

O'ZBEKISTON RESPUBLIKASI
OLIY VA O'RTA MAXSUS TALIM VAZIRLIGI TERMIZ
DAVLAT UNIVERSITETI
FIZIKA – MATEMATIKA FAKULTETI
“ALAMIY MATEMATIKA” KAFEDRASI



“Borland C++ Builder dasturlash muhitida chop etish jarayonini boshqaruvchi
dasturiy ilova yaratish”

BITIRUV MALAKAVIY ISHI

Bajardi: 5130200 – “amaliy
Matematika va informatika”
Talim yo'nalishi bitiruvchisi
4-kurs talabasi Shaydullayev Sh.H

Ilmiy rahbar: Umarzoda Sh.A

Bitiruv malakaviy ishi kafedradan dastlabki himoyadan o'tdi.

_____ sonli bayonnomasi “__” “_____” 2018

Termiz – 2018 yil

Kirish.

I bob. Chop etish jarayonini boshqaruvchi dasturiy ilova yaratishning nazariy yoritilishi.

1.1 Borland C++ Builder dasturlash muhitida chop etish jarayonini boshqaruvchi dastur yaratish uchun zarur algoritmlar.

1.2 Chop etish jarayonini boshqaruvchi dastur yaratish davomida qo'llaniladigan C++ Builder komponentalari.

II bob. Borland C++ Builder dasturlash muhitida chop etish jarayonini boshqaruvchi dasturini yaratish texnologiyasi.

2.1 Chop etish jarayonini boshqaruvchi dasturning algoritmlari.

2.2 Borland C++ Builder dasturlash muhitida chop etish jarayonini boshqaruvchi dastur kod qismi.

2.3 Dasturiy taminot interfeysini ishlab chiqish va amaliyotda qo'llash sohalari.

Xulosa.

Foydalanilgan adabiyotlar.

Mundarija:

<u>Kirish</u>	
<u>I bob. Chop etish jarayonini boshqaruvchi dasturiy ilova yaratishning nazariy yoritilishi</u>	
<u>1.1 Borland C++ Builder dasturlash muhitida chop etish jarayonini boshqaruvchi dastur yaratish uchun zarur algoritimlar</u>	
<u>1.2 Chop etish jarayonini boshqaruvchi dastur yaratish davomida qo'llaniladigan C++ Builder komponentalari</u>	
<u>II bob. Borland C++ Builder dasturlash muhitida chop etish jarayonini boshqaruvchi dasturini yaratish texnologiyasi</u>	
<u>2.1 Chop etish jarayonini boshqaruvchi dasturning algoritimlari</u>	
<u>2.2 Borland C++ Builder dasturlash muhitida chop etish jarayonini boshqaruvchi dastur kod qismi</u>	
<u>2.3 Dasturiy taminot interfeysini ishlab chiqish va amaliyotda qo'llash sohalari</u> ...	
<u>Xulosa</u>	
<u>Foydalanilgan adabiyotlar</u>	

Kirish

Axborot texnologiyalari sohasi rivojlangan mamlakatlar qatorida O'zbekiston Respublikasida xam zamonaviy kompyuter va axborot texnologiyalari sohasida katta ishlar amalga oshirilmoqda. Fan va ta'limning barcha soxalariga infokommunikatsion texnologiyalarni keng miqyosda qo'llash, yuqori malakali dasturchi mutaxassislar tayyorlash darajasini oshirish, xalqoro axborot tizimlarini joriy etish, shu jumladan Internetdan kengaytirish maqsadida O'zbekiston Respublikasi Prezidentining 9 ta farmon va qarorlari, 18 ta hukumat qarorlari va 45 dan ortiq boshqa normativ-huquqiy hujjatlar qabul qilingan. Axborot-kommunikatsiya texnologiyalarini rivojlantirishga yunaltirilgan O'zbekiston Respublikasi Prezidentining farmoni va qarorlarini keltiramiz: 1997 yil 27 iyulda "Axborot tizimlari sohasini qayta tashkil etish va boshqarishni takomillashtirishga oid chora-tadbirlar to'g'risida" gi PF-1823-sonli farmoni, 2002-yil 30-mayda "Kompyuterlashtirishni yanada rivojlantirish va axborot kommunikatsiya texnologiyalarini joriy etish to'g'risida" gi PF – 3080 sonli farmoni, 2005 yil 2 iyunda "Axborot texnologiyalari sohasida kadrlar tayyorlash tizimini takomillashtirish to'g'risida"gi PQ-91-sonli qarori, 2005 yil 8 iyulda "Axborot-kommunikatsiya texnologiyalarini yanada rivojlantirishga oid qo'shimcha chora-tadbirlar to'g'risida"gi PQ-117-sonli qarori, 2005 yil 28 sentyabrda "O'zbekiston respublikasining jamoat ta'lim axborot tarmog'ini tashkil etish to'g'risida" gi PQ-191-sonli qarori, 2007 yil 3 aprelda "O'zbekiston Respublikasida axborotni kriptografik muhofaza qilishni tashkil etish chora-tadbirlari to'g'risida"gi PQ-614-sonli qarori, 2012 yil 21 martda "Zamonaviy axborot-kommunikatsiya texnologiyalarini yanada joriy etish va rivojlantirish chora-tadbirlari to'g'risida"gi PQ-1730-sonli qarori, 2013 yil 20 sentyabrda "Mamlakatimizning dasturiy ta'minot vositalari ishlab chiquvchilarini rag'batlantirishni yanada kuchaytirish chora-tadbirlari to'g'risida"gi PQ-2042-sonli qarori, 2013 yil 25 sentyabrda "Milliy geografik axborot tizimini tashkil etish investitsiya loyihasini amalga oshirish chora-tadbirlari to'g'risida"gi PQ-2045-sonli qarorilari bu boradagi ishlarni yanada rivojlantirish imkonini berdi.

O'zbekiston Respublikasi birinchi Prezidentining "Malakali pedagog kadrlar tayyorlash hamda o'rta maxsus, kasb-xunar ta'limi muassasalarini malakali kadrlar bilan ta'minlash tizimini yanada takomillashtirishga oid chora-tadbirlar to'g'risida" gi qarorida axborot kommunikatsiya texnologiyalari, elektron ta'lim resurslari va multimediya taqdimotlaridan foydalanish borasidagi chet el tajribasini chuqur xar tomonlama o'rganib chiqish hamda ta'lim jarayonida qo'llash xaqida aloxida ta'kidlab o'tilgan. Zamonaviy axborot va kommunikatsiya texnologiyalari o'qitishning yangi shakl va metodlarini ishlab chiqish uchun keng imkoniyat yaratildi.

Kompyuter, dasturiy mahsulotlarsiz foydalanuvchi uchun hech qanday qiymatga ega bo'lmaydi. Foydalanuvchi uchun aynan dasturiy mahsulotlar shunday afzalliklar va qulayliklar beradiki, ular tufayli biz o'z ishimizni tezroq, aniqroq, chuqurroq va nihoyat, samaraliroq bajaramiz. Shuning uchun dasturiy mahsulotlarni tanlash, kompyuterdan foydalanuvchilar uchun, ayniqsa dasturiy mahsulotlar bozorida foydalanuvchilarga yuzlab dastur mahsulotlaridan birini tanlash imkoniyati taklif etilayotgan va u tanlangan dasturga ko'nikib qolgandan keyin kelajakda undan voz kechish qiyin bo'ladigan hozirgi paytda, juda muhimdir.

Amaliy dasturlar orasida tilshunoslik tarjimalarini amalga oshiruvchi dasturiy mahsulotlar alohida o'rin egallaydi. Kompyuterlar tomonidan bajariladigan tarjimalar tarixi AQSH ning Djordjtaun universitetida 1954 yilda o'tkazilgan tadqiqotga borib taqaladi. O'shanda jahonda ilk bor matn (to'g'ri, bir nechtagina jumladan iborat) rus tilidan ingliz tiliga EHM yordamida tarjima qilingan edi. Bugungi kunda turli mamlakatlarda tarjima bo'yicha kamida yuzlab tijoriy kompyuter tizimlari ma'lum va mashhur, tadqiqiy tajriba tizimlari esa juda ko'p. Davriy nashrlarda tez-tez kompyuter tarjimasi tizimlarini qo'llash masalalari ko'tarilmoqda, ularning tavsifnomalari berilmoqda, tilshunoslik tarjimalarini amalga oshiruvchi dasturiy mahsulotlar taqqoslanmoqda.

Bizga ma'lumki kompyuter dasturlarida matn tarjimasida muammolarga duch kelamiz, yani matn tarjimasida matnning ma'no mazmuni buziladi. Buni bartaraf etish muhim ahamiyatga ega.

