3. Qaysı jag'daylarda operator-funktsiya dos funktsiya ko'rinishinde anıqlanadı?
4. Binar operatorının' operator-funktsiyasının' birinshi parametrine qanday talaplar qoyıladı?
5. Inkrement ha'm dekrement a'mellerin qayta ju'klewde olardıń yamasa ko'rinishinleri qanday inabatqa alıladı?
6. Ma'nis beriw operatorın qayta ju'klewdin' o'zine ta'nligi nede?

In C++, operators can be overloaded relative to class types that you define. The principal advantage to overloading operators is that it allows you to seamlessly integrate new data types into your programming environment.

Operator overloading allows you to define the meaning of an operator for a particular class. For example, a class that defines a linked list might use the + operator to add an object to the list. A class that implements a stack might use the + to push an object onto the stack. Another class might use the + operator in an entirely different way. When an operator is overloaded, none of its original meaning is lost. It is simply that a new operation, relative to a specific class, is defined. Therefore, overloading the + to handle a linked list, for example, does not cause its meaning relative to integers (i.e., addition) to change.

Operator overloading is closely related to function overloading. To overload an operator, you must define what the operation means relative to the class to which it is applied. To do this, you create an **operator** function, which defines the action of the operator. The general form of an **operator** function is

```
type classname::operator#(arg-list)
{
    operation relative to the class
}
```

Here, the operator that you are overloading is substituted for the #, and *type* is the type of value returned by the specified operation. Although it can be of any type you choose, the return value is often of the same type as the class for which the operator is being overloaded. This correlation facilitates the use of the overloaded operator in compound expressions. The specific nature of *arg-list* is determined by several factors,

Operator Overloading Tips and Restrictions

The action of an overloaded operator, as applied to the class for which it is defined, need not bear any relationship to that operator's default usage, as applied to C++'s built-in types. For example, the << and >> operators, as applied to **cout** and **cin**, have little in common with the same operators applied to integer types. However, to maintain the transparency and readability of your code, an overloaded operator should reflect, when possible, the spirit of the operator's original use. For example, the + relative to **three_d** is conceptually similar to the + relative to integer types. There would be little benefit in defining the + operator relative to some class in such a way that it acts more the way you would expect the || operator, for instance, to perform. The central concept here is that, while you can give an overloaded operator any meaning you like, for clarity, it is best when its new meaning is related to its original meaning.

There are some restrictions to overloading operators. First, you cannot alter the precedence of any operator. Second, you cannot alter the number of operands required by the operator, although your operator function could choose to ignore an operand. Finally, except for the function call operator (discussed later), operator functions cannot have default arguments.

The only operators that you cannot overload are shown here:

. :: .* ?

The .* is a special-purpose operator, discussed later in this book.

34-Leksiya. Polimorfizm xa'm virtual funktsiyalar

Joba:

1. Wakıtlı xa'm keshiktirilgen baylanıslar. Dinamiklik polimorfizm

2. Birtual funktsiyalar

3. Virtual xa'm virtual emes funktsiyalar

Wakıtlı xa'm keshiktirilgen baylanıslar. Dinamiklik polimorfizm:

C++ tilinde polimorfizm eki usılda boladı. Birinshi, funktsiya operatorların qayta jukleu usılları menen komplyatsya uaktında. Polimorfizmnın bul korinisı *statikalık* polimorfizm delinedı, sebebi programma orınlanıudan aldın, yagnıy komplyatsya xa'm jıynap (komponovka) uaktında funktsiya identifikatorlardı fizikalık adresler menen wakıtlı baylanıs arkalı amelge asırıladı. Ekinshiden, programma orınlanıuında virtual funktsiya jardeminde. Programma kodında virtual funktsiya shakırğanda komplyator, bul shakırğanın tek belgilep koyadı, funktsiya identifikatorlar adres penen baylanıslı programma orınlau baskışına kaldıradı. Bul protseske *keshiktirilgen baylanıs* delinedı. Virtual funktsiya- bul sonday funktsiya, onı shakırğan xa'm saykes amellerdi orınlau, onı shakırğan ob'ekt turine baylanıslı boladı. Ob'ekt programma orınlau uaktında kaysı funktsyanı shakırıu kerekligin anıklaydı. Bunday polimorfizmler *dinamiklik polimorfizm* delinedı. Dinamiklik polimorfizmdi amelge asırıudın tiykarı sıpatında C++ tilindegi tayanış klasına korsetkishti anıklastırıu. Oz naubetinde bul korsetkish tek gana tayanış klasına balkim usı klastın miyrası bolgan ıktıyarıy klass ob'ektine korsetiui mumkin. Klaslardın bul kasiyeti miyraslıktan kelip shıgadı, sebebi xar kanday miyraslık klass ob'ekti tayanış klass turinde boladı. Programma jıynau uaktında (komponovka uaktında) tayanış klass korsetkishin iyesi bolgan paydalanıushı tarepinen kaysı klass ob'ekti jaratılıuı belgisiz boladı. Sol sebepli, korsetkish oz ob'ekti menen tek programma isleu uaktında, yagnıy dinamiklik ravishte baylanıslıuı mumkin. Kuramında xesh bolmaganda bir virtual funktsyası bolgan klass *polimorfizm klass* delinedı. Xar bir polimof turindegi berilgenler ushın komplyator *virtual funktsiyalar kestesin* jaratadı xa'm sol kestege klastın xar bir ob'ektine jasırınan korsetkishti

jaylastıradı. Komplyator virtual funktsiyalar kestege korsetkishti initsiyalizatsiya kılıushı kod bolganı polimorf klass konstukturı baslanıuına jaylastıradı. Virtual funktsiya shakırılghanda bul kod virtual funktsiyalar kestesine korsetkishti tabadı xa'm kestedegi saykes funktsiya adreske otıu menen funktsiya shakırılıuı juz beredi.

Keste. Klass virtual funktsiyalar kestesı

Klass shejiresi xa'm korsetkishler	Korsetkish ma'nis turi	Klastag'ı virtual funktsalardıń adresleri		
		f1()	f2(
Tayanish *tayanish;	Tayansh	Adres11	Adres21	...
Miyras1*miyras1; //Tayansh miyrasi	Miyras1	Adres12	Adres22	...
Miyras2*miyras2; //Tayansh miyrasi	Miyras2	Adres13	Adres23	...
...

Bizge belgili tuwındı klass ob'ekti jaratılıuında tayanish klass konstrukturı shakırıladı, yagnıy sol baskıshta virtual funktsiyalar kestesı xa'm ogan korsetkish payda boladı. Tuwındı klass konstrukturı shakırılghannan keyin virtual funktsiyalar kestesine korsetkish, usı klass ob'ekti ushın qayta anıqlangan virtual funktsiya variantın korsetiu ushın tuwrılasadı (eger ol bar bolsa).

Korinip turıptı, keshiktirilgen baylanıstı amelge asırıu bir resurslar sarıpladı talap etedi xa'm onnan durıs paydalanıu zarur boladı.

Birtual funktsiyalar: Mazmunınan kelip shıkkán xalda virtual funktsiyaga baska usınıslardı beriuı mumkin.

1) Shakırıu interfeysi (protatipi) belgili, amelge asırılıuı ulıumalık koriniste beriliuı mumkin bolmıstan, tek gana konkret jagdaylarda anıqlanatuğın funktsiyalarga *virtual funktsiyalar* delinedi.

2) *Virtual funktsiyalar* –bul shakırılıuı ushın kanday anlatpa isletiliuine karamıstan ob'ekt ushın durıs (saykes) funktsiya shakırılıuın taminleytuğın funktsiya.

Meylı ,tiykargı klastagı funktsiyanın virtual dagazası ,tuwındı klasında xa'm sol funktsiya dagazası bolsın. Ol gezde tuwındı klass ob'ektleri ushın tuwındı klass funktsiyası shakırıladı, eger olar shakırılıuında tiykargı klaska korsetkish yaki muraajat isletilgen bolsa da.

```

#include <iostream>
#include<string.h>
using namespace std;
class Tayanish
{
public:
    Tayanish(int _x) {x= _x;}
    virtual int Manis_x()
    { return x;}
    virtual void Chop_X();
private :
    int x;
};
void Tayanish::Chop_X(){ cout<<"Tayanish ::x=" <<Manis_x() <<' \n';}
class Tuwindi1: public Tayanish
{
public:
    Tuwindi1(int _x): Tayanish(_x){}
    void Chop_X();
};
void Tuwindi1::Chop_X() {cout <<"Tuwindi 1::x=" <<Manis_x()
<<' \n';}
class Tuwindi2:public Tayanish
{
public :
    Tuwindi2(int _x): Tayanish(_x){}
    void Chop_X();
};
void Tuwindi2::Chop_X() {cout <<"Tuwindi 2 ::x=" <<Manis_x()
<<' \n';}
int main(int argc, char *argv[]){
    Tayanish*tayanish=new Tayanish(10);
    Tuwindi1*tuw1=new Tuwindi1(20);
}

```

```

Tuwindi2*tuw2 =new Tuwindi2(30);

tayanish->Chop_X();
tayanish=tuw1;
tayanish->Chop_X();
tayanish=tuw2;
tayanish->Chop_X();
return 0;
}

```

Tuwindi klaslarında Chop_X() funksiyaları virtual esaplanadı, sebebi ol Tayanish tayanish klasında virtual dep dagaza kilingan. Virtual funksiyaların shakırıw ushın tomendegi kodlar kollanılğan:

```

tayanish=tuw1;
tayanish->Chop_X();
tayanish=tuw2;
tayanish->Chop_X();

```

Klaslardagi Chop_X() funksiyaları virtual bolğanlıgı ushın xar bir ob'ekttin oz funksiyası shakırıladi. Tuwindi1 xa'm Tuwindi2 klaslardagi Chop_X() funksiyalar Tayanish tayanish klasında Chop_X() funksiyanı qayta anıklaydı. Eger tuwindi klasta Chop_X() funksiyası qayta anıklanbasa, kelisiw boyınsha tayanish klasında Chop_X() funksiyasın kabıl etedi. Jokarıdagı dastur islewi natiyjesinde ekranga

```

Tayanish::=10
Tuwindi::=20
Tuwindi::=30

```

natiyjesi baspaga shıgarıladi:

Funksiyalardı korsetkish xa'm adresti alıw amelleri jardeminde shakırıwdagi tomendegi kagıydalarga amel kılınadı:

- virtual funksiyanı shakırıw onı shakırılıp atırğan ob'ekt turine saykes turde sheshiledi.
- virtual bolmagan funksiyalardı shakırıw korsetkish turine saykes turde amelge asırıladi.

Virtual funktsiyalar tek gana kadayda bir klaska tiyisli ob'ektler ushin shakiriliwin itibarga alatugin bolsak, global yaki statikalik funktsiyalardi virtual dep dagaza kilw mumkin emes. Virtual gilt sozi tuwindi klasında funktsiyani qayta aniklawda isletiliwi mumkin, biraq bul majburiy emes. Tayanish klastagi virtual funktsiyalar sol klasta aniklanıwı kerek, eger olar xakıkiy virtual dep dagaza kilingan bolmasa. Tuwindi klasında dagaza kilingan funktsiya tayanish klasındağı virtual funktsiyani qayta aniklaydı, eger onın atı virtual funktsiya atı menen saykes tusse, olar birdey mugdarda xa'm turleri saykes kelgen parametrlerge iye bolsa. Eger funktsiya virtualdan xatteki birgana parametri menen parık kılssa, bunday jagdayda tuwindi klasındağı bul funktsiya taza esaplanadı xa'm qayta aniklaw juzbermeydi. Tuwindi klasındağı bul funktsiya virtual funktsiyadan tek gana qaytarıwshı manis penen parıklanbaydı, olardıń parametrleri turlishe bolıwı kerek.