Bizga ma'lumki dasturlash tillarini o'qitish bir muncha qiyinchiliklardan iboratdir. Bu qiyinchiliklarni bartaraf etish uchun biz interfaol metodlardan foydalanishimizga to'g'ri keladi. Zamonaviy pedagogik texnologiya o'zining pedagogika va boshqa fan yutuqlari bilan bog'liq xususiy nazariyasiga ega: u birinchi galda o'quv-tarbiyaviy jarayonni ilmiy asosda qurishga yo'naltirilgan, o'qitishning axborotli vositalaridan, didaktik materiallardan, faol metodlardan keng foydalanishga asoslangan o'qituvchi va talabalarning birgalikdagi faoliyatiga zamin yaratadi.

Hozirgi yetishib kelayotgan yosh avlod uchun dasturlash tillarini o'rganish va uni amalda tadbiiq etish dolzarb vazifalardan biridir. Albatta bu dolzarb vazifalarni bajarishda dasturlash tillarini o'rgatuvchi va o'z bilimini tekshiruvchi tizimlarning o'rni judayam muhim. Dasturash tillarini o'rganishda judayam ko'plab o'quv qo'llanmalar, kitoblar, monografiyalar va uslubiy ko'rsatmalar ishlab chiqilgan. Bularni o'zlashtirishda o'quvchidan sabr, iroda va matonat talab etiladi. O'quvchiga bularni tezda o'rganishda zamonaviy pedagogik texnologiyalarni qo'llash zamon talablaridan biridir. Dars samaradorligini oshirishda o'qituvchi avvalo maqsad

belgilab olishi, noan'anaviy va an'anaviy darslarning farqini ko'ra olishi shart. Noan'anaviy darslardan maqsad talabalarning o'zlashtirish ko'rsatkichini ko'tarish uchun, fanni o'qitishda yanada mazmunli, qiziqarli va tushunarli o'tkazishda o'qituvchi va talaba faoliyatiga yangilik kiritib, interfaol metodlardan foydalanish hamda talabalarni tez va samarali baholashga erishishdir. Shu bilan birga noan'anaviy darslarda o'quv jarayonida talabalar mustaqil fikrlay olishga, ijodiy yondashishga, izlanishga, tahlil eta olishga, o'zlari xulosa qilib, o'z-o'zini hamda guruhni va guruh talabani baholay olishga o'rgatadi. O'qituvchi esa ularning bu faoliyatiga imkoniyat va sharoit yaratib bera olishi kerak. Bu esa o'qitish jarayonining asosi hisoblanadi.

Dasturlashni o'qitish uchun ta'lim oluvchilarni birinchi navbatda guruhlariga ajratib olamiz:

1-guruh. Boshlovchilar (maktabning 7-9 sinf o'quvchilari). Bu bosqichdagi o'quvchilarning yoshini hisobga olgan holda quyidagi o'qitish usullarni qo'llashimiz mumkin: Puzzle (jumboq) – usuli. Bu usulda o'qituvchi to'g'ri tuzilgan dasturni oladi hamda uni satrlarini o'rnini almashtirib yuboradi va o'quvchilarga taqdim etadi. O'z navbatida o'quvchilar dasturning satrlarini to'g'ri dastur ko'rinishiga olib kelishlari kerak bo'ladi; Hand tracing (qadamlarni tekshirish) – usuli. Bu usulda o'qituvchi biror to'g'ri tuzilgan dasturni olib uni qadamma-qadam bajarilish jarayonini hamda o'zgaruvchilarning qiymatlarini o'zgarishini jadval ko'rinishida ko'rsatib beradi.

Bu usulning yutuqli tarafi o'quvchilar dasturni tushunishlari oson va unga qiziqishlari ortadi.

2-guruh. O'rta guruh (Akademik lisey va kasb-hunar kollejlari o'quvchilari). Ushbu guruhda o'quvchilar dasturlashdan xabardor bo'ladi. Bu guruh yoshidagilar uchun quyidagi usullarni qo'llash maqsadga muvofiq: Teamwork (jamoaviy ishlash) – usuli. Ushbu usulda o'quvchilar kichik guruhlariga bo'linadi hamda har bir guruhga sardor tayinlanadi. O'qituvchi tomonidan kichik loyiha yoki masala o'rta tashlanadi va har bir jomoa o'z usulida yondashib natijaga erishishi lozim bo'ladi. Work in pairs (juft bo'lib ishlash) – usuli. Bu usulda o'quvchilar juft-juft bo'lib bo'linishadi va o'qituvchi tomonidan beriladigan topshiriqni birgalikda bajarishadi. Yuqoridagi usullarning ikkisi ham o'quvchilarni jamoaviy ishlash va o'z fikrini jamoadoshlariga yetkaza olish qobiliyatini shakllantiradi. Undan tashqari dasturlashga bo'lgan qiziqishlari yanada ortadi.

3-guruh. Ilg'or guruh (Oliy ta'lim muassasalari talabalari). Ushbu guruh juda muhim xisoblanib bunda talabalarni mustaqil ishlash qobiliyatini shakllantirish lozim bo'ladi. Bu guruh ta'lim oluvchilariga qisman yuqorida ta'kidlab o'tilgan

usullarni ham qo'llash mumkin. Ammo "Automata approach" (Avtomatlashtirilgan yondashuv) usuli ko'proq samarali bo'ladi. Bu usulda talabalar berilgan topshiriqlarni avtomatlashtirilgan tizimlar yordamida tekshiradi va eng to'g'ri yechimga erishish darajasiga yetadi. Bunday tizimga <http://codeforces.com/> saytini misol qilib keltirishimiz mumkin.

Dasturlash yoki kod yozish deb odatda kompyuterga ba'zi operatsiyalarni qanday bajarishni buyurish tushuniladi. Dasturlash o'z ichiga ko'plab matematik hisob-kitob va strukturani olsada, aslida u yuqori darajada ijodiy ish hisoblanadi: bir qator dastur kodi bilan bir vaqtda mahsulotning tayyor holati haqida o'ylaysiz. Dasturlashni o'qitishda ko'p mamlakatlar dasturiy yechim to'g'riligini avtomatik testlovchi tizimlardan foydalanilgan holda o'tishadi. Hozirgi dasturlash bo'yicha jahon olimpiadalarida Rossiya mamlakatidan qatnashuvchilar doimo yuqori natijalarni ko'rsatib kelishmoqda. Ular bunday natijalarga qanday erishishadi? Quyida biz unga javob olishimiz mumkin. Rossiyada dasturlash tillari maktab davridan boshlanadi. Ular maktab ta'limida 7-sinfdan boshlab dasturlash tillarini o'qitishni boshlashadi. Ularning ta'lim tizimi barcha mamlakatlarning ta'lim tizimi bilan bir xil bo'lishiga qaramasdan ularning yutuq jihatlari nimada ekan deb o'ylashimiz mumkin. Ularning asosiy yutuqlaridan biri shundaki ular dasturlash tillarini o'qitishni boshlashgandan so'ng ma'lum bir vaqt oralig'i(masalan, har 1 oy)da online kontestlar o'tkazib turishadi. Bu online kontestlar ikki ko'rinishda yakka tartibda yoki komandaviy ko'rinishda bo'ladi. Har bir kontest tugatilgandan so'ng natijalar tahlil qilinadi va har bir masala muhokama qilinadi. Bu usul ularning eng yutuqli jihatlari hisoblanadi. Ular o'tkazadigan online kontestlar komandaviy holatda ham ko'p bo'lib turishi ularning dasturlash bo'yicha xalqaro komandaviy olimpiadalarda yuqori natija olishlariga tayorgarlik bo'ladi. Rossiyaning dasturlashni o'qitish bo'yicha yana bir yutuqli jihati shundaki ular bir yil davomida 2 marotaba Yozgi va Qishki lagerlar tashkil qilinishidir. Bu lagerda ma'lum kun tartibi bo'yicha ish ko'riladi. Har kuni ertalab kontest o'tkaziladi va kontest yakunlangandan so'ng bu masalalar muhokama qilinadi. Masalani muhokama qilishda barcha qatnashadi, yani bir kishi biror masalani muhokama qiladi. Agarda bu masalaga boshqa kontestchida boshqacha qulayroq va samaraliroq usul bo'lsa u ham chiqib o'z fikrini bildirishi mumkin. Bu usul orqali barcha lager qatnashchilari bir masalani yechish uchun bir nechta usullarni bilib olishadi. Lager davomida lagerchilar faqatgina tayorgarlik emas balki ular uchun qo'shimcha mashg'ulotlar, sport bilan ham shug'ullanishadi. Rossiya dasturlash bo'yicha xalqaro olimpiadalarda qatnashishi bo'yicha yuqori o'rinlarda turadi. Ular asosan ACM ICPC olimpiadasida yuqori natijalarni ko'rsatib kelishmoqda. Association for Computing Machinery (ACM) bu Hisoblash mashinalar uyushmasidir. International Collegiate Programming Contest (ICPC) bu dunyodagi universitet talabalari