Tomendegi misalda vitual funktsiya tayanish klasındağı ozi menen bir turdegi prototipge iye virtual funktsiyani qayta aniklaydı.

```
#include <iostream>
#include <string.h>
using namespace std;
class Tayansh
{int x;
public:
virtual void Manis(int _x){x=_x;cout<<"Tayansh::"<<x<<"\n";}
virtual void baspaga_shigariw(Tayansh*pOb){Manis(10); }
};
class Tuwindi:public Tayansh
{int x,y;
public:
virtual void Manis(int _x,int _y){x= _x; y= _y;
cout<<"Tuwindi::x="<<x<<"Tuwindi::y="<<y<<"\n"; }
virtual void baspaga_shigariw(Tayansh*pOb){Manis(15,20); }
};
int main()
```

```

{
Tayansh*pOb1=new Tayansh;
Tayansh*pOb2=new Tuwindi;
//Tayanish klassinan virtual baspaga_shigariw() funktsiyani shaqiriw
pOb1->baspaga_shigariw(pOb1);
//Tuwindi klassinan virtual baspaga_shigariw() funktsiyani shaqiriw
pOb2->baspaga_shigariw(pOb1);
//Tuwindi klassinan virtual baspaga_shigariw() funktsiyani shaqiriw
pOb2->baspaga_shigariw(pOb2);
return 0;
}

```

Dastur islewi natijesinde ekranga tomendegiler baspaga shıgarıladı:

Tayanish::x=10

Tuwindi::x=15 Tuwindi::y=20

Tuwindi::x=15 Tuwindi::y=20

Keltirilgen misalda tayanish xa'm tuwindi klaslar eki birdey attagi virtual funktsiyalarga iye. Birak kompilyator olardi turlishe usunis etedi. Manis() funktsiyasının prototipi tuwindi klasında ozgergenligi sebepli, ol putkilley baska virtual funktsiya dep karaladı. Ekinshi tarepten, tuwindi klasında baspaga_shigariw() funktsiyası tayanish klasında saykes virtual funktsiyanın qayta anıklanıwı dep karaladı.

Virtual xa'm virtual emes funktsiyalar: Tomendegi misalda korsetkish arkalı shakırılğanda virtual xa'm virtual emes funktsiyalar ozin kaday tutıwı korsetilgen:

```
#include <iostream>
```

```
#include <string.h>
```

```
using namespace std;
```

```
class Tayanish
```

```
{
```

```
public:
```

```
virtual void Virtual_Fun(){cout<<"Tayanish::Virtual_Fun()\n";}
```

```

void Virtualemes_Fun(){cout<<"Tuwindi::Virtual emes_Fun()\n";}
};
class Tuwindi:public Tayanish
{
public:
virtual void Virtual_Fun(){cout<<"Tuwindi::Virtual_Fun()\n";}
void Virtualemes_Fun(){cout<<"Tuwindi::Virtual emes_Fun()\n";}
};
int main()
{
Tuwindi tuwindi;
Tuwindi *pTuwindi =&tuwindi;
Tayanish *pTayanish=&tuwindi;
//Tuwindi klass Virtual_Fun() funktsiyani shaqiriw
pTayanish->Virtual_Fun();
//Tayanish Virtual emes_Fun() funktsiyani shaqiriw
pTayanish->Virtualemes_Fun();
//Tuwindi Virtual_Fun() funktsiyani shaqiriw
pTuwindi->Virtual_Fun();
//Tuwindi Virtual_Fun() funktsiyani shaqiriw
pTuwindi->Virtualemes_Fun();
return 0;
}

```

programma isleui natiyjesinde ekranga tomendegi katarlar shıgarıladı:

```

Tuwindi::Virtual_Fun()
Tayanish::Virtual emes_Fun()
Tuwindi::Virtual_Fun()
Tuwindi::Virtual emes_Fun()

```

Usıgan itibar beriu kerek ,Virtual_Fun() funktsiyası kaysı klaska tayaanish yamasa tuwindı korsetkish arkalı shakırılıuına karamastan tuındı klasınan Virtual_Fun() funktsiyası shakırıladı. Bagan sebeb Virtual_Fun() funktsiyası virtual xa'm pTayanish xa'mde pTuwindi kórsetkshleri turindegi ob'ektke kórsetedı. Ekinshi tarepten, tayanish klaska kórsetkishi

pTayansh tek gana virtual emes funktsiyalarga iye tuindi klass ob'ektine korsetse xa'm, tayanish klastagi saykes funktsiyani shakiradi. Koru kaniygesine ruxsat beru operatori jardeminde keshiktirilgen baylanisti jok etuii mumkin :

```
#include <iostream>
using namespace std;
class Tayansh{
    public:
    virtual void Virtual_Fun() {
        cout<<"Tayansh::Virtual_Fun()\n";
    }
};
class Tuwindi:public Tayansh
{
    public:
    virtual void Virtual_Fun(){
        cout<<"Tuwindi::Virtual_Fun()\n";}
};
int main()
{
    Tayansh*pTayansh=new Tuwindi;
    // Tuwindi klassinan virtual_Fun() funktsiyani shaqiriw
    pTayansh->Virtual_Fun();
    //Tayansh klassinan Virtual_Fun() funktsiyani shaqiriw
    pTayansh->Tayansh::Virtual_Fun();
    return 0;
}
Programma ekranında
Tayansh::Virtual_Fun()
Tuwindi::Virtual_Fun()
Xabardi ekranda shıgaradı.Körinip turganıday
pTayansh->Tayansh::Virtual_Fun();
korsetpesi keshiktirilgen baylanisti jok etedi.
```

Klaslar shejiresinde virtual funktsiya atı menen birdey qayta
Jukleniushi funktsiyalardı jasırangan boladı.

Jokardagı pikirlerdi tastıyklauga mısıl keltiremiz:

```
#include <iostream>
#include <string.h>
using namespace std;
class Tayansh
{
public:
Tayansh(char*ati) {
strcpy(Ati,ati);}
virtual void Fun(char c){
cout<<"Virtual"<<Ati<<"::Fun"<<c<<"parametirin qabil qiladi.\n";}
protected:
char Ati[20];
};
class Tuwindi1:public Tayansh
{
public:
Tuwindi1(char * ati):Tayansh(ati){}
void Fun(const char *s)
{
cout<<Ati<<"::Fun"<<s<<"parametrlerdi qabil etedi.\n";}
void Fun(int n) {
cout<<Ati<<"::Fun"<<n<<"parametirlerdi qabil etedi.\n";}
virtual void Fun(char c)
{cout<<"Virtal"<<Ati<<"::Fun"<<c<<"parametirlerdiqabil      etedi.\n";
};
class Tuwindi11:public Tuwindi1
{public:
Tuwindi11(char *ati):Tuwindi1(ati){}
void Fun(const char *s) {
cout<<Ati<<"::Fun"<<s<<" parametirledi qabil etedi.\n";}
```

```

void Fun(double d)
{cout<<Ati<<"::Fun"<<d<<"parametirledi qabil etedi.\n";}
virtual void Fun(char c){
cout<<"Virtual"<<Ati<<"::Fun"<<c<<"parametirledi qabil etedi.\n";}
};
int main(){
Tayansh tayansh("tayansh");
Tuwindi1 tuwindi1("tuwindi1");
Tuwindi11 tuwindi11("tuwindi11");
tayansh.Fun('x');
tuwindi1.Fun('y');
tuwindi1.Fun(10);
tuwindi1.Fun("Tuw1");
tuwindi11.Fun('Z');
tuwindi11.Fun(10.1234);
tuwindi11.Fun("tuw11");
return 0;
}

```

Programma orinlangan keyin ekranda tomendegi katarlar payda boladi:

```

Virtual Tayansh::Fun X parametirledi qabil etedi .
Virtual Tuwindi1::Fun Y parametirledi qabil etedi.
Tuwindi1::Fun 10 parametirledi qabil etedi.
Tuwindi1::Fun Tuw1 parametirledi qabil etedi.
Virtual Tuwindi11::Fun Z parametirledi qabil etedi.
Tuwindi1::Fun 10.1234 parametirledi qabil etedi.
Tuwindi11::Fun Tuw11 parametirledi qabil etedi.

```

Keltirilgen misalda sızıklı miyraslıktı payda kılıushi 3 klass anıqlangan. Tayanış klasın da Fun(char) virtual funksiya dagaza kılıngan. Tuwindi1 klass Fun(char) virtual funksiyanın oz variantın xa'm eki qayta jukleniushi virtual emes Fun(const char*) xa'm Fun(int)funksiyalar dagaza kılıngan. Öz naubetinde, Tuwindi11 klass Fun(char) virtual funksiyanın oz variantın xa'm eki qayta jukleniushi virtual emes Fun(const char*) xa'm

Fun(double) funksiyalardı dagaza kılğan. Sonday-ak , Fun(const char*) funksiyası Tuwindi1 klasında analogı menen tolıq ustpe-ust tusse de, usı Tuwindi11 klasında qayta dagaza kılınan. Sonlıktan,

Tuwindi1 klasında sol atındağı virtual xa'm virtual emes Funksiyalar barlığı sebepli, Fun(const char*) funksiyası jasırınan boladı. Usıǵan uksas ,Fun(char) virtual funksiyası

Tuwindi1 xa'm Tuwindi11 klaslarında qaytadan dagaza kılıuga turı keledi, sonlıktan olar xa'm klaslarda usıǵan uksas attağı qayta yukleniushi funksiyalardıń bar ekenligi sebepli jasırınan boladı.Eger miyras klaslardadı virtual funksiyalar dagazaları eshirilse, funksiyanın belgili argumentli shakırıuda, amelde Tuwindi1 klasındağı Fun(int) funksiyası shakırıu juz beredi. Tayanısh klasındağı virtual funksiyanı shakırıu zarur bolsa, korigu kaniygesine ruxsat beriu amelinen paydalanıu mumkin.

Tuwindi1.Tayanish::Fun('Y');

In C++, polymorphism is supported both at run time and at compile time. Operator and function overloading are examples of compile-time polymorphism. As powerful as operator and function overloading are, they cannot perform all the tasks required by a true, object-oriented language. Therefore, C++ also allows run-time polymorphism through the use of derived classes and *virtual functions*, and these are the major topics of this chapter.

This chapter begins with a short discussion of pointers to derived types, because they provide support for run-time polymorphism.

Virtual Functions

Run-time polymorphism is achieved through a combination of two features: inheritance and virtual functions. You learned about inheritance in the preceding chapter. Here, you will learn about the virtual function.

A *virtual function* is a function that is declared as **virtual** in a base class and redefined in one or more derived classes. Thus, each derived class can have its own version of a virtual function. What makes virtual functions interesting is what happens when one is called through a base class pointer (or reference). In this situation, C++ determines which version of the function to call based upon the *type* of the object *pointed to* by the pointer. And, this determination is made *at run time*. Thus, when different objects are pointed to, different versions of the virtual function are executed. In other words, it is the type of the object being pointed to (not the type of the pointer) that determines which version of the virtual function will be executed. Therefore, if a base class contains a virtual function, and if two or more different classes are derived from that base class, then when different types of objects are pointed to through a base class pointer, different versions of the virtual function are executed. The same thing occurs when using a base class reference.