o'rtasidagi dasturlash bo'yicha xalqaro olimpiada hisoblanadi. Xitoy davlati bizga ma'lumki rivojlangan davlatlardan biri hisoblanadi. Bu davlatda ham dasturlashga juda katta e'tibor beriladi. International Olympiad in Informatics (IOI) bu dasturlash bo'yicha maktab o'quvchilari o'rtasidagi xalqaro olimpiada bo'lib, bu olimpiadada Xitoy medallar soni bo'yicha birinchi o'rinda turadi(1-rasm). Endi dasturlashni yurtimizda qanday o'tilishiga to'xtalib o'tadigan bo'lsak. O'quvchilarga dasturlash tillari 9-sinfdan boshlab o'tila boshlanadi. Respublikamizning har bir viloyatlarida har xil dasturlash tillari o'tiladi. Masalan, Basic, Paskal, Delphi – dasturiy vosita, C/C ++, Java va boshqa dasturlash tillari o'tiladi. Ular har xil dasturlash tillarida o'z darslarini olib borishadi. Lekin ularning dasturlash tillarida olib boradigan dasturlari deyarli bir xil. Ya'ni bir-biriga yaqin saviyadagi masalalarni ishlashadi. Maktab o'quvchilari o'rtasi(9-sinf)da Informatika (dasturlash) fani bo'yicha Tuman, viloyat fan olimpiadalari o'tkazilib kelinmoqda. Olimpiadadan faxrli o'rinni egallagan o'quvchilar akademik litsey va kasb hunar kollejlari imtiyozli davlat test markazi imtixonini topshirmagan holatda o'qish imkoniyatini qo'lga kiritadilar. Akademik litsey va kasb hunar kollejlari ham dasturlash fanlari o'tilib ular o'rtasida ham tuman, viloyat va Respublika fan olimpiadalari o'tkazilib kelinadi. Undan tashqari Akademik litsey va kasb hunar kollejlari o'quvchilari o'rtasida Informatika va dasturlash bo'yicha Butun Rossiya maktab komandalari o'rtasidagi olimpiadada qatnashishlari mumkin bo'ladi. Bu olimpiadada O'zbekiston yoshlari 2006-yildan beri qatnashib kelishmoqda. Bu olimpiadada O'zbekiston yoshlarining natijalari unchalik qoniqarli bo'lmagan bo'lsada yildan yilga ularning natijalari yuqorilab bormoqda. 2014-yilgi olimpiadada esa O'zbekistonning Buxoro davlat universiteti qoshidagi 3-sonli Qorako'l akademik litseyi 3 nafardan iborat komandasi Sharipov Abdulla, Jabborov Ahror, Atoyev Abubakir 3-darajali diplom bilan taqdirlanishgan.

	Country	IOI Host	Medals v			
			G	S	B	Total
	China	2000	75	24	12	111
	Russia	2016	55	33	12	100
	United States of America	2003	45	34	15	94
	Poland	2005	36	37	28	101
?	Republic of Korea	2002	35	35	26	96
?	Romania		28	45	26	99
?	Bulgaria	1989, 2009	24	41	31	96
	Slovakia		24	37	27	88
	Iran		21	49	22	92
	Taiwan	2014	18	43	24	85
	Japan		18	19	8	45
?	Thailand	2011	15	33	44	92
?	Belarus		14	34	35	83
?	Germany	1992	14	26	37	77

1-rasm.

Davlatlarning IOI da olgan medallar soni IOI olimpiadasida O'zbekiston yoshlari 2015-yildan boshlab qatnashib kelishmoqda. 2017-yil yanvar oyida Qozog'istonda o'tkazilgan dasturlash bo'yicha olimpiadaga O'zbekistondan 2 nafar o'quvchi borib qatnashib kelishdi. Bular, Toshkent axborot texnologiyalari

universiteti qoshidagi 2-akademik litsey 2-kurs o‘quvchisi Xo‘jayev Temur va Namangan davlat universiteti qoshidagi 2-akademik litsey 3-kurs o‘quvchisi Hasanov Husaynlardir. Bu olimpiadada Xo‘jayev Temur faxrli qatnashib “Bronza” medalini qo‘lga kiritib qaytdi. Bu maktab o‘quvchilari o‘rtasida dasturlash bo‘yicha xalqaro olimpiadadagi birinchi yutug‘imiz bo‘ldi. codeforces.com saytida online musobaqalar bo‘lib turadi. Bu tizim orqali har bir foydalanuvchiga reyting beriladi. Bu reytinglar kesimida mamlakatlar reytingini ham ko‘rishimiz mumkin. Hozirda O‘zbekiston bu reytingda 43-o‘rinni egallab turibdi.

Rank	Name	Member count	Rating
1	Russia	1693 (5222)	3109 (3117)
2	China	1778 (6108)	3027 (3042)
3	Japan	289 (866)	2806 (2842)
4	Korea, Republic of	262 (489)	2800 (2819)
5	Belarus	274 (806)	2787 (2813)
6	Ukraine	378 (1245)	2755 (2779)
7	Poland	170 (407)	2751 (2770)
8	United States (USA)	298 (878)	2596 (2746)
9	Taiwan	140 (375)	2579 (2672)
10	Romania	243 (594)	2512 (2550)
11	Croatia	81 (156)	2499 (2535)
12	Iran	395 (1345)	2470 (2531)
13	Switzerland	17 (48)	2434 (2522)
13	Australia	35 (73)	2434 (2445)
15	Canada	72 (182)	2398 (2422)
16	Korea, DPR	18 (45)	2382 (2404)
17	Vietnam	450 (1209)	2381 (2401)
18	Brazil	286 (610)	2364 (2394)

2-rasm.

codeforces.com saytida mamlakatlar reytingi Agarda biz dasturlashni o‘qitishda jahon tajribasidan foydalanishni yo‘lga qo‘ysak bizda ham rivojlanish sezilarli darajada o‘zgaradi.

Ilmiy tahmin. Elektron hujjatlar bilan ishlovchi yuridik va jismoniy shaxslarda Microsoft Office dasturlar paketidan foydalanishda word dasturida yaratilgan hujjatlarni “kitobcha” shaklida chop etish munosabatlari orqali yuzaga keluvchi muommolarni qisqa vaqt oralig‘iga bartaraf qilinadi.

Bitiruv malakaviy ishning maqsadi.

– ta’lim bo‘yicha nazariy va amaliy bilimlarni mustahkamlash va kengaytirish, olingan bilimlarni muayyan ilmiy, texnikaviy, ishlab chiqarish, iqtisodiy, ijtimoiy, madaniy vazifalarni hal etishda ijodiy ishlash, hal etilayotgan masalaning (muammoning) qo‘yilish jarayonidan boshlab, uni nihoyasiga etkazish bo‘yicha qaror qabul qilishda bo‘lgan mas’uliyatni his etishga o‘rgatish;

– zamonaviy ishlab chiqarish, iqtisodiyot, texnika va madaniyatning rivojlanishi sharoitida talabalarni mustaqil ishlashga tayyorgarligini ta'minlashdan iborat.

- Microsoft Office dasturlar paketidan doimiy foydalanuvchilar uchun, hozirgi kunda ko'p uchrayotgan muammolardan biri bu Word dasturida tayyorlagan hujjatlarni "kitobcha" holatida chopetish jarayoni bo'lib, Word va boshqa amaliy dasturlarda tayyorlangan elektron hujjatlarni o'qish va o'qitish jarayoni uchun mos holda chopetishdan iborat.

Bitiruv malakaviy ish vazifalari. Microsoft Office dasturlaridan foydalanishda chopetish jarayonini soddalashtirish.

Bitiruv malakaviy ish ob'ekti. Tadqiqot ob'ekti etib – yuridik va jismoniy shaxslar tomonidan elektron hujjatlar bilan ishlash jarayoni olingan. Chunki mehnat samaradorligini oshirishda vaqtni tejash jiddiy ahamiyatga ega.

Bitiruv malakaviy ish predmeti. Tadqiqot ishimizda BMI predmeti bo'lib, C++ Builder 6 dasturida foydalanuvchilar uchun amaliy dastur ishlab chiqish, metodlari va vositalari hisoblanadi.

Tadqiqot metodlari. Nazariy (pedagogik-psixologik va o'quv-metodik adabiyotlar (Oliy oquv yurtlari dasturlash asoslari darsliklari), o'quv-metodik qo'llanmalar, me'yoriy hujjatlar (yillik mavzuiy-taqvimiy rejalari va uzviylashtirilgan davlat ta'lim standarti va o'quv dasturi, ixtisoslashtirilgan davlat umumta'lim maktab-internat hamda ayrim fanlar chuqurlashtirilib o'qitiladigan yo'nalishlar uchun dasturlash fanidan uzviylashtirilgan o'quv reja va dasturlar), ta'lim jarayoni uchun mo'ljallangan pedagogik dasturiy vositalarni tahlil qilish. Amaliy (talabalar va o'qituvchilar bilan suhbatlar o'tkazish, talabalarinina amaliy dasturlar yaratish jarayonida ilg'or pedagogik tajribalarni o'rganish va umumlashtirish, natijalarni tahlil qilib pedagogik dasturiy vositalardan o'quv jarayonida foydalanish).