You declare a function as virtual inside the base class by preceding its declaration with the keyword **virtual**. When a virtual function is redefined by a derived class, the keyword **virtual** need not be repeated (although it is not an error to do so).

A class that includes a virtual function is called a *polymorphic class*. This term also applies to a class that inherits a base class containing a virtual function.

Examine this short program, which demonstrates the use of virtual functions:

Why Virtual Functions?

As stated at the beginning of this chapter, virtual functions in combination with derived types allow C++ to support run-time polymorphism. Polymorphism is essential to object-oriented programming for one reason: It allows a generalized class to specify those functions that will be common to all derivatives of that class, while allowing a derived class to define the specific implementation of some or all of those functions. Sometimes this idea is expressed as follows: The base class dictates the general *interface* that any object derived from that class will have, but lets the derived class define the

actual *method* used to implement that interface. This is why the phrase “one interface, multiple methods” is often used to describe polymorphism.

35-Leksiya. Polimorfizm ha'm virtual funktsiyalar

Joba:

1. Dinamik polimorfizmdi qollaw
2. Virtual destruktolar
3. Abstrakt klasslar ha'm sap virtual funktsiyalar

Dinamik polimorfizmdi qollaw: Dinamik polimorfizm ja'rdemide da'stu'r orinlanıwın basqarıwdın' sa'ykeslesiwshi basqarıwdı a'melge asırıw mu'mkin. To'mende, pu'tin sanlardın' baylanısqa dizimi ko'rinishinde a'melge asırılǵ'an stek ha'm gezek du'zilmeleri u'stinde islew qaralg'an. Bizge belgili, gezek- «*birinshi kelgen-birinshi ketedi*», stek-«*aqırında kelgen-birinshi ketedi*» printsipi boyınsha berilgenlerdi saqlaw ha'm qayta islewdi a'melge asırıwshı du'zilmeler esaplanadı. Da'stu'rde baylanısqa dizimdi jaratıw, og'an ma'nis jaylastırıw ha'm o'shiriwdi a'melge asırıwshı Dizim tayanış klası ha'm onın' miyrasları sıpatında gezek payda qılıwshı saykes ra'wishte Gezek ha'm Stek klassları jaratıladı. Sonda da bul du'zilmalar menen islew tu'rlishe a'melge asırılsada, olardı isletiwde bir g'ana interfeysten paydalanıladı.

```
#include<iostream>
#include<stdlib.h>
#include<ctype.h>
using namespace std;
class Dizim
{
public:
Dizim() { Basi=Aqiri=Keyingi=0; }
virtual void Jaylastiriw(int n)=0;
virtual bool Aliw(int &n)=0;
void Manis_n(int n) { San=n; }
int n_Manis() { return San; }
Dizim*Basi;
Dizim*Aqiri;
Dizim*Keyingi;
```

```

private:
int San;
};
class Gezek: public Dizim
{
public:
void Jaylastiriw (int n);
bool Aliw(int&n);
};
void Gezek::Jaylastiriw(int n)
{
Dizim*Jana;
Jana=new Gezek; //Gezektegi jan'a elementti jaratiw
Jana->Manis_n(n);
Jana->Keyingi=NULL;
if(Aqiri) Aqiri->Keyingi=Jana;
Aqiri=Jana;
if(!Basi) Basi=Jana;
}
bool Gezek::Aliw(int&n)
{
Dizim*Element;
if(!Basi){n=0; return false;}
n=Basi->n_Manis();
Element=Basi;
Basi=Basi->Keyingi;
delete Element;
return true;
}
class Stek: public Dizim
{
public:
void Jaylastiriw(int n);

```

```

bool Aliw(int&n);
};
void Stek::Jaylastiriw(int n)
{
Dizim*Jana;
Jana=new Stek; // Stektin' jan'a elementin jaratiw
Jana->Manis_n(n);
Jana->Keyingi=NULL;
if(Basi) Jana->Keyingi=Basi; //Elementti stek basına jaylastırıw
Basi=Jana;
if(!Aqiri) Aqiri=Jana;
}
bool Stek::Aliw(int&n)
{
Dizim*Element;
if(!Basi){n=0; return false;}
n=Basi->n_Manis();
Element=Basi;
Basi=Basi->Keyingi;
delete Element;
return true;
}
int main()
{
Dizim*dizim;
Gezek gezek;
Stek stek;
int san;
char stek_gezek;
cout<<"Sanlardi gezek ha'm stekke jaylastiriw:\n";
do
{
cout<<"Sandi kiritin(0-tamam):";

```

```

cin>>san;
if(san)
{
do
{
cout<<"Jaylastiriw? Stekke(S) yaki Gezekke (N):";
cin>>stek_gezek;
}
while(stek_gezek!='S'&&stek_gezek!='s'
&&stek_gezek!='N'&&stek_gezek!='n');
switch(stek_gezek)
{
case'S':case's': dizim=&stek; break;
case'N':case'n': dizim=&gezek;
}
dizim->Jaylastiriw(san);
}
}
while(san);
for(;;)
{
do
{
cout<<"Oqiw? Stekden(S) yaki Gezekten(N):\n";
cout<<"Dasturden shigiw(Q):\n";
cin>>stek_gezek;
}
while (stek_gezek!='S'&&stek_gezek!='s'
&&stek_gezek!='N'&&stek_gezek!='n'&&
stek_gezek!='Q'&&stek_gezek!='q');
switch(stek_gezek)
{
case'S':

```

```

case's':dizim=&stek; break;
case'N':case 'n':dizim=&gezek; break;
case'Q':case'q':return 0;
}
if(dizim->Aliw(san))cout<<san<<endl;
else cout<<"Dizim bos!"<<endl;
}
}

```

Da'stu'r kiritiw ag'ımınan pu'tin sanlardı oqıydı ha'm paydalanıwshı tan'lag'an dizimge-gezek yaki stekke jaylastıradı. Sanlardı kiritiw protsessi gezektegi san sıpatında 0 sanı kiritilgende toqtaydı. Keyinshelik paydalanıwshı ko'rsetken dizimnen san ma'nisleri oqıladı ha'm ekrang'a shıg'arıladı. Dinamik polimorfizm dizim ko'rsetkishi gezek ha'm stek ob'ektlerine ko'rsetiwine sa'ykes virtual Aliw() ha'm Jaylastiritiw() funktsiyaların shaqırıwda anıq boladı.

Virtual destruktorklar: Konstruktorlar virtual bolmaydı, biraq destruktorklar virtual bolıwı mu'mkin ha'm ayırım jag'daylarda sonday boladı. Tayanısh klassqa ko'rsetkish tuwındı klass ob'ektine ko'rsetip turg'anda, eger destruktork virtual qılıp dag'aza qiling'an bolsa, tuwındı klass destruktorkı shaqırılıdı. Tuwındı klass destruktorkı o'z nawbetinde tayanısh klass destruktorkın shaqıradı ha'm ob'ekt tuwrı (tolıg'ı menen) o'shiriledi. Keri jag'dayda ko'rsetkish tu'rine sa'ykes ra'wishte tayanısh klass destruktorkı shaqırıladı, tuwındı klass ushın ajratıl'g'an yad bosatılmay qaladı- yadta ba'nt qiling'an, biraq qayta islew imkanı bolmag'an yad bo'legi –«yad shıg'ındısı» payda boladı.

Mısal ko'reyik:

```

#include <iostream>
class Tayansh
{
int *px;
public:
Tayansh(int _x) {px=new int; *px= _x;}

```

```

    /*virtual*/ ~Tayansh() {cout<<"Tayansh klass destruktori isledi!\n";
delete px;}
};
class Tuwindi: public Tayansh
{
int *pxx;
public:
    Tuwindi (int n): Tayansh (n) {pxx=new int; *pxx=n*n;}
    ~Tuwindi() {cout<<"Tuwindi klass destruktori isledi!\n"; delete
pxx;}
};
int main()
{
Tayansh *pTayansh=new Tuwindi (5);
delete pTayansh;
return 0;
}

```

Tuwindi klass- Tuwindi klassta destruktör virtual dep dag'aza qilinbag'an ha'm

```
delete pTayansh;
```

til ko'rsetpesi orinlanıwı na'tiyjesinde ekrang'a

Tayansh klass destruktörı isledi!

xabarı shıg'adı. Bul jag'day yadtag'ı Tuwindi klass ob'ekti ushın ajratılǵ'an yad bosatılmay qalg'anlıǵ'ın bildiredi. Eger tayanış klass destruktörin virtual dep dag'aza qılınsa, ob'ekti o'shiriw tuwrı ju'z beredi-aldın tuwindi klass destruktörı, keyin tayanış klass destruktörı orınlanadı. Da'stu'rdin' ekrang'a shıg'aratug'ın

Tuwindi klass destruktörı isledi!

Tayansh klass destruktörı isledi!

xabarları bunı da'lilleydi.

Abstrakt klasslar ha'm sap virtual funktsiyalar: Klasslar, sol tu'rge tiyisli bolg'an ob'ektlardin' o'z-ara orinlaytug'in a'melleri qag'iydalardi aldinnan aniqlap beriw ushin jaratiliwi mu'mkin. Bunday klasslarga *abstrakt klasslar* delinedi. Abstrakt klasslardin' ob'ektlarin jaratip bolmaydi. Olar tek g'ana tuwindi klasslardi jaratiw ushin xizmet qiladi.

Abstrakt klass keminde bir virtual funktsiyag'a iye boliwı kerek. Tayanish klasstin' sap virtual funktsiyalari tuwindi klasslarda albette aniqlanıwı kerek, keri jag'dayda tuwindi klassta virtual esaplanadı.

Sap virtual funktsiya to'mendegi sintaksis penen dag'aza qilinadi.

virtual <funktsiya ati><(parametrler dizimi)>=0;

Mısal ko'reyik. Oylap ko'reyik, klasslar shejeresin jaratiw zaru'r bolsın ha'm tayanish klass uliwma funktsional imkaniyatlardi ta'miynlewi kerek bolsın. Biraq, tayanish klass sol da'rejede uliwmalasqan bolıp, na'tiyjede ondag'ı ayırım funktsiyalardi konkretlestiriw imkanı bolmaslıg'ı mu'mkin. Bunday tayanish klası abstrakt klass ushin en' jaqsı at esaplanadı:

Mısal ushin haywanatlar shejeresin suwretlewshi haywanat abstrakt tayanish klass ha'm onın' miyrasları Ku'shik ha'm Pıshıq klassların dag'aza qılıwdı ko'reyik.

```
class Haywanat
{
public:
Haywanat(char*ati)
{
Ati=new char[15];
strcpy(Ati,ati);
};
virtual void Dawsı()=0;
virtual void Awqati()=0;
protected:
char*Ati;
};
```

```

class Kushik: public Haywanat
{
public:
    Kushik(char*ati):Haywanat(ati){};
    void Dawsı(){cout<<Ati<<" dawsı: Waw"<<endl;}
    void Awqati(){cout<<Ati<<" awqati: Go'sh"<<endl;}
};

class Pishiq: public Haywanat
{
public:
    Pishiq(char*ati):Haywanat(ati){};
    void Dawsı(){cout<<Ati<<" dawsı: Miyaw"<<endl;}
    void Awqati(){cout<<Ati<<" awqati: Su't"<<endl;}
};

int main()
{
    Pishiq pishiq("Baraq");
    Kushik kushik("Tuzik");
    kushik.Dawsı();
    kushik.Awqati();
    pishiq.Dawsı();
    pishiq.Awqati();
}

```

Bul mısaldın' itibarlı ta'repi sonda, Haywanat klasında dag'aza qiling'an dawsı() ha'm azıg'ı() funktsiya ag'zalar abstrakt funktsiyalar. Bul funktsiyalardı konkretlestiriwdin' imkanı joq, sebebi Haywanat klası haywanlar shejeresin anıqlap beriwshi , ulıwmalastırıwshı klass ha'm konkret haywan anıqlanbag'ansha onın' dawsının' qanday bolıwı ha'm ne menen azıqlanıwın bilip bolmaydı. Biraq, ayırım haywanlar dawıs shıg'aradı ha'm a'lbette azıqlanadı. Sol sebepli dawsı() ha'm azıg'ı() funktsiyaları ulıwma bolıp, ol haywanat klasında jormal halda dag'aza

qiling'an. Mısalda bul funksiyalar Ku'shik ha'm Pıshıq klasslarında konkretlestirilgen.(ma'jburiy ra'wishte)

Da'stu'r islewi na'tiyjesinde to'mendegi xabarlar shıg'adı.