Tadqiqotning metodologik asosini «Ta'lim to'g'risida»gi Qonun, «Kadrlar tayyorlash Milliy dasturi», O'zbekiston Respublikasi ta'lim tizimini axborotlashtirishni rivojlantirish bo'yicha qabul qilingan direktiv va me'yoriy hujjatlar, pedagogika, informatika va axborot texnologiyalari sohasidagi taniqli olimlarning ilmiy ishlari tashkil etadi.

Ilmiy yangiligi. Dasturiy vositalar ishlab chiqarish tizimini zamon talabi darajasida tashkillashtirish dasturchidan ko'proq mehnat talab qilib, ishlab chiqarishni sifatli bajarishni taqozo etadi. BMI da:

- talabalarning bilimlarini mustahkamlash, o'zlashtirish jarayonini faollashtirishga xizmat qiluvchi, kursga oid mavzularni o'rgatishning samarali yo'llari ishlab chiqildi;

- amaliy dasturlar ishlab chiqarish jarayonini pedagogik dasturiy vositalar asosida tashkil etish, ta'lim tizimining zamonaviy talablariga mosligi ilmiy nuqtai nazardan asoslab berildi;

- alaliy dasturlar ishlab chiqarish jarayoni bo'yicha pedagogik dasturiy vosita ishlab chiqildi va ularning ishlab chiqarish jarayonidagi ahamiyati ilmiy asoslab berildi;

- pedagogik dasturiy vositalarni qo'llash orqali, malum bo'lgan muommalarni amaliy jihatdan bartaraf etuvchi dasturlar, mehnat jarayonini intensivlashtiruvchi va optimallashtiruvchi imkoniyatlari ko'rsatildi.

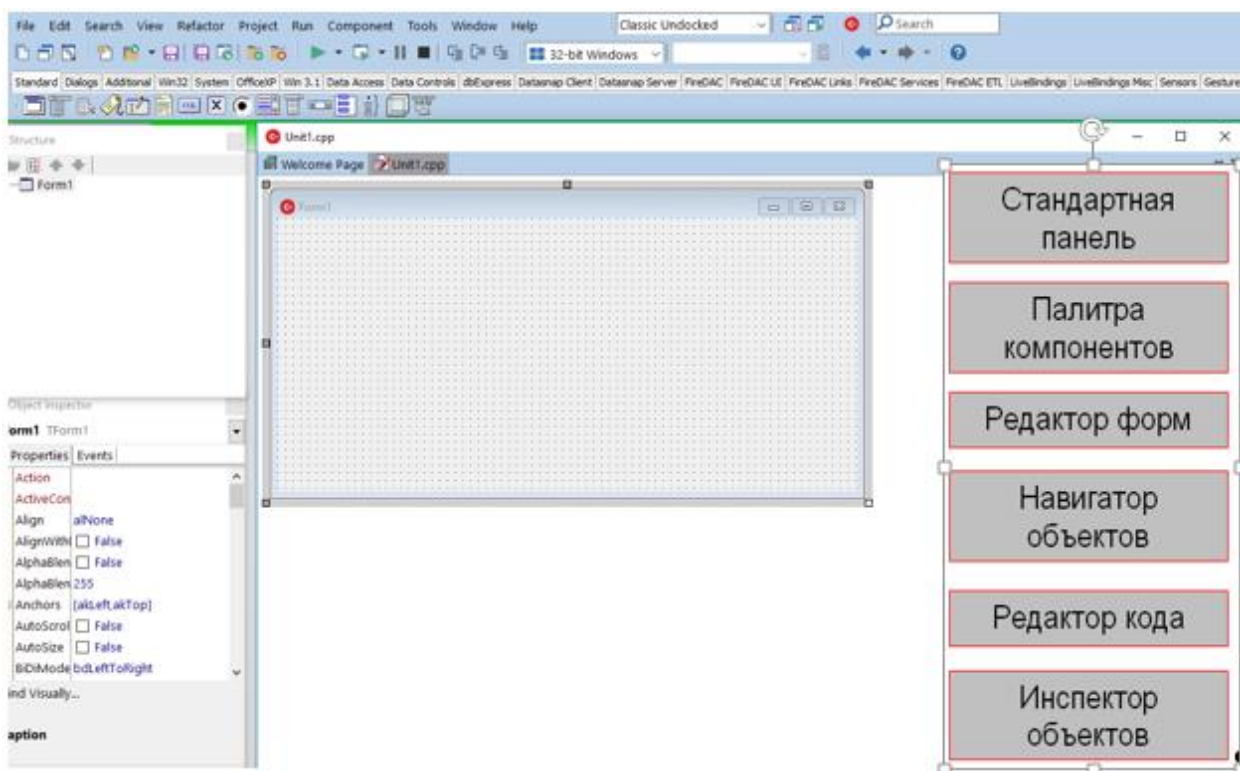
Bitiruv malakaviy ish natijalarining ilmiy va amaliy ahamiyati. Microsoft Office dasturlar paketidan doimiy foydalanuvchilar uchun, hozirgi kunda ko'p uchrayotgan muammolardan biri bu chopetish bo'lib, Word va boshqa amaliy dasturlarda tayyorlangan elektron hujjatlarni foydalanishga qulay, o'qitish va o'qish jarayonida foydalanishni tashkil qilish tayorlashni talab qilinadigan vaziyatlardir. Quyidagi muommoni hal qilish uchun ko'pgina amaliy dasturlar va Microsoft Office amaliy dasturida VBA tilida yozilgan makrosalar mavjud bo'lib ularda dasturlarning Word dasturidan ajralgan holda ishlatilishi borasida kamchiliklar kuzatiladi. Bitiruv malakaviy ishda tavsiya qilingan Borlan C++ Builder 6 dasturlash muhitida yaratilgan dastur bitiruv malakaviy ishda yoritilgan muammoli holatlardan holidir.

Ishning tuzilishi. Bitiruv malakaviy ish kirish qismi, asosiy qism (I-bob, II-bob,), xulosa va takliflar, foydalanilgan adabiyotlar ro'yxati va internet saytlardan iborat.

I bob. Chop etish jarayonini boshqaruvchi dasturiy ilova yaratishning nazariy yoritilishi.

Word dasturida tayyorlangan hujjatlarni chop etish turlari qo'yiladigan talablarga mos ravishda mavjud. Hujjatlar chopetishda foydalaniladigan qog'ozlarning A4, A5 va boshqa o'lchamlarning mavjudligi, tayyorlangan hujjatlarning hizmat qilish sohalarining xilma xilligidan kelib chiqqan holda chopetish jarayonini ham huddi shunday xilma xil tashkil etish zarur. Men talabaman va har bir talaba kabi izlanuvchanlik qiziquvchanlik menga hos. Har doim yangi bilimlarni o'rganishga kirishganimda menga darsliklar adabiyotlar kerak bo'ladi, elektron kitoblardan foydalanishni o'zimga noqulay deb hisoblaganim uchun men qog'oz holatidagi adabiyotlar va qo'llanmalardan ko'proq foydalanaman. Lekin bu borada ham muommolarga tez tez duch kelib turaman. Foydali adabiotlarning ko'pligi, chopetish uchun ishlatiladigan qog'oz tan narxi qimmatliligi meni uni tejashga undadi va har bir offis jihozlaridan foydalanuvchilar kabi hujjatlarimni "kitobcha" holatida chop etish jarayoni bilan to'qnashdim. Muommoning eychimi sifatida mavjud dasturlardan foydalanish mumkin, ammo ulardagi kamchiliklar chopetish qurulumalaridagi xilma xillik va birinchi marta foydalanuvchi uchun tushunarsiz bo'lgan "4 ga karrali son kiriting" yoki "5 ta list qo'ying" kabi jumlar noqulaylik tug'diradi. Shuning uchun ham yuqorida sanab o'tilgan kamchiliklardan holi bo'lgan dasturiy ilova yaratishga qaror qildim. Bunig uchun dastlab C++ dasturi bilan tanishib uning imkoniyatlarini o'rganishimiz kerak.

C++ BUILDER bilan dastlabki tanishuv. Ishlab chiqishning integratsiyalashgan muhiti Komponentalar palitrasini birlashtiradi. Shakllar Muharriri, Kod Muharriri, Ob'ektlar Noziri, Ob'ektlar Xazinasini - bular hammasi kod va zahiralar ustidan to'liq nazoratni ta'minlovchi dasturiy ilovalarni tez ishlab chiqish instrumetlari(3-rasm).



3-rasm. Ishlab chiqish muhitining tuzilishi

■ **Komponentlar Palitrasi** ilovalarni qurishda taklif qilinadigan 100 dan ortiq takroran qoʻllanadigan komponentlardan iborat.