Baraq dawsı. Miyaw;

Baraq awqati. Su't;

Tuzik dawsı.Waw;

Tuzik awqati. Go'sh;

Abstrakt klass penen baylanıslı ja'ne bir o'zine ta'n jag'day sonnan ibarat, eger abstrakt klass konstruktory sap abstrakt funktsiyany shaqırsa, ne ju'z beriwın aldınnan aytıwdın' ilajı joq.

Sap abstrakt funksiyalar sıpatlamasına «qarsı» ra'wishte bunday funksiyalar abstrakt klassta tek g'ana dag'aza qılınıwı, ba'lki anıqlanıwı da mu'mkin. Olar to'mendegi sintaksis tiykarında shaqırılıwı mu'mkin.

<abstrakt klass atı>::<abstrakt funktsiya atı>(<parametrler dizimi>)

A'dette bul sintaksisten sap virtual destruktorg'a iye klasslar shejeresin jaratıwda paydalanıladı:

```
#include<iostream>
using namespace std;
class Tayanch
{
public:
Tayanch() {
};
virtual~Tayanch()=0; // Sap virtual destruktory
};
Tayanch::~~Tayanch(){ //Destruktordı anıqlaw
class Hosila: public Tayanch
{
public:
Hosila (){
};
```

```

    ~Hosila(){
    };
    int main();
    Hosila *pHosila=new Hosila;
    delete pHosila;
    }

```

Bizge belgili, destruktur virtual bolg'anda, aldın tuwindı klass destrukturı, keyin tayanış klass destrukturı orınlanadı. Sap virtual destruktordın sonda da tayanış klassta anıqlanıwı, onın' qandayda a'melge asırılǵ'an variantın jaratadı, ol destrukturlar izbe-izligin durıs orınlanıwın ta'miynleydi.

Juwmaq sıpatında abstrakt klasslarga qollanılaturǵın qag'ıydalardı keltiremiz.

-abstrakt klasstı funktsiyag'a uzatılaturǵın argumenttin' tu'ri sıpatında isletip bolmaydı;

-abstrakt klasstı funktsiya qaytaraturǵın ma'nistin' tu'ri sıpatında isletip bolmaydı.

-obekt tu'rin a'shkar ra'wishte abstrakt klass tu'rine keltiriw mu'mkin emes.

-abstrakt klass obektin jaratıp bolmaydı.

-abstrakt klassqa ko'rsetkish yaki adresti alıw a'melin dag'aza qılıw mu'mkin.

36-Leksiya. Ayırıqsha jag'daylar

Joba:

1. Ayırıqsha jag'daylar tusnigi
2. Ayırıqsha jag'daylardı juzege keltiriw.
3. Ayırıqsha jag'daylardı ilip alıu

Ayırıqsha jag'daylar: Bul programmanın' normal isleuin uzetug'in jag'day bolıp esaplanadı. C++ tilinde ayırıqsha jag'daydı qayta isleudin' tamamlangan model dep atalatug'in varyantı amelge asırılǵ'an ayırıqsha jag'day juz berip, ol qayta islengennen keyin basqariw ayırıqsha jag'day juz bergен kodqa qaytıp kelmeydi, yag'nıy programmanın' tap usı uziliu juz bergен jerden dauam etiui tamiyinlenbeydi. Ekinshi tarepten C++ tilinde apparat qurılmaldan keletug'in ayırıqsha jag'daylar (uziliuler) qayta islenbeydi, tek g'ana qandayda funktsiya tarepinen juzege keltirilgen ayırıqsha jag'daylar qayta islenedi. Ayırıqsha jag'daylar basqariu ushin C++ tilinde ush gilt so'zi isletiledi: try, catch, throw.

Ayırıqsha jag'daydı juzege keltiriwi mumkin bolǵ'an kod bloǵı try gilt so'zi menen belgilenedi. Bul blok figuralı qausqa alınadı ham qorg'alg'an yamasa try-blok delinedi:

```
try
{
//Qorg'alg'an oblost
}
```

Try-blok ishinde shıǵ'arılatug'in funktsiyalar ham qorg'alg'an oblostg'a tiyisli boladı. Qorg'alg'an oblostdag'ı funktsiya yaki til ko'rsetpesi ayırıqsha jag'daylardı juzege keltiriw mumkin. Eger ayırıqsha

jaqdaylar juz berse, saykes funktsiya yaki til ko'rsetpesinin' bejeriliui toqtatiladi,try-blok tag'ı qalg'an ko'rsetpeler shetlep o'tiledi(bejerilmeydi) ham basqariu bloginan tusqarig'a uzatiladi.

Satch gilt so'zi aniq try-blok tan keyin keledi ham ayriqsha jag'day juz bergende basqariw o'tiui kerek bolg'an kod bo'legin belgileydi.Kod bo'legi figuralı qauisqa alınadı ham catch-blok (ayriqsha jag'daylardı qayta isleushi)delinedi. catch gilt so'zinen keyin qauis ishine aling'an ayriqsha jag'daylar turi ham o'zgeriwshiden payda bolg'an ayriqsha jag'dayları keledi.

```
satch(<ayriqsha jag'daylar turi><ayriqsha jag'daylar o'zgeriwshisi>
//Ayriqsha jag'daylardı qayta isleu
}
```

Ayriqsha jag'day turinin' atında qayta isleytug'in ayriqsha jag'daylar turi ko'rsetiledi ham catch-blok tap usı turdegi ayriqsha jag'daylardı "ilip" aladı.Eger ayriqsha jag'daylar ilip aling'an bolsa, ayriqsha jag'daylar o'zgeriwshisi ayriqsha jag'daylar turi xaqqındag'ı mag'lumatı qabil qiladı. Eger ayriqsha jag'day xaqqındag'ı mag'lumatqa mutajlik bolmasa,ayriqsha jag'daylar o'zgeriwshisi isletilmesligi mumkin. Ayriqsha jag'daylar o'zgeriwshisi iqtıyariy turde bolıui mumkin. Bug'an paydalanıushi jaratqan klass turide kiredi.

Bir try-blok tan keyin bir neshe catch-blok keliui mumkin.Ayriqsha jag'daylar turinin' atında tochkalar bolg'an catch-blok iqtıyariy jag'daylardı ilip aladı ham ol try-blok tan keyin keliushi catch operatorlar royxatının' aqırında keliui kerek.

Ayriqsha jag'day auxalın qayta isleushi misaldı ko'reyik;

```

#include<iostream>
#include<exception>
using namespace std;
int AddivB(int A,int B) //Qate juz beriwı mumkin bolg'an funktsiya
{
//Ayriqsha jag'daylar juzege keletug'in jer
}
int main ()
{
int a, b, c;
cout<<"a=";cin>>a;
cout<<"b=";cin>>b;
try
{
c=AddivB(a, b);
cout<<" Ayriqsha jag'day juz bermedi\n";
cout<<"c="<<c;
return 0;
}
catch(int)
{
cout<<"int turdegi ayriqsha jag'daylar juz berdi\n";
return 1;
}
catch(...)
{

```

```

    cout<<"Qandaydir ayriqsha jag'day juz berdi\n";
return 2;
}
return 0;
}

```

Programmadaǵı AdivB() funktsiyası ayriqsha jag'daydı juzege keltiriwi mumkin, misalı no'lge bolıu. Funktsiyada juzege keletug'in ayriqsha jaqdaydı qayta isleu ushin eki catch-blok anıqlang'an. Birinshisinde int turdegi ayriqsha jag'day ilip alınadı. Eger basqa turdegi ayriqsha jag'day juz berse onı catch(...) operatori qayta isleydi.

Ayriqsha jag'daylardı juzege keltiriw: Programmadaǵı juzege keletug'in ayriqsha jag'daylar ham onın' turin ko'rsetiu ushin to'mende keltirilgen sintaksiske iye throw gilt so'zi isletiledi;

```
throw<tariyiplep beriw>
```

bul jerde <tariyiplep beriw>-esaplang'an mug'darı, sol manis turi menen anıqlang'an uaqtinshalıq obyektı initsializatsiya qılıushı manis.

throw operantı bolmaslıǵı mumkin. Bul ko'rinis tap usı uaqıtta qayta islenip atırg'an ayriqsha jag'day turindegi ayriqsha jag'daydı juzege keltiriw ushin isletiledi ham ol tek g'ana catch-blok ishinde isletiliui mumkin.

Programmanın' throw operantı jaylasqan jerine "ayriqsha jag'daydı juzege keltiriw tochkası" delinedi.

Ayriqsha jag'daydı juzege keltiriwge misal keltiremiz.

```

#include<iostream>
#include<exception>
//using std::cout;

```

```

using namespace std;

class Qate1{};

class Qate2 {};

int AdivB(int a, int b) // Qatelik juz beriwı mumkin bolg'an funksiya
{
    if(!b)throw Qate1();
    if(!a)throw Qate2();
    return a/b;
}

int main()
{
    int a, b, c;
    cout<<"a=";cin>>a;
    cout<<"b=";cin>>b;
    try
    {
        c=AdivB(a,b);
        cout<<"Ayrıqsha jag'day juz bermedi!\n";
        cout<<"c="<<c;
        return 0;
    }
    catch(Qate1 & Qate1)
    {
        cout<<"Qandayda ayrıqsha jag'day juz berdi!\n";
        return 1;
    }
}

```

```

}
catch(...)
{
cout<<"Qandayda ayriqsha jag'day juz berdi!\n";
return 2;
}
return 0;
}

```

Programma isleude, bas funktsiyadan AdivB() funktsiya shug'arilg'anda, eger

- b argument mug'dari 0 bolsa, Qate 1 turdegi ayriqsha jag'day juzege keltiriledi ham catch(Qate 1 & qate 1)

blok tarepinen ilip alinadi ham qayta islenedi;

- a argument mug'dari 0 bolsa, Qate 1 turdegi ayriqsha jag'day juzege keltiriledi ham catch(...) blok tarepinen ilip alinadi ham qayta islenedi;

- a ham b argumentler nolden parqli bolsa, ayriqsha jag'day juz bermeydi, hamde dastur o'z jumisin normal tamalaydi.

throw gilt so'zin tek g'ana ayriqsha jag'day juzege keltiriwi ushin, balkim bir funktsiya tarepinen juzege keltiriwi mumkin bolg'an ayriqsha jag'daylardi tariyipleu ushin isletiu mumkin. Ayriqsha jag'daylardi eariyipleu to'mendegi formatda beriledi:

```
throw (<tur 1><tur 2>....)
```

Ayriqsha jag'daylardi tariyipleu funktsiya parametrleri diziminen keyin keledi ham ol funktsiya qanday turdegi jag'day juzege keltiriwi mumkinligin ko'rsetedi. Biraq bul jag'day funktsiya basqa turdegi ayriqsha jag'daylardi juzege keltirmeydi degeni emes. Ko'rsetilgen turden parqli

juzege kelgen ayırıqsha jag'daylarg'a kutilmegen ayırıqsha jag'daylar delinedi ham olarda ayırıqsha jol menen qayta islenedi. Eger funktsiyag'a qanday bir ayırıqsha jag'day juzege keltiriwdi qadag'an qılıu zarur bolsa, turi ko'rsetilmegen ayırıqsha jag'day tariyipten paydalanıladı. Ekinshi tarepten, ayırıqsha tariyipke iye bolmag'an funktsiya xar qanday ayırıqsha jag'daydı juzege keltiriwi mumkin.