■ **Shakllar Muharriri** dasturning foydalanuvchi bilan interfeysini yaratish uchun moʻljallangan.

■ **Kod Muharriri** dastur matnini, xususan, voqealarga ishlov berish funksiyalarini yozish uchun moʻljallangan.

■ **Ob'ektlar Noziri** qotib qolgan chigal dasturlash zaruratisiz ob'ektlar xususiyatlarini vizual oʻrnatish imkonini beradi hamda shunday voqealarni oʻz ichiga oladiki, bu voqealarni ularning paydo boʻlishiga nisbatan ob'ektlar reaksiyasi kodlari bilan bogʻlash mumkin boʻladi.

■ **Ob'ektlar Xazinasini** maʼlumotlarning shakl va modullari kabi ob'ektlarga ega boʻlib, ular ishlab chiqishda muvaqqat sarflarni kamaytirish maqsadida koʻplab ilovalar bilan boʻlinadi.

C++Builder ilovalarni qurishning vizual metodikasini Komponentlar Palitrasidan kerakli boshqarish elementlarini tanlab olish vositasida joriy etadi. Har bir komponenta (masalan, tugmacha) bilan ushbu komponenta turini va xulq-atvorini oʻzgartiradigan xususiyatlar bogʻliq boʻladi. Har qanday komponenta ushbu komponentaning turli xildagi taʼsirlarga reaksiyasini (munosabatini) aniqlab beradigan voqealar seriyasini keltirib chiqarishi mumkin. Bunday keyin => belgilari siz C++Builder muhitida amalga oshiradigan xatti-harakatlarni bildiradi.

=>C++Builder ni chaqiring va bosh menyudagi File | New Arrlication komandasi bo'yicha yangi ilovalar ustida ishlashni boshlang.

=>sichqonchani Komponentalar Palitrasining qo'shimcha ilovalari ustida bosib, foydalanuvchi ish ko'radigan dastur interfeysi elementlarining mavjud assortimentini ko'rib chiqing.

Palitraning bir qo'shimcha ilovasidan ikkinchisiga o'tib, kirish mumkin bo'lgan komponentlar to'plami o'zgarayotganining guvohi bo'lishimiz mumkin. Sichqoncha kursori komponentlar belgisi ustida to'xtaganda, aytib turish nomi paydo bo'ladi. Agar F1 klavishasini bossak, tizimning ma'lumotnomalar xizmati tanlab olingan komponenta haqida to'liq ma'lumot chiqarib beradi.

Vizual loyihalas: Bizning birinchi ilovamiz bolalarning «O'nta negr bolasi» sanoq she'rini generatsiya qiladi. Dastlabki versiyada faqat uchta ob'ekt kerak bo'ladi: ro'yxat, tahrir qilish maydoni va tugmacha. Komponentalarni loyihalash shakliga olib o'tamiz hamda ilovani asta-sekin rivojlantira boshlaymiz. Tashib olib o'tish metodi (drag-and-drop) quyidagilardan iborat: sichqoncha tugmachasini tanlab olingan komponenta ustida bosib, kursorni shaklning to'g'ri kelgan eriga o'tkazib, keyin esa sichqoncha tugmachasini yana bosib. Boshida faqat «standart» Palitra Komponentlari bilan cheklanamiz:

=> Standard qo'shimcha ilovani tanlab oling.

=> Ro'yxat komponentasini ListBox shakliga olib o'ting.

=> Tahrir qilinatgan kiritish maydoni EditBox ni olib o'ting.

=> Button tugmachasi komponentasini olib o'ting.

=>Komponentalarni o'zingizning ilovangizdagi darchada qanday ko'rmoqchi bo'lsangiz, shunday joylashtiring va o'lchamlarini shunday o'zgartiring.

Ob'ekt Noziri yordamida komponentalar xususiyatlarining boshlang'ich qiymatlarini aniqlang. Items ro'yxatining xususiyatlar qiymatlari katagida tugmachani bosib, ochilgan muharrir darchasida she'rning dastlabki 7 satrini kiriting. SHakl va tugmachaning Caption xususiyatida ularning ma'noli nomlarini ko'rsating (mos ravishda, «O'nta negr bolasi» va «Natija»). Tahri qilish maxdonining Text xususiyatida natijani aytib berish satrini kiriting («To'qqizta negr bolasi»).

Endi Kod Muharririga ulanish hamda, avval qabul qilinganidek, C++tilidagi har qanday dasturni yozish mumkin, shu jumladan, ANSI/ISO standartining so'nggi kengaytmalarini ham. Biroq, avval ilovalarni tez ishlab chiqishning yangi vositalari

hamda C++Builder da mavjud bo'lgan qo'shimcha komponentalar atributlaridan foydalanishga harakat qilib ko'ramiz.

Ikki yo'nalishli ishlanma texnologiyasi

Loyihaviy shablonlarni qo'llash

Xususiyatlar, metodlar va voqealar:

Ilovalarning tez ishlab chiqilishi ob'ektli mo'ljallangan dasturlash doirasida xususiyatlar, metodlar va voqealarning qo'llab-quvvatlanishini bildiradi. *Xususiyatlar* komponentalarning nomlar, matniy aytib berishlar yoki ma'lumotlar manbalari kabi turli xildagi tavsiflarini osongina o'rnatish imkonini beradi. *Metodlar* (a'zo-funksiyalar) komponentadagi ob'ekt ustida ma'lum operatsiyalarni amalga oshiradi. Bunday operatsiyalar jumlasida qayta tiklash yoki multimedia qurilmasini qayta o'rash kabi murakkab operatsiyalarni ham ko'rsatish mumkin. *Voqealar* komponentalarga foydalanuvchi ko'rsatayotgan faollashtirish (aktivizatsiya), tugmalarni bosish yoki tahrir qilinadigan kiritish kabi ta'sirlarni ushbu ta'sirlarga sizning munosabat kodlaringiz bilan bog'laydi. Bundan tashqari voqealar komponentalar holatlarida sodir bo'ladigan ayrim o'ziga xos o'zgarishlar paytida ham yuzaga kelishi mumkin. Bunday o'ziga xos o'zgarishlar qatorida ma'lumotlar bazasiga kirishning interfeysli elementlarida ma'lumotlarni yangilashni ko'rsatib o'tish kifoya. Xususiyatlar, metodlar va voqealar, birgalikda ish olib borar ekan, ular Windows uchun ishonchli ilovalarni intuitiv tarzda dasturlash muhiti - RAD ni hosil qiladi.

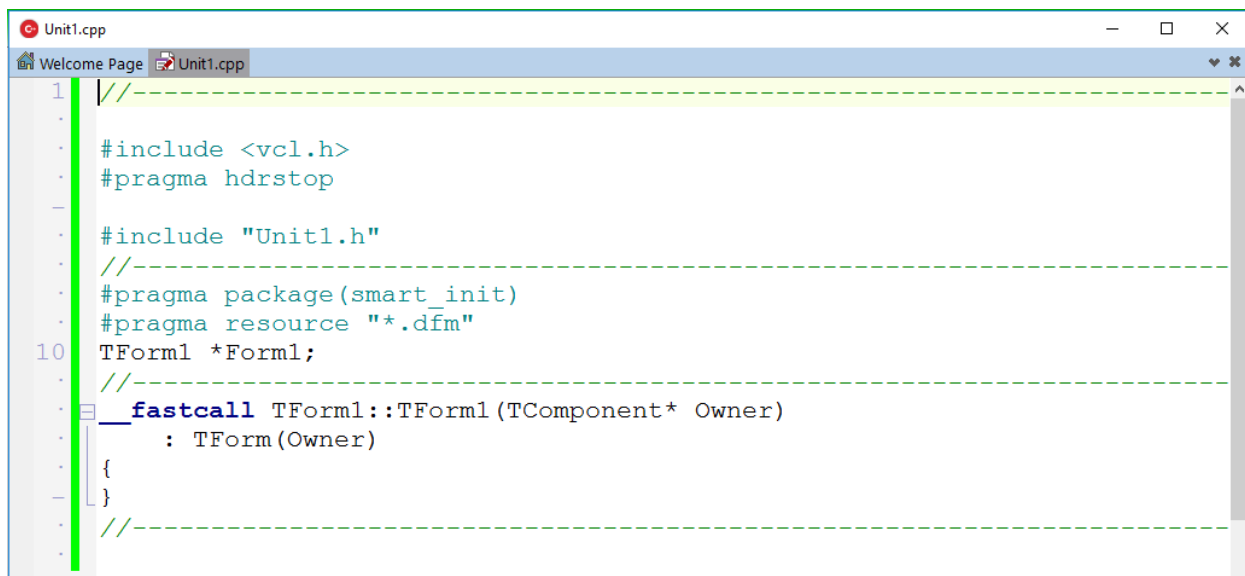
=> Tanlangan ob'ekt bilan assotsiatsiyalanadigan (birgalikda yodga olinadigan) voqealarni ko'rish uchun, Ob'ektlar Nozirida Voqealar (Events) qo'shimcha ilovasini ko'rsating.