Mısal keltireyik.

```
#include<stdio.h>

bool test;

class Qate1 {};

void Func1(bool bul) throw(Qate1) {if(bul) throw Qate1();}

void Func2() throw()
{
try{Func1(true);}
catch (Qate1 & qate)
{
puts("Qate1 turdegi ayırıqsha jag'day qayta islenedi!" );
test=true;
}
}

int main()
{
try
{
test=false;
```

```

Func2();
test?puts("Func2() funksiyası ayrıqsha jag'dayın qayta isleydi");
puts("Xesh qanday ayrıqsha jag'day juz bermedi" );
}
catch(...) {puts("Func2()funksiyası ayrıqsha jag'dayın juzege
keltirdi!" );
return 0;
}}

```

Usı mısıl ayrıqsha jag'day tariyipin ayrıqsha jag'dayın funksiyadan tısqarig'a shıg'ıug'a jol qoymaslıg'ın ko'rsetedi.Usı sebepli, bas funksiyadag'ı catch(...) blogı xesh qashan isletilmeydi.Funksiya ishinde juzege keletug'ın ayrıqsha jag'daydı funksiya ishinde qayta isleniun mumkin.Eger Func1()funtsiyası Qate1 turinen parqlı ayrıqsha jag'daydı juzege keltirse, ol kutilmegen ayrıqsha jag'day boladı.

Ayrıqsha jag'daylardı ilip alıu: Programmanın' qandayda bir tochkasında ayrıqsha jag'day juzege kelse, programma orınlanıuı uziledi ham to'mendegiler juz beredi:

-eger tiykar turdegi o'zgeriwshi yaki mug'dar boyınsha klass obyektı payda etilgen bolsa,o'zgeriwshinin' kopiyası jaratıladı(obyekt ushın kopiyalau konstruktı isletiledi).

-eger murajat boyınsha o'zgeriwshi payda etilgen bolsa, kopiyalau juz bermeydi;

-payda etilgen o'zgeriwshi menen saykes keliushi parametrđi qabıl qıla alatug'ın en' jaqın qayta isleu blogı izlenedi;

-eger qayta isleu blogı tabılg'an bolsa, stek usı tochkag'a shekem "bosatıladi",bunda ko'rinisten shıqqan local obektler destruktoralı shaqırıladi;

-basqariw tabılg'an qayta isleu blogına uzatıladi;

-ayrıqsha jag'daydı qayta isleytug'in blok tabılmag'an jag'day ushin programma duziushisi programmanı tamalau qayta isleushisi-set_terminate() funktsiyasın shaqıriw arqalı programmanı tamamlauı mumkin, bolmas dizimin o'zi tamamlau funktsiyasın shaqıradi.

Ayrıqsha jag'dayg'a mass qayta isleu blogın izleude kompilyator to'mendegi qag'ıydalarg'a boysınadi.Ayrıqsha jag'dayg'a qayta isleushi mas esaplanadi;

-payda etilgen o'zgeriwshi turi qayta isleushi kutip atırg'an tur bolsa.

Basqasha aytqanda, eger payda bolg'an o'zgeriwshi turi T bolsa, parametrleri T, const T,T& ham const T& turide bolsa;

-payda bolg'an o'zgeriwshi qandaydir miyras klass obekti bolıp, qayta isleushi turi og'an qarata tayanish klass turinde ham miyraslıq public murajat tiykarı boyınsha payda etilgen bolsa.

Sonı qayt etiu kerek, kompilyator payda bolg'an o'zgeriwshi turi menen mas keliushi parametrđi qabil etetug'in en' jaqın qayta isleushini qıdıradi.Sol sebepli, try-blok ushin anıqlang'an catch-blok lar izbe-izligine itibar beriw zarur boladı.Tayanish klass ayrıqsha jag'daydı kutip atırg'an qayta isleushi miyras klass qayta isleushilerdi jasıradi. Tap usınday void* turindegi ko'rsetkish ushin anıqlangan qayta isleushi ıqtıyariy turdegi ko'rsetkish turdegi qayta isleushilerdi jasıradi.

Mısal ko'reyik.

```

#include<iostream.h>
class Tayanish {};
class Tuindi; public Tayanish {};
void Fun Tuindi Ayriqsha () {Tuindi tuindi:throw tuindi;}
void Fun_ Ayriqsha jag'day() {throw "Fun Ayriqsha jag'day ()
funksiyasında qate!\n";}

int main()
{
try
{
Fun_Tuindi_Ayriqsha jaqdayı ();
}
catch(Tuindi &) {cout<<"Tuindi & ayriqsha jag'dayı ilip alinadı!\n";}
catch(Tayanish &) {cout<<"Tayanish & ayriqsha jag'dayı ilip
alinadı!\n";}

try
{
Fun_Ayriqsha jag'day();
}
catch(const char*s)
{
cout<<"const char* ayriqsha jag'dayı ilip alındı!\n";
cout<<"Ayriqsha jag'day manzili"<<s<<"\n";
}
catch(void*) {cout<<"void" ayriqsha jag'dayı ilip alındı!\n";
return 0;
}

```

```
}
```

Programma isleui natiyjesinde ekrang'a to'mendegi qatarlar shıg'adı:

```
Tuındı & ayrıqsha jag'dayı ilip alındı!
```

```
const char* ayrıqsha jag'dayı ilip alındı!
```

Ayrıqsha jag'day manzili:Fun_Ayrıqsha jag'day() funksiyasında qate!

Eger programmada catch(Tayanış &) bloğı catch(Tuındı &) blogınan aldın jazılğ'an bolsa, ol tuındı klası turindegi qayta isleushisin jasıradı ham sol turdegi ayrıqsha jag'daydı o'zi ilip aladı.Tap usınday,catch(void*) bloğı catch(const char*s) bloktan aldın kelse, ko'rsetkish turdegi hamme ayrıqsha jag'daylardı ilip aladı,const char* turindegi ayrıqsha jag'daydı da.Joqarıdag'ı programmada bunday orın almastırıwlar qılınsa, ekrang'a

```
Tayanış & ayrıqsha jag'dayı ilip alındı!
```

```
void* ayrıqsha jag'dayı ilip alındı!
```

Xabarları ko'rsetiledi.

Ishpe-ish jaylasqan try-catch bloklar: C++ tili ishpe-ish jaylasqan try-catch bloklardı isletuige imkan beredi.Bunday ko'rinisler ushın jeke talap-xar bir try-blok tan keyin albette catch-blok keliui kerek.

Mısalı:

```
#include<iostream.h>
```

```
class Tayanış {}
```

```

void Fun_Tuindi_Ayriqsha jag'daylari(){Tuindi tuindi;throw tuindi;}
int main()
{
try
{
//...
try
{
Fun_Tuindi_Ayriqsha jag'daylari();
}
catch(Tuindi &){cout<<"Tuindi& Ayriqsha jag'daylari ilip
alindi!\n";}
return 0;
}

```

Programma bejirilgenda ekrang'a to'mendegi natijeler shug'adi;

Tuindi& ayriqsha jag'dayi ilip alindi!

Tayanish& ayriqsha jag'dayi ilip alindi!

Ayirim jag'daylarda ishpe-ish jaylasqan try-catch bloklar sirlı rauishte juzege keledi;ximayag'a aling'an funktsiyanın' denesinde ximayalang'an bloklar bolg'anda to'mendegi misaldı tap usınday jag'day ko'rsetilgen:

```

#include<iostream.h>
void Fun_Ayriqsha jag'daylari ()
{
//...

```

```

throw"Fun_Ayriqsha jag'day() funksiyasında qate!\n";
try
{
throw QateQatar;
}
catch(chars*) {cout<<s;}
Fun_Ayriqsha jag'day();
}
void main()
{
try
{
Fun_Fun_Ayriqsha();
}
catch(char*s) {cout<<s;}
}

```

Programma ekrang'a to'mendegi qatarlardı payda etedi.

Fun_Fun_Ayriqsha jag'day() funksiyasında qate!

Fun_Ayriqsha jag'day() funksiyasında qate!

Ishpe-ish try-catch bloklar isletiuqe qandayda funktsiyanın' kutilmegen ayriqsha jag'daydı juzege keltiriw mumkinligi sebep boladı. Bunday jag'daylarda ayriqsha jag'daydı sheshiudin' jollarınan biri-bas funktsiya denesinin' tolıqlıg'ınsha ximayag'a aliw ham onun' isleushisi sıpatında catch(...) ko'rinsin isletiu bolıp tabıladı.

Kutilmegen ayriqsha jag'daylar ham tamamlaudi qayta isleu:

Programma isleuinde funktsiya, onin' ayriqsha jag'day tariyipinde ko'rsetilmegen ayriqsha jag'daydi juzege keltirse, kutilmegen ayriqsha jag'day juzege keldi delinedi. Bul jag'dayda unexpected() funktsiyasi shaqiriladi. Ol o'z naubetinde kutilmegen ayriqsha jag'daydin' qayta isleushisin shaqiradi. Kelisiu boyinsha bul terminate() funktsiyasi esaplanadi. Terminate() funktsiyasinin' shaqiriluu programma orinlanu xareketin tamamlaug'a alip keledi.

unexpected() funktsiyasinin' to'mendegi sintaksiske iye:

```
void unexpected():
```

set_unexpected() funktsiyasi jardeminde programma duziushinin' o'zi kutilmegen ayriqsha jag'daylardi qayta isleushisi funktsiyani aniqlau mumkin. Bul funktsiyaning prototipi to'mendagi ko'rinisike iye;

```
unexpected_handlerset_unexpected(unexpected_handler  
ph)throw();
```

Bull jerde unexpected_handler kutilmegen ayriqsha jag'daydi qayta isleushi funktsiyag'a ko'rsetkish:

```
typedef void(*unexpected_handler)();
```

set_unexpected () funktsiyasi ph parametric ko'rsetip turqan funktsiyani kutilmegen ayriqsha jag'daydi jan'a qayta isleushisi qilip ornatadi ham aldindig'1 qayta isleushisinin' adresin qaytaradi. Jan'a qayta isleushi parametrga iye bolmaslig'1 kerek ham ol xesh qanday mug'dar qaytarmaydi. Bul funktsiya onni shaqirg'an kodqa basqariwdi qaytarmaydi ham ol to'mendegi jollardind' bir menen programmaning orinlanu xareketin tamamlaydi:

-Kutilmegen ayrıqsha jag'daydı juzege keltirgen funksiyanın' ayrıqsha jag'daylar tariyipinde keltirilgen turdegi ayrıqsha jag'daydı juzege keltiriw arqalı:

-bad_exception turdegi ayrıqsha jag'daydı juzege keltiriw arqalı;

-terminate (),abort() yaki exit(1) funksiyalardan birin shaqiriw arqalı.

Joqarıda qayt etilgenindey ayrıqsha jag'day qayta isleushisi terminate() funksiyanın shaqıradı.

```
bad_exception ayrıqsha jag'dayı
```

```
class bad_exception: public exception: public exception{};
```

ko'riniste anıqlang'an bolıp, ol unexpected_handler tarepinen juzege keltiriliui mumkin bolg'an ayrıqsha jag'daylardı tariyipleydi.

Kutilmegen ayrıqsha jag'daydı qayta isleuge misal keltiremiz:

```
#include<iostream>
```

```
#include<exception>
```

```
using namespace std;
```

```
void f() throw ()//Ayrıqsha jag'daydı juzege keltirmeytug'in funksiya
```

```
{
```

```
    throw"Kutilmegen ayrıqsha jag'day!"//Kutilmegen ayrıqsha jag'daydı juzege keltiriw
```

```
}
```

```

void Jeke_Qayta_isleushi()
{
cout<<"Kutilmegen ayriqsha jag'day qayta islendi";
exit(-1);
}

int main()
{
set_unexpected(Jeke_Qayta_Isleushi);
f();
cout<<"Bull Qatar xesh qashan dag'aza etilmeydi!";
return 0;
}

```

Programma ekrang'a

Kutilmegen ayriqsha jag'day qayta islendi!

Xabardi dag'aza etedi.

Endi juzege kelgen ayriqsha jag'dayg'a mas qayta isleushi tabilmag'anda qanday jag'day juz beriwin ko'zemiz. Bunday jag'daylarda kelisiu boyinsha terminate() funktsiyasi shaqiriladi. Oz naubetinde terminate() funktsiyasi, programmani toqtatug'a alip keliushi abort() funktsiyasin shaqiradi. Programmanin' tamamladi qayta isleushisin set_terminate() funktsiyasi quralinda ornatiu arqali programmani toqtatu xareketine o'zgeris kiriti mumkin.

set_terminate() funktsiyasi to'mendegi prototipke iye:

```

terminate_function set_terminate(terminate_function_terv_func);

```

bul jerde terminate_function prorammanın' tamalaushı funktsiyag'a ko'rsetkish:

```
typedef void(*terminate_function)();
```

Tamamlau funktsiya adresi-term_func parametric set_terminate() funktsiyasının' tek bir g'ana kiriwshi parametri esaplanadı ham aldın'g'ı tamamlau funktsiya adresi onın' qaytarıushı mug'darı boladı.Eger funktsiyanı shaqırıw set_terminate(NULL) ko'rinisinde bolsa, kelisiu boyınsha tamamlau funktsiyasın shaqırıw jag'dayı tiklenedi.

terminate() funktsiyası prototipi

```
void terminate();
```

ko'rinisinde de payda etip atırg'an tamamlau funktsiyalarıda tap usınday prototipke iye bolıuı shart.

Pamyattı tazalau menen baylanıslı zarur ameller orınlang'annan keyin terminate() funktsiyası programmadan shıg'ıu ushın exit() funktsiyasın shaqırıu kerek boladı(onı shaqırg'an funktsiyag'a yaki ayırıksha jag'day juzege kelgen kodqa qaytpaslıq ushın). Kerisinshe jag'dayda abort() funktsiyası shaqırıladı.

Mısal ko'reyik:

```
#include<iostream>
#include<exception.h>
using std::cout;
void Tamamlau_Funktsiyası()
{
    cout<<"Tamamlau_Funktsiyası() funktsiyası terminate() tarepinen
shaqırıldı!\n";
    exit(-1);
```

```

}
int main()
{
int i=10, j=0, rez;
set_terminate(Tamamlau_Funktsiyası);
try
{
if(j=0)throw"Nolge bo'liu";
else rez=i/j;
}
catch(int)
{
cout<<"int turindeki ayriqsha jag'day ilip alindi!\n";
return 0;
}
}

```

Programma islangennen keyin ekrang'a to'mendegi Qatar dag'aza etiledi:
Tamamlau_Funktsiyası() funktsiyası terminate() funktsiyası tarepinen
shaqirildi.

Sorawlar ha'm tapsırmalar

1. Programma orinlanuunda ayriqsha jag'day qashan juz beredi?
2. Ayriqsha jag'daylardı basq'ariw ushin C++ tilinde ush gilt so'zlerin sanap ko'rsetin'.
3. Programmadağ'ı juzege keltiriw ham qayta isleu mexanizmi qanday?

4. *Ayriqsha jag'daylardı ilip aliuda ayriqsha jag'daylardı ilip alıu blogın anıqlauda parametr roli neden ibarat?*

5. *Kutilmegen ayriqsha jag'daylar qashan juz beredi?*

Exception Handling Fundamentals

C++ exception handling is built upon three keywords: **try**, **catch**, and **throw**. In the most general terms, program statements that you want to monitor for exceptions are contained in a **try** block. If an exception (i.e., an error) occurs within the **try** block, it is thrown (using **throw**). The exception is caught, using **catch**, and processed. The following discussion elaborates upon this general description.

Code that you want to monitor for exceptions must have been executed from within a **try** block. (A function called from within a **try** block is also monitored.) Exceptions that can be thrown by the monitored code are caught by a **catch** statement, which immediately follows the **try** statement in which the exception was thrown. The general form of **try** and **catch** are shown here:

```
try {
    // try block
}
catch (type1 arg) {
    // catch block
}
catch (type2 arg) {
    // catch block
}
catch (type3 arg) {
    // catch block
}
// ...
catch (typeN arg) {
    // catch block
}
```

The **try** block must contain the portion of your program that you want to monitor for errors. This section can be as short as a few statements within one function, or as all-encompassing as a **try** block that encloses the **main()** function code (which would, in effect, cause the entire program to be monitored).

When an exception is thrown, it is caught by its corresponding **catch** statement, which then processes the exception. There can be more than one **catch** statement associated with a **try**. The type of the exception determines which **catch** statement is used. That is, if the data type specified by a **catch** statement matches that of the exception, then that **catch** statement is executed (and all others are bypassed). When an exception is caught, *arg* will receive its value. Any type of data can be caught, including classes that you create.

The general form of the **throw** statement is shown here:

```
throw exception;
```

throw generates the exception specified by *exception*. If this exception is to be caught, then **throw** must be executed either from within a **try** block itself, or from any function called from within the **try** block (directly or indirectly).

Look carefully at this program. As you can see, there is a **try** block containing three statements, and a **catch(int i)** statement that processes an integer exception. Within the **try** block, only two of the three statements will execute: the first **cout** statement and the **throw**. Once an exception has been thrown, control passes to the **catch** expression, and the **try** block is terminated. That is, **catch** is *not* called. Rather, program execution is transferred to it. (The program's stack is automatically reset, as necessary, to accomplish this.) Thus, the **cout** statement following the **throw** will never execute.

After the **catch** statement executes, program control continues with the statements following the **catch**. Thus, it is the job of your exception handler to remedy the problem that caused the exception, so that program execution can continue normally. In cases where the error cannot be fixed, a **catch** block will usually end with a call to **exit()** or **abort()**, or otherwise terminate program execution. (The **exit()** and **abort()** functions are described in the In Depth box.)

As mentioned earlier, the type of the exception must match the type specified in a **catch** statement. For example, in the preceding program, if you change the type in the **catch** statement to **double**, then the exception will not be caught, and abnormal termination will occur. This change is shown here:

Catching Class Types

An exception can be of any type, including class types that you create. Actually, in real-world programs, most exceptions will be class types, rather than built-in types. Perhaps the most common reason that you will want to define a class type for an exception is to create an object that describes the error that occurred. This information can be used by the exception handler to help it process the error. The following example demonstrates this.

Using Multiple catch Statements

As stated earlier, you can associate more than one **catch** statement with a **try**. In fact, it is common to do so. However, each **catch** must catch a different type of exception. For example, the program shown here catches both integers and character pointers:

Options for Exception Handling

There are several additional features and nuances to C++ exception handling that make it easier and more convenient to use. These attributes are discussed here.

Catching All Exceptions

In some circumstances, you will want an exception handler to catch all exceptions, instead of just a certain type. This is easy to accomplish. Simply use this form of **catch**:

```
catch(...) {  
    // process all exceptions  
}
```

Here, the ellipsis matches any type of data.

The following program illustrates **catch(...)**:

Rethrowing an Exception

If you want to rethrow an exception from within an exception handler, you can do so by calling **throw** by itself, with no exception. This causes the current exception to be passed on to an outer **try/catch** sequence. The most likely reason for calling **throw** this way is to allow multiple handlers to have access to the exception. For example, perhaps one exception handler manages one aspect of an exception, and a second handler copes with another aspect. An exception can be rethrown only from within a **catch** block (or from any function called from within that block). When you rethrow an exception, it will not be recaptured by the same **catch** statement. It will propagate to the immediately enclosing **try/catch** sequence.

37-Leksiya. Shablonlar

Joba:

1. Ulumalasqan programmalastırıw
2. Funktsiyalar shablonları
3. Klaslar shablonı
4. Shablonlardıń standart bibliotekası (STL). STL tiykarg'ı konsepsiyası
5. Konteynerler
6. Iteratorlar
7. Konteynerler ulıwma qa'siyetleri
8. Algoritmeler

Ulumalasqan programmalastırıw: Bizge belgili, obektke bag'darlang'an programmalastırıwda itibar berilgenlerge qaratıladı, yag'nıy obektke anıqlawshı berilgenler xam olar ustinde orınlanatug'ın metodlardı (funktsiyalardı) anıqlaydı. Og'an qarama-qarsı *ulumalasqan programmalastırıwda* itibar algoritmelerge qaratıladı. Ulıwmalaskan programmalastırıw makseti berilgenler turlerine baylanıslı bolmagan programma kodın jazıw. S++ tilindegi shablonlar ulıwmalaskan programmalastırıwdın kuralı esaplanadı.

Shablonlar – berilgenler turlerinen, funktsiyalar xam klaslar anıqlawshı paydalanıw manisinde ulıwmalastırıwga imkan beriushi tusinikler esaplanadı. Sol sebepli olardı *parametrlestirilgen funktsiyalar* yamasa *parametrlestirilgen klaslar* delinedi. Ayrım uakıtlarda “*shablonlı funktsiyalar*” xam “*shablonlı klaslar*” terminleri isletiledi.

Funktsiyalar shablonları: *Funktsiya shablonı* funktsiyanın ulıwmalaskan anıqlanıwı bolıp, onın tiykarında kompilyator paydalanıushı tarepinen berilgen turdegi funktsiya uakilin jaratadı.

Funktsiya shablonının dagazasının sintaksisi tomendegishe:

```
Template<class T1 | T1<identifiktor1>,  
<class T2 | T2<identifiktor2>,>
```

```

...
<class Tn|Tn<identifiktor n>>
<qaytariwshi manis turi><funksiya ati>( <parametrler dizimi>)
{
//funksiya denesi
}

```

Bul jerde template gilt sozinen keyin muyeshli kausırma ishinde bir-birinen utir menen ajratılğan parametrler dizimi keledi. Xar bir parametr-class gilt sozi yamasa tur atı xam odan keyin keliushi identifikatordan ibarat boladı. Ayırım xallarda, berilgenlerdin parametrlesken turin beriu ushın class sozi ornuna typename gilt sozi isletiliui mumkin.