=> O'zingiz shaklga joylashtirgan tugma komponentasini sichqoncha bilan ikki marta uring

=> Ochilgan Kod Muharriri darchasida kursor ButtonIClick funksiyasi tanasiga instruktsiyalarni kiritish uchun pozitsiyani ko'rsatadi. Bu funksiya esa tugmachani bosishda yuzaga keladigan OnClick voqeasiga ishlov berish uchun mo'ljallangan.

4-rasmda oddiy kod ko'rsatilgan bo'lib, u «Natija» tugmasini yana bir bor bosilishiga javoban avval turgan plev aytib berishini ro'yxat oxiriga, hamda navbatdagi next aytib berishini tahrir qilish maydoniga qo'shadi. ListBoxI->Items->Append(prev) yo'riqnomasi, Append metodi yordamida, rrev satrini ListBoxI ro'yxati ob'ektining Items xususiyatiga qo'shadi. EditI->Text=next yo'riqnomasi tahrir qilinayotgan EditI kiritish ob'ektining Text xususiyatiga next satrini taqdim etadi. Aytib berish satrlari ikki o'lchamli count massivida saqlanadi va static turdagi

butun o‘zgaruvchi tomonidan indekslanadi. Bu o‘zgaruvchi esa ButtonI tugmasini bosish bilan yuzaga keladigan voqeaga ishlov berish funksiyasining chaqirilishlari o‘rtasida o‘zining joriy qiymatini saqlaydi.



```
1 //-----
.
. #include <vcl.h>
. #pragma hdrstop
.
. #include "Unit1.h"
. //-----
. #pragma package(smart_init)
. #pragma resource "*.dfm"
10 TForm1 *Form1;
. //-----
. fastcall TForm1::TForm1(TComponent* Owner)
.     : TForm(Owner)
. {
. }
. //-----
```

4-rasm. Kod muharriri bajarilayotgan modul matnining Unit1.cpp faylida kiritilishi va tahrir qilinishini ta'minlaydi.

Birinchi versiyali ilovani loyihalash bosqichi shuning bilan tugallanadi va ishchi dasturni yaratishga kirishish mumkin bo‘ladi.

=>Run | Run bosh menyusini komandasi bilan ilovani kompilyatsiya qilish (ko‘chirish) va yig‘ish jarayonini ishga tushirib yuboring

=>Dastur chaqirilgach, bir necha marta «Natija» («Rezultat») tugmasini bosing.

Ikki yo‘nalishli ishlanma texnologiyasi:

C++Builder dasturchi va uning kodi o‘rtasida hech qanday to‘siqlarni qo‘ymaydi. Two-Way Tools ikki yo‘nalishli ishlanma texnologiyasi vizual loyihalash instrumentlari va Kod Muharriri o‘rtasida moslashuvchan, integrallashgan va sinxronlashtirilgan o‘zaro aloqa vositasida sizning kodingiz ustidan nazoratni taminlaydi. Ikki yo‘nalishli ishlanma instrumentlari qanday amal qilishini kuzatib borish uchun, quyidagi operatsiyalarni bajaring:

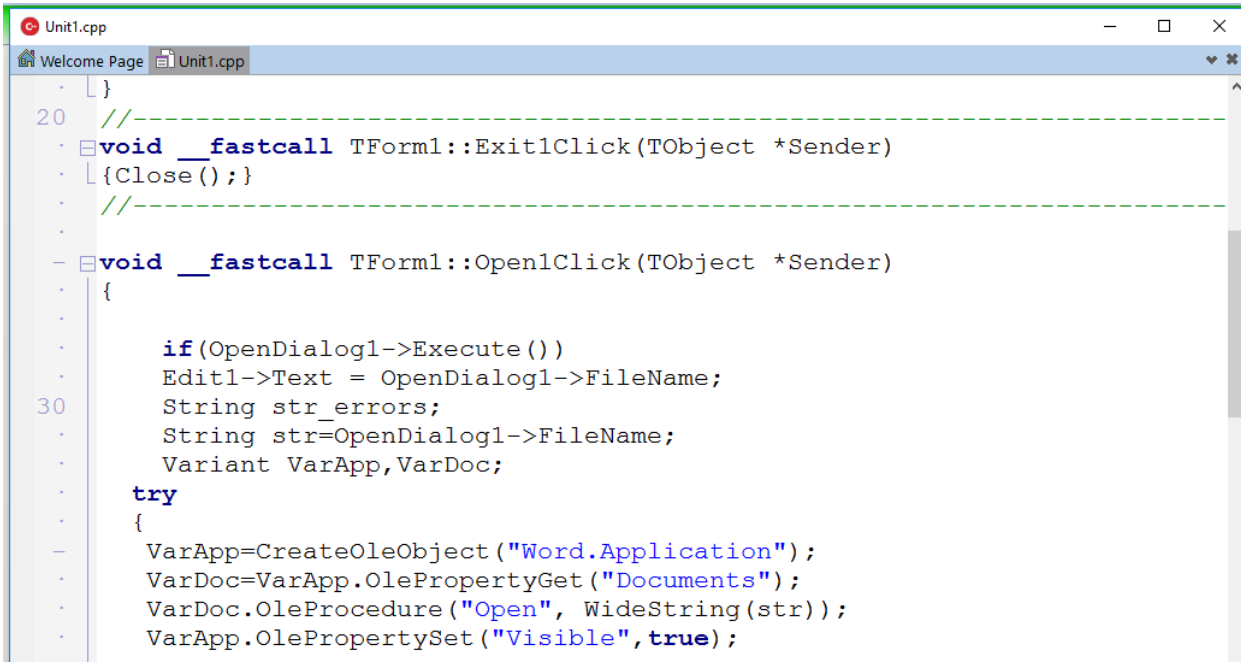
=>Sichqonchani o‘ng tugmasini bosib, Kod Muharririning kontekstli menyusini oching, keyin Swap Cpp/Hdr Files operatsiyasi yordamida Unit1.h. e’lonlar fayliga ulaning.

=> Instrumentlarning ekrandagi aksini shunday tashkil qilingki, bunda Kod Muharriri darchasida bir paytning o'zida loyihalanayotgan shakl va Unit.h fayli ko'rinsin.

=>OK Button tugmasining yana bitta komponentasini shaklga olib o'ting. Tugmaning Caption xususiyatida uning ma'noli nomi «Yangi band» deb ko'rsating.

Quyidagilarni kuzatib boring: siz tugmani shaklga olib o'tishingiz bilan, shu ondayoq Unit1.h faylida Button2 ob'ektining e'loni paydo bo'lishi kerak, OnClick voqeaning aniqlanishi esa ushbu vorqeaning qayta ishlovchisi bo'lgan Button2Click metodining e'lon qilinishini generatsiyalaydi. shaklni loyihalashning va kodni avtomatik generatsiyalash jarayonlarining mana shunday sinxronlashtirilishi C++ilovaning vizual ishlanmasini haqiqatan ham tezlashtiradi va shuning bilan birga dasturning dastlabki matni ustidan nazoratni to'la saqlab qoladi.

O'zimizning birinchi ilovamizni ishlashda yana bir qadam qo'yamiz - uni she'r bandini avtomatik tarzda generatsiyalashga majbur qilamiz. Buning uchun OnClick voqeasi ishlanmasining funksiyasini «Yangi band» tugmasini bosib, mazmun bilan to'ldirishga to'g'ri keladi.



```
Unit1.cpp
Welcome Page Unit1.cpp
. }
20 //-----
. void __fastcall TForm1::Exit1Click(TObject *Sender)
. {Close();}
. //-----
.
- void __fastcall TForm1::Open1Click(TObject *Sender)
. {
.
.     if(OpenDialog1->Execute())
.         Edit1->Text = OpenDialog1->FileName;
30     String str_errors;
.     String str=OpenDialog1->FileName;
.     Variant VarApp,VarDoc;
.
.     try
.     {
-         VarApp=CreateOleObject("Word.Application");
.         VarDoc=VarApp.OlePropertyGet("Documents");
.         VarDoc.OleProcedure("Open", WideString(str));
.         VarApp.OlePropertySet("Visible", true);
.     }
```

5-rasm. Unit1.cpp faylida voqeaning yangi qayta ishlatgichi

5-rasmda oddiy kod ko'rsatilgan bo'lib, u «Yangi band» tugmasining navbatdagi bosilishiga javoban yangi bandning ketma-ket yettita satrini chiqarib beradi, bunda birinchi va uchinchi satrlar prev o'zgaruvchisidan olinadi. Bu o'zgaruvchi qiymatini «Natija» tugmasi voqeasining qayta ishlatgichi o'zlashtirib olishi tufayli, bu qiymatni shakl sinfining foydalanuvchilar e'lonlarida public

seksiyasida qayta aniqlashga to'g'ri keldi. Bu ish ikkala tugma voqealarining qayta ishlatgichlariga bu qiymatga kirish uchun imkon yaratish maqsadida qilindi.