Dagazadagı class yamasa typename gilt sozinen keyingi shablon parametrlerine *parametrlesken turler* deyiledi. Olar kompilyatorga shablonda parametr sıpatında kaday berilgenler turi isletilip atırğanın bildiredi. Tur atı xam identifikatordan ibarat shablon parametri kompilyatorga shablon parametri korsetilgen turdegi konstanta ekenligin bildiredi.

```

#include<iostream>
template <class T>
T Kvadrat(T x)
{ return x*x; }
template<class T>
T*Almastiriw(T*t, int ind1, int ind2)
{
T waqitsha =t[ind1]; t[ind2]; t[ind2]=waqitsha;
return t;
}
template<typename T1, typename T2 >
void ekranga(T1 x,T2y)
{ cout<<x<<"\t"<<y<<endl; }
template <class T,int jiljiw>
obekt_Adresi(T*obj,unsigned int *pAdres)
{ *pAdres=(unsigned int) & obj[0]+jiljiw*sizeof(T); }
int main()

```

```

{
int n=10,kv_n,i=1,j=3;
double d=10.21,kv_d;
char*Qatar="shablon";
kv_n=Kvadrat(n); kv_d= Kvadrat(d);
int Massiv[10];
unsigned int adres=0;
cout<<"n="<<n<<"Kvadrat n="<<kv_n<<endl;
cout<<"d="<<d<<"Kvadrat d="<<kv_d<<endl;
cout<<"qatar="<<qatar<<"ozgergen
qatar="<<almastiriw(qatar,i,j)<<endl;
cout<<"san ham onin kvadrati:\n";
ekranga(n,d);
knekranga(kv_n,kv_d);
obekt_Adresi<int,5>(Massiv,&adres);
cout<<"arr[5]element adresi="<<hex<<showbase<<adres;
return 0;
}

```

Programma isleui natiyjesinde ekranga tomendegiler shıgarıladı:

n=10 Kvadrat n=100

d=10 Kvadrat d=104.244

qatar=shablon ozgergen qatar=shablon

san ham onin kvadrati:

10 10.21

100 104.244

Arr[5] element adresi=0x12ff50

Tap apiuayı funktsiyalarday funktsiyalar shablonının prototipin dagaza kılıu mumkin.Mısalı:

```
template<class T,class S> //funktsiya shablони prototipi
```

```
void Fun(T,S);
```

...

```
template<class U,class V> //funktsiya shablони aniqlaniwi
```

```
void Fun(U,V)
```

```
{
    // funksiya denesi
}
```

Funktsiya shablonının prototipindegi xar bir parametr anıqlanıuında parametr sıpatında isletiliui kerek. Mısalı, tomendegi jagday katelikke alıp keledi:

```
template<class T, class S>
T Fun(S);
```

Funktsiya shablonının parametrlerinin atları birdey bolmaui kerek. tomendegi dagaza kate esaplanadı:

```
template<class T, class T>
T Fun(T, T);
```

Shablonlı funksiya sırtkı, statik xam jaylasıushı dep dagazalanıui mumkin. Bunın ushın saykes gılt sozler template katarının akırında jazılıui kerek boladı:

```
template<class T, class S>extern // sırtkı shablonlı funksiya prototipi
void Fun2(T, S);
template<class T>static // statik shablonlı funksiya prototipi
    T Fun2(T);
template<class T, class S>static // jaylasıushı shablonlı funksiya prototipi
T*Fun3(T, S);
```

Shablonlı funksiyanın xar bir tur menen birinshi shakırılıuında onın uakili jaratıladı xam bul jarayanga *shablonlı funksiyanı konkretleu* deyiledi. Jokarıda keltirilgen programmadagı

```
kv_n=Kvadrat(n); kv_d=Kvadrat(d);
```

korsetpelerinin birinshisi putin tur ushın Kvadrat() funksiyanın ekinshi korsetpeni orınlau ushın bolsa funksiyanın xakiykiy tur ushın uakilin jaratadı.

Funktsiya shablonın konkretlestiriude onın parametrleri ushın baska turden ashkara ravishte kutilgen turge keltiriü amelin orınlau mumkin.

Mısalı:

```
template<class T>
```

```
void Fun1(T) {...};
```

...

```
void Fun2(char ch) {Fun1<int>(ch); }
```

Bul mısalda kompilyator char turindegi ch ozgeriushisin int turine otkizedi.

Klaslar shablони: C++ tilinde klaslar shablonın jaratıu da kollap kuuatlanadı. Abstrakt turdegi klass shablonı anıklangannan keyin ol konkret klaslardı jaratıuda kollanıluı mumkin. Kompilyator tarepinen klass shablonı xam shablon argumentleri boyınsha konkret klastı anıklau jarayanına *shablondı instansiyalau* deyiledi (*template instantiation*).

Mısal ushın tegisliktegi nokattı karayık. Nokat koordinatası double turinde bolgan Noqat klasın jaratıuımız mumkin. Baska bir atamada nokattın putin sanlar koordinatalarında int turinde anıklau zarur bolıuı mumkin. Tap sonday koordinatalar turleri short, unsigned char turinde talap kılınıuı mumkin.

Xar bir jagday ushın zarur turdegi koordinataları bolgan Noqat klasın anıklau arkalı maselelerdi sheshıu mumkin. Bul albette mashkalanın makul sheshimi emes. Durıs sheshim – Noqat klasının shablonın jaratıu esaplanadı.

Klass shablonın anıklau tomendegi koriniske iye:

```
template<shablon parametrleri> class <klass ati> { <shablon denesi> }
```

Mısal ushın Nokat klasının shablonı anıklanıuı

```
template<class T>class Noqat
```

```
{
```

```
public:
```

```
Noqat(T_x=0,T_y=0):x(_x),y(_y){}
```

```
void baspag'a_shig'ariw() const{cout<<"("<<x<<" "<<y<<"")"<<endl;}
```

```
private:
```

```
T x,y;
```

```
};
```

Bul jerde T – kandaydur abstrakt tur bolıp, onın anıklanıuındagı class sozi klastı emes, balki T – shablon parametri ekenligin bildiredi.

Shablon ishindegı funksiya – agzalardıń anıqlanıuı tap apıuayı klaslar anıqlanıuında boladı. Eger funksiya – agzalar anıqlanıuı klas anıqlanıuınan sırtta bolatugın bolsa, olardıń sintaksisi kuramalasadı. Mısalı, Noqat klasının baspag'a_shig'ariw() funksiya – agzasının klastan sırttaǵı anıqlanıuı tomendegishe boladı:

```
Template<class T> void class Noqat<T>::baspag'a_shig'ariw() const
{cout<<"("<<x<<" ,"<<y<<"")<<endl;}
```

Programma kuramına klass shablonı kiritilgende abstrakt tur (T)ornına konkret bir tur korsetilip klass nuskası jaratılmagansha birde-bir klass jaratılmaydı. Abstrakt tur ornına konkret tur koyıuga *shablondı aktuallau* deyiledi. Bul jarayan tap apıuayı klaslar obektin dagaza kılıu sıyaklı amelge asırıladı, mısalı:

```
Noqat<int>noqat_int(10,5);
```

yamasa aktuallastırılğan shablon Turine korsetkish dagaza kılıu xam ogan new arkalı adres beriu uaktında: Noqat<double*>noqat_k=new Noqat<double>(3.14,2.7);

Tomende klass shablonı isletilgen programma keltirilgen.

```
#include<iostream>
using namespace std;
template <class T>class Noqat
{
public:
Noqat(T_x=0,T_y=0):x(_x),y(_y){}
void baspaga_shigariw() const {cout<<"("<<x<<" ,"<<y<<"")<<endl;}
private:
T x,y;
};
int main()
{
Noqat<int> noqat_i;
Noqat<double> noqat_d(3.14,2.7);
Noqat<short> noqat_s(3,2);
noqat_i.baspaga_shigariw();
```

```

noqat_d.baspaga_shigariw();
noqat_s.baspaga_shigariw();
return 0;
}

```

Klass shablonlari klaslar konteynerlerinin (baska obektlerin saklauga molsherlengen obektler) standart bibliotekalarin (STL) jaratuidaken kollarilgan. Konteynerler vektorlar, dizimler, naubetler, koplikler xam sogan uxsas standart duzimler menen isleuge molsherlengen.

Shablonlardin standart bibliotekasi (STL). STL tiykargi konsepsiyasi

C++ tili kuramına kiriushi STL bibliotekasi algoritmlerdin ulumalaskan xalda aniklauga imkan beredi. Soni aytip otu kerek, STL bibliotekasi obektke bagdarlangan programmalastiru ulgisi emes, ol shablon klaslar jardeminde aniklangan bolsa da. Baskasha aytkanda, STL bibliotekasi ulumalaskan programmalastiru timsali bolip, paydalanushiga "*ol(STL bibliotekasi) kaday amellerdi (funktsiyalardi) orinlay aladi xam olar kaday amelge asirilgan*" siyakli kiziktiradi.

STL eki tiykargi boleklardan ibarat: konteyner klaslar jıynagi xam ulumalaskan algoritmler jıynagi.

Konteynerler - bul baska bir topardagi obektlerdi oz ishine algan obektler esaplanadi. Konteyner klaslar shablonlar esaplanadi, sol sebepli larda saklanatugin obektler ornatilgan yaki paydalanushı tarepinen aniklangan turlerde boluu mumkin. Bul obektlerge karaganda nuskalau xam manis beriu amellerin kollau mumkin. Konteyner klaslar berilgenlerdin ulgili turleri - stek, dizim, naubet xam baska turleri ustindegi amellerdi oz ishine aladi.

Ulumalaskan algoritmler – konteynerge kollanatugin kop sandagi amellerdin, misali, izleu, tartipleu, kosiu xam sol siyakli amellerdin orinlanuu esaplanadi. Birak olar konteyner klaslarinin metodlari esaplanbaydi. Olar STL bibliotekasindagi global shablon funktsiyalar korinisinde boladi xam sonin esabına universallikka erisiledi: bul funktsiyalar tek gana turli konteyner klaslar obektlerine kollanıp kalmastan, olardi turli topardagi elementlerge iye massivlerge de kollau

mumkin. Algoritmaların konteyner türlerine karaganda baylanıslı bolmaui olardı konteynerler menen kurallı baylanıslı esabına boladı: funktsiyaga konteyner uzatılmaydı, balki qayta islenetugin elementtin (diapazondı anıklaushı first xam last) adresler juplığı uzatıladı. Bul oz-ara tasir mexanizmi iteratorlardan paydalanıu esabına amelge asırıladı.

Iteratorlar – bul korsetkishlerdin ulıumalastırılğan konsepsiya esaplanadı: olar konteyner elementlerine korsetedi(murajat kıladı). Olarga karaganda apiuayı korsetkishlerdegi inkrement amelin kollau mumkin (konteyner boylap izbe-iz xareketleniu ushın), '*' ameli menen maniske murajat kılıu mumkin.

Konteynerler: Konteynerlerdi eki toparga bolıu mumkin: izbe-iz xam assotsiativ.