She'rni butunicha ko'rib chiqish imkonini yaratish maqsadida ro'yxat vertikal aylantirish chizig'iga ega bo'ldi.

C++Builder har bir ilova bilan yashirin nomlari quyidagicha bo'lgan uchta dastlabki faylni eslatishini yodda saqlab qolish kerak:

■ UnitI.cpp ilovangizning bajarilayotgan ishga tushirish kodini saqlaydi. Aynan shu erda siz foydalanuvchining komponentalar ob'ektlariga ta'siri paytidagi dastur reaksiyasiga javob beradigan voqealarning qayta ishlatgichlarini yozib qo'yasiz.

■ UnitI.h barcha ob'ektlar va ularning konstruktorlarining e'lonlariga ega. Voqealarni qayta ishlash funktsiyalari e'lonlaridagi _fastcall kalit-so'zga e'tibor bering (C++Builder bu funktsiyalarni avtomatik tarzda generatsiya qiladi). _fastcall tufayliparametrlar stek orqali emas, balki markaziy protsessor registrlari orqali uzatiladi. Voqealarni qayta ishlatgichlarning chaqirishlari tez-tez ro'y berib turadi, Shuning uchun stek xotirasidan parametrlarni tanlab olishga sraflanadigan vaqtning tejalishi ancha sezilarli natijalarni beradi. C++Builder komplyatsiya qiladigan va to'playdigan ilovalarning yuqori darajada tez harakatlanishining sabablaridan biri ham shu erda yashiringan.

■ ProjectI.cpp ilovada mujassamlangan barcha ob'ektlarga xizmat ko'rsatadi. Har qanday yangi shakl, dasturiy modul yoki ma'lumotlar moduli avtomatik tarzda loyihaviy faylga kiritiladi. Siz bosh menyu komandasi - View | Project Source yordamida yoki Loyiha Administratorining kontekstli menyusidan shu nomdagi opsiyani tanlab olib, Kod Muharriri darchasida loyihaviy fayl dastlabki matnining mazmunini ko'rib chiqishingiz mumkin. Hech qachon loyihaviy faylni qo'lda tahrir qilmang!

Balki siz, birinchi ilova ishlanmasini tugatib, dastlabki fayllarni keyingi seans uchun saqlab qolishni xohlarsiz. Buning uchun quyidagi xatti-harakatlardan birini bajarish kerak:

=>File | Save All komandasi ilovaning hamma dastlabki fayllarini saqlaydi.

File | Save komandasi dasturiy modulning ikkala komandasini saqlaydi, File | Save As komandasi esa ularga yangi nom berishga ruxsat etadi.

File | Save Project As komandasi, fayllarning joriy nomlaridan foydalanib, loyihaviy fayl tarkibiy qismlarining hammasidagi o'zgarishlarni saqlaydi.

Ikki yo'nalishli ishlanma texnologiyasi

Loyihaviy shablonlarni qo'llash:

Ob'ektlar Xazinasidagi tayyor loyihaviy shablonlardan foydalanar ekansiz, siz dasturni ishlab chiqishda ko'pchilik ilovalar uchun tipik bo'lgan operatsiyalarni chetlab o'tish imkoniyatiga ega bo'lasiz. Bu qanday operatsiyalar dersiz. Bular, masalan, menyu va tez chaqirib olish tugmalari panelini tuzish, standart chaqirishlar dialogi va fayllarni tuzishni tashkil etish bilan bog'liq operatsiyalardir. Siz shablonga kiritgan o'zgartirishlar xuddi shu loyihaviy shablondan boshqa ishlab chiquvchilarning foydalanishiga ta'sir qilmaydi.

Ko'p hujjatli interfeys (MDI) rejimida ishlash uchun loyihaviy shablon asosida ilova prototipini yaratish uchun quyidagi xatti-harakatlarni amalga oshiring:

Filtrlar muharriri darchasida TOpenDialog komponentasining Filter xususiyati qiymatlari ustunida matniy hujjatlar fayllarining nomlari va kengayishlarini ko'rsating.

Agar siz shunday ilovani kompilyatsiya qilib, to'play olsangiz, bu holda u faqat MDI rejimida darchalar bilan amallar bajara olishini hamda, darchani tanlab olingan fayllarning matniy mazmuni bilan to'ldirmay turib, fayllarni ochish dialogini chaqirib olishni «bilishi»ni ko'rishingiz mumkin. Ya'ni prototip nofunktsional va amaliy jihatdan befoyda bo'lib qoldi. Ilova qandaydir ongli xulq-atvorga ega bo'lishi uchun, quyidagi xatti-harakatlarni bajaring:

Bosh menyudan **View | Forms** komandasini bering va ro'yxatdan MDIChild nomli shaklni tanlab oling.

Memo tahrir qilish maydonining ko'p satrli komponentasini Palitraning **Standart** qo'shimcha ilovasidan sho''ba shaklga olib o'ting.

Lines xususiyatli satriy muharrirni tugmani bosish bilan chaqirib olib, TMemo komponentasining tahrir maydonini tozalang. Tahrir maydoni sho''ba darchasining hammasini egallashi uchun, Align xususiyatli alClient qiymatini o'rnatib. Uzun matniy fayllarni ko'rib chiqishni osonlashtirish maqsadida, ScrolBars xususiyatli Ssboth qiymatini o'rnatib.

=>Bosh shaklni sichqoncha yordamida faollashtirib, yana unga qayting hamda ilovalar menyusidan **File|Open** komandasini tanlab oling.

=>Kod Muharriri darchasida kursor menyuning tegishli elementini tanlashda yuzaga keladigan OnClick voqeasining qayta ishlatgichiga yo'riqnomani kiritish uchun kerakli pozitsiyani ko'rsatib beradi. C++Builder TOpenDialog bosh shakli (komponentalar Palitrasining Dialogs qistirmasidan) komponentasi uchun ushbu funktsiyaning e'lonini avtomatik tarzda generatsiya qiladi.

6 - rasmda shu voqeaning qayta ishlatgichi bo'lgan FileOpenItemClick funksiyasi tanasini tashkil qiluvchi zarur yo'riqnomalar ko'rsatilgan.

```
Unit1.cpp
Welcome Page Unit1.cpp
10 #pragma package(smart_init)
   · #pragma link "Word_XP_srvr"
   · #pragma resource "*.dfm"
   · TForm1 *Form1;
   ·
   ·
   · //-----
   · __fastcall TForm1::TForm1(TComponent* Owner)
   ·     : TForm(Owner)
   ·     {
   ·     }
20 //-----
   · void __fastcall TForm1::Exit1Click(TObject *Sender)
   ·     {Close();}
   · //-----
   ·
   · void __fastcall TForm1::Open1Click(TObject *Sender)
   ·     {
28     if(OpenDialog1->Execute())
       Edit1->Text = OpenDialog1->FileName;
```

6-rasm. *Main.cpp* faylida sho‘ba darcha yuklanishining amalga oshirilishi

Ajratib olingan yo‘riqnoma Child sho‘ba darchasi Memo1 ob’ektining Lines satrlarini OpenFileDialog->FileName nomli ochiq matniy faylning ichidagilari bilan yuklatadi.

Bu faylning ishlanishi hali tugallanganicha yo‘q, albatta. Siz uni kompilyatsiya qilib, to‘plab bo‘lsangiz, bir paytning o‘zida bir necha darchalardagi matniy fayllarni tahrir qila olasiz. Biroq natija beruvchi fayllarning saqlanishi hozircha ko‘zda tutilgan emas - o‘quvchining o‘zi File [Save va File | Save As menyulari komandalari uchun osongina kod yozib oladi.

Ilovani mantiqan eng sodda matniy muharrirga aylantirish uchun bu Edit nomli bosh menyu elementining tushib qoluvchi ro‘yxatiga qidirish va almashtirish komandalarini qo‘shish kerak.

1.1 Borland C++ Builder dasturlash muhitida chop etish jarayonini boshqaruvchi dastur yaratish uchun zarur algoritmlar.

Chopetish jarayonini boshqaruvchi dastur yaratish uchun dastlab if, for, massivlarda foydalanish, satrlar va fayllar bilan ishlashga imkon beruvchi kutubxonalar e'lon qilinishi hamda ularga doir funksiyalar haqida malumotlarga ega bo'lishimiz zarur. Shu o'rinda fayllar bilan ishlash jarayonida qo'llaniladigan funksiyalarni ham yoritib o'tamiz.

Operator tushunchasi: Programmash tili operatorlari yechilayotgan masala algoritmini amalga oshirish uchun ishlatiladi. Operatorlar chiziqli va boshqaruv operatorlariga bo'linadi. Aksariyat holatlarda operatorlar nuqtali vergul (;) belgisi bilan tugallanadi va u kompilyator tomonidan alohida operator deb qabul qilinadi (for operatorining qavs ichida turgan ifodalari bundan mustasno). Bunday operator ifoda operatori deyiladi. Qiymat berish amallari guruhi, xususan, qiymat berish operatorlari ifoda operatorlari hisoblanadi:

$i++; --j; k+=I;$

Programma tuzish amaliyotida bo'sh operator – ';' ishlatiladi. Garchi bu operator hech nima bajarmasa ham, hisoblash ifodalarini til qurilmalariga mos kelishini ta'minlaydi. Ayrim hollarda yuzaga kelgan «boshi berk» holatlardan chiqib ketish imkonini beradi.

O'zgaruvchilarni e'lon qilish ham operator hisoblanadi va ularga e'lon operatori deyiladi.

Shart operatorlari: Oldingi bobda misol tariqasida keltirilgan programmalarda amallar yozilish tartibida ketma-ket va faqat bir marta bajariladigan holatlar, ya'ni chiziqli algoritmlar keltirilgan. Amalda esa kamdan-kam masalalar shu tariqa yechilishi mumkin. Aksariyat masalalar yuzaga keladigan turli holatlarga bog'liq ravishda mos qaror qabul qilishni (yechimni) talab etadi. C++ tili programmaning alohida bo'laklarining bajarilish tartibini boshqarishga imkon beruvchi qurilmalarning yetarlicha katta majmuasiga ega. Masalan, programma bajarilishining birorta qadamida qandaydir shartni tekshirish natijasiga ko'ra boshqaruvni programmaning u yoki bu bo'lagiga uzatish mumkin (tarmoqlanuvchi algoritmlar). Tarmoqlanishni amalga oshirish uchun shartli operatorlardan foydalaniladi.

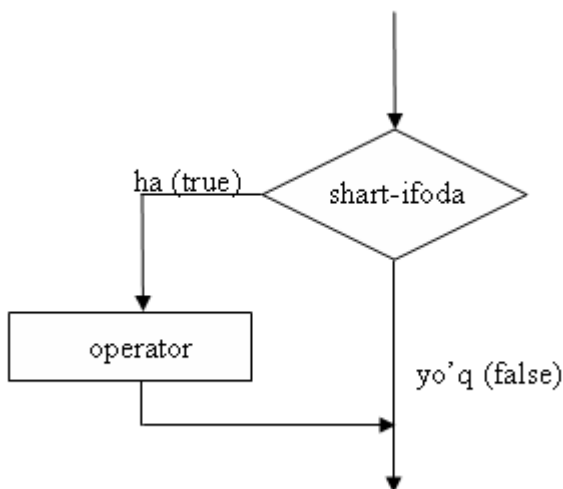
If operatori: if operatori qandaydir shartni rostlikka tekshirishi natijasiga ko'ra programmada tarmoqlanishni amalga oshiradi:

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

Bu yerda <shart> har qanday ifoda bo'lishi mumkin odatda u taqqoslash amali bo'ladi.

Agar shart 0 qiymatidan farqli yoki rost (true) bo'lsa, <operator> bajariladi, aks holda, ya'ni shart 0 yoki yolg'on (false) bo'lsa, hech qanday amal bajarilmaydi va

boshqaruv if operatoridan keyingi operatorga o'tadi (i (agar u mavjud bo'lsa). Ushbu holat 7 –rasmda ko'rsatilgan.



7-rasm. if() shart operatorining blok sxemasi

C++ tilining qurilmalari operatorlarni blok ko'rinishida tashkil qilishga imkon beradi. Blok – '{' va '}' belgi oralig'iga olingan operatorlar ketma-ketligi bo'lib, u kompilyator tomonidan yaxlit bir operator deb qabul qilinadi. Blok ichida e'lon operatorlari ham bo'lishi mumkin va ularda e'lon qilingan o'zgaruvchilar faqat shu blok ichida ko'rinadi (amal qiladi), blokdan tashqarida ko'rinmaydi. Blokdan keyin ';' belgisi qo'yilmasligi mumkin, lekin blok ichidagi har bir ifoda ';' belgisi bilan yakunlanishi shart.

Quyida keltirilgan programmada if operatoridan foydalanish ko'rsatilgan.

```
#include <iostream.h>

int main()
{
    int b;
    cin>>b;
    if (b>0)
    {          //b>0 shart bajarilgan holat
    ...
    cout<<"b- musbat son";
    ...
    }
    if (b<0)
```

```

cout<<"b – manfiy son"; //b<0 shart bajarilgan holat
return 0;
}

```

Programma bajarilishi jarayonida butun turdagi b o‘zgaruvchi e‘lon qilinadi va uning qiymati klaviaturadan o‘qiladi. Keyin b qiymatini 0 sonidan kattaligi tekshiriladi, agar shart bajarilsa(true), u holda ‘{‘ va ‘}’ belgilar ichidagi operatorlar bajariladi va ekranga “b – musbat son” xabari chiqadi. Agar shart bajarilmasa, bu operatorlar cheklab o‘tiladi. Navbatdagi shart operatori b o‘zgaruvchi qiymati manfiylikka tekshiradi, agar shart bajarilsa, yagona cout ko‘rsatmasi bajariladi va ekranga “b – manfiy son” xabari chiqadi.

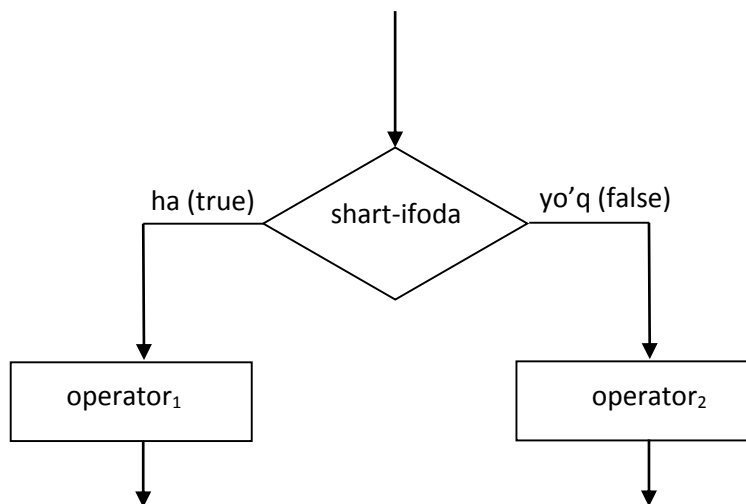
If – else operatori:Shart operatorining if – else ko‘rinishi quyidagicha:

```

if (<shart-ifoda>) <operator1>; else <operator2>;

```

Bu yerda <shart-ifoda> 0 qiymatidan farqli yoki true bo‘lsa <operator₁>, aks holda <operator₂> bajariladi. If-else shart operator mazmuniga ko‘ra algoritmning tarmoqlanuvchi blokini ifodalaydi: <shart-ifoda> - shart bloki (romb) va <operator₁> > blokning «ha» shoxiga, <operator₂> esa blokning «yo‘q» shoxiga mos keluvchi amallar bloklari deb qarash mumkin .



8-rasm. if(), else shart operatorining blok sxemasi

Misol tariqasida diskriminantni hisoblash usuli yordamida $ax^2 + bx + c = 0$ ko‘rinishidagi kvadrat tenglama ildizlarini topish masalasini ko‘raylik:

```

#include <iostream.h>

```

```

#include <math.h>

```

```

int main()

```

```

{
float a,b,c;
float D,x1,x2;
cout<<'ax^2+bx+c=0; tenglama ildizini topish .';
cout<<'\\n a- koiffitsiyentni kiriting: ';
cin>>a;
cout<<'\\n b- koeffitsientni kiriting: ';
cin>>b;
cout<<'\\n c- koeffitsientni kiriting: ';
cin>>c;
D=b*b-4*a*c;
If(D<0)
{cout<<'tenglama haqiqiy ildizga ega emas!';
return 0;
}
if (D==0)
{cout<<'tenglama yagona ildizga ega: ';
x1=-b/(2*a);
cout<<'\\nx='<<x1;
return 0;
}
else
{cout<<'tenglama yagona ildizga ega: ';
x1=(-b+sqrt(D))/(2*a);
x1=(-b-sqrt(D))/(2*a);
cout<<'\\nx1='<<x1;
cout<<'\\nx1='<<x1;
}
return 0;

```


}

Programma bajarilganda, birinchi navbatda tenglama koeffitsientlari – a, b, s o‘zgaruvchilar qiymatlari kiritiladi, keyin diskriminant – D o‘zgaruvchi qiymati hisoblanadi. Keyin D qiymatining manfiy ekanligi tekshiriladi. Agar shart o‘rinli bo‘lsa