Izbe-iz konteynerler bir turdegi shekli obektlerdi uzliksiz izbe-izlik korinisinde saklaudu tamiynleydi. Tayanış izbe-iz konteynerlerge vektorlar (vector), dizimler (list), xam eki tarepli naubetler (deque), bunnan tıskarı arnaulı konteynerler (konteynerler adaptirleri) stekler (stack), naubetler (queue) xam jokarılı naubetler (priority_queue) kiredi.

Programmada konteynerlerden paydalanıu ushın saykes at fayldı kosıu kerek. Konteynerlerde saklanatugin obektler turi shablon argumenti jardeminde beriledi.

Mısalı:

```
vector <int> Avec; //int turindegi Avec vektori jaratıladı
```

```
list <Student> Gruppa; //Student turindegi dizimdi jaratıw
```

Assotsiativ konteyner gilt boyınsha berilgenlerge murajat kılıudu tamiynleydi. Bes turde assotsiativ konteynerler bar: lugatlar (map), dublikatlı lugat (multimap), koplikler (set), dublikatlı koplikler (multiset) xam bitli koplikler (bitset).

Iteratorlar: Konteyner elementleri turli koriniste yadta jaylasıuı mumkin, mısalı, massivlar yadta izbe-iz uzliksiz jaylasadı, onda bir elementten ekinshisine otıu korsetkish arifmetikası arkalı amelge asırılıuı mumkin. Sızıklı dizim korinisinde obektler ushın korsetkishler arifmetikasın kollap bolmaydı, sebebi dizimnin naubettegi elementin yadtın kalegen adresinde jaylasıuı mumkin, yagnıy dizim elementleri

yadta uzliksiz jaylasıuı mumkin.Sol sebepli xar bir klass ushın ozinin iteratorı bolıuı kerek.Xamme STL konteyner klasları ushın iterator turi anıqlangan bolıp, birak olardı amelge asırıu turlishe.Mısalı, vector klassı ushın ondagı elementler izbe-iz jaylaskanlıgı ushın iterator typedefT* iterator korinisinde anıqlanadı, list klası ushın iterator bolekl bir klass korinisinde amelge asırıladı.

Xar bir iterator tarepinen orınlanatugın tiykargı ameller:

- iterator manisin alıu: *p*- iterator, **p*-ol murajat kılátugın obekt manisi;

- bir iteratordı ekinshisine manis sıpatında berıu;

- iteratorlardı tenlik (==) xam ten emeslikke tekserıu(!=);

- konteyner boyınsha prefiks rauishte (++r) yamasa postfiks (r++) xareketlenıu.

Xar bir klass ushın iterator, korınıu oblastına ruxsat berıu (“::”) arkalı korsetiledi.

Mısalı,

```
vector<int>::iterator iter1;
```

```
list<Student>::iterator iter2;
```

Konteyner elementlerin “*aylanıp shıguı*” ushın takırarlau operatorı ozine saykes koriniste anıqlanadı: Eger *i*-kandaydur iterator bolsa

```
for(i=0;i<n;i++)
```

ornına

```
for(i=first;i!=last;++i)
```

jazıladı. Bul jerde first – konteyner birinshi elementine korsetkish manisi,last–konteyner akırgı elementinen keyin “*anıq emes*” elementke korsetkish manisi, ‘<’ ameli ‘!’ ameli menen almastırılğan.

Konteynerlerdin uliuma kasiyetleri: Tomende kopshilik konteyner klaslarda typedef jardeminde anıqlanátugın turler dizimi keltirilgen.

STLde anıqlangan unifikatsiya kılınğan turler

Maydan	Tusindirme
value_type	Konteyner elementi turi
size_type	Indeksler turi,elementler esaplagışı x.t.b (unsigned int)
iterator	Iterator

const_iterator	Ozgermes iterator
Reference	Elementke korsetkish
const_reference	Elementke ozgermes korsetkish
key_type	Gilt turi(assotsiativ konteynerler ushin)
key_compare	Salıstıru kriteriyası turi(assotsiativ konteyner ushin)

Xamme konteynerler ushin ulıumalıq bolgan ameller xam metodlar

Amel yamasa metod	Tusindirme
==, !=	true yamasa false qaytaradı
=(ma'nis beriw)	Bir konteynerdi ekinshisine nuskalaydı
clear	Xamme elementlerdi oshiredi
insert	Bir elementti yamasa elementler diapazonın kosıu
erase	Bir elementti yamasa elementler diapazonın oshiriu
size_type size() const	Elementler sanın qaytaradı
size_type max_size() const	Konteynerdin mumkin bolgan maksimal olshemin qaytaradı
bool empty() const	Konteyner bos bolsa, true qaytaradı
iterator begin()	Iteratordı konteyner basına qaytaradı
iterator end()	Iteratordı konteyner akırına otkeredi
reverse iterator begin()	Reversiv iteratordı konteyner akırına otkeredi(duris jonelistegi iteratsiya tausıladı)
reverse iterator end()	Reversiv iteratordı konteyner basına qaytaradı(keri jonelistegi iteratsiya tausıladı)

Algoritmler: *Algoritm* – konteyner (konteynerler) elementleri ustinde kandaydur amellerdi orınlaytugın funksiya.Ulumalaskan algoritmlerdi programmada kollau ushin <algorithm> atın faylın kiritiu kerek.

STL algoritmleri

Algoritm	Uazyypası
accumulate	Berilgen diapazondagı elementler jıyndısın esaplau
copy	Birinshi elementten baslap izbe-izlikti nuskalau
count	Izbe-izlikke kiriushi manisler mugdarın esaplau
count_if	Izbe-izlikdegi shart orınlanıushı elementler sanı
equal	Eki izbe-izliktin saykes elementleri juplıgının tenligin tekseriu
fill	Izbe-izliktin xamme elementlerin berilgen manis penen almasıru
Find	Manistin izbe-izlikke birinshi kirisin tabıu
find_first_of	Bir izbe-izliktin ekinshi izbe-izliktegi birinshi kirisin tabıu
find_if	Izbe-izliktegi berilgen shartke saykes keliushi birinshi elementin tabıu
for_each	Izbe-izliktin xar bir elementi ushin funktsiyanı shakıru
Merge	Tartiplengen izbe-izliklerdi kosıu
Remove	Berilgen manisli elementlerdi koshiriu
Replace	Berilgen manisli elementlerdi almasıru
Search	Bir izbe-izlikti ekinshi izbe-izlikke birinshi kirisin tabıu
Sort	Tartiplestiriu

Swap	Eki elementti oz-ara almastiruv
Transform	Izbe-izlikтин xar bir elementi ustinde berilgen ameldi ornlau

Dizimde keltirilgen algoritmlerdin basındađı eki parametrleri qayta islenetugin elementlerdin [first, last) – yarım interval korinisindegi diapazonın beredi, bul jerde first – diapazon basın korsetiushi iterator, last – diapazonnan sırtka shıkkanlıktı korsetiushi iterator.

Mısalı, eger sanlar massivı

```
int massiv[7]={5,9,-3,6,1,3,2}
```

korinisinde anıqlangan bolsa, onı sort algoritmı menen tartipleu tomendegi koriniste amelge asırıladı :

```
sort(massiv,massiv+7);
```

Bul jerde massiv iterator sıpatında isletiliushi int* turindegi korsetgish.

Mısal.

```
#include<iostream.h>
#include<fstream.h>
#include<vector.h>
#include<algorithm.h>
int main()
{
ifstream in("sanlar.txt");
if(!in) {cerr<<"fayl tabilmadi!"; exit(1); }
vector<int> v;
int x;
while(in>>x) v.push_back(x);
sort(v.begin(), v.end());
vector<int>::const_iterator i;
for(i=v.begin(); i!=v.end(); ++i)
cout<<*i<<" ";
return 0;
}
```

Generic Functions

A generic function defines a general set of operations that will be applied to various types of data. The type of data that the function will operate upon is passed to it as a parameter. Through a generic function, a single general procedure can be applied to a wide range of data. As you probably know, many algorithms are logically the same no matter what type of data is being operated upon. For example, the Quicksort sorting algorithm is the same whether it is applied to an array of integers or an array of **floats**. It is just that the type of data being sorted is different. By creating a generic function, you can define the nature of the algorithm, independent of any data. Once you have done this, the compiler will automatically generate the correct code for the type of data that is actually used when you execute the function. In essence, when you create a generic function, you are creating a function that can automatically overload itself.

A generic function is created by using the keyword **template**. The normal meaning of the word “template” accurately reflects its use in C++. It is used to create a template (or framework) that describes what a function will do, leaving it to the compiler to fill in the details, as needed. The general form of a template function definition is shown here:

```
template <class Ttype> ret-type func-name(parameter list)
{
    // body of function
}
```

Here, *Ttype* is a placeholder name for a data type used by the function. This name can be used within the function definition. However, it is only a placeholder that the compiler will automatically replace with an actual data type when it creates a specific version of the function. Although the use of the keyword **class** to specify a generic type in a **template** declaration is traditional, you may also use the keyword **typename**.

The following example creates a generic function that swaps the values of the two variables with which it is called. Because the general process of exchanging two values is independent of the type of the variables, it is a good candidate for being made into a generic function.

You can define more than one generic data type in the **template** statement by using a comma-separated list. For example, this program creates a template function that has two generic types:

Even though a generic function overloads itself as needed, you can explicitly overload one, too. This is formally called *explicit specialization*. If you overload a generic function, then that overloaded function overrides (or “hides”) the generic function relative to that specific version. For example, consider the following, revised version of the first example in this chapter:

As you can see, the new-style syntax uses the **template**<> construct to indicate specialization. The type of data for which the specialization is being created is placed inside the angle brackets following the function name. This same syntax is used to specialize any type of generic function. While there is no advantage to using one specialization syntax over the other at this point in time, the new-style is probably a better approach for the long term.

Explicit specialization of a template allows you to tailor a version of a generic function to accommodate a unique situation—perhaps to take advantage of some performance boost that applies to only one type of data, for example. However, as a general rule, if you need to have different versions of a function for different data types, you should use overloaded functions rather than templates.

Overloading a Function Template

In addition to creating explicit, overloaded versions of a generic function, you can also overload the **template** specification, itself. To do so, simply create another version of the template that differs from any others in its parameter list. For example:

Generic Classes

In addition to generic functions, you can also define a generic class. When you do this, you create a class that defines all the algorithms used by that class; however, the actual type of the data being manipulated will be specified as a parameter when objects of that class are created.

Generic classes are useful when a class uses logic that can be generalized. For example, the same algorithms that maintain a queue of integers will also work for a queue of characters, and the same mechanism that maintains a linked list of mailing addresses will also maintain a linked list of auto part information. When you create a generic class, it can perform the operation you define, such as maintaining a queue or a linked list, for any type of data. The compiler will automatically generate the correct type of object, based upon the type you specify when the object is created.

The general form of a generic class declaration is shown here:

```
template <class Ttype> class class-name {  
    .  
    .  
    .  
}
```

Here, *Ttype* is the placeholder type name, which will be specified when a class is instantiated. If necessary, you can define more than one generic data type by using a comma-separated list.

Once you have created a generic class, you create a specific instance of that class by using the following general form:

```
class-name <type> ob;
```

Here, *type* is the type name of the data that the class will be operating upon. Member functions of a generic class are, themselves, automatically generic. You don't need to use **template** to explicitly specify them as such.

In the following program, the **queue** class (first introduced in Chapter 11) is reworked into a generic class. Thus, it can be used to queue objects of any type. In this example, a character queue and a floating-point queue are created, but any data type can be used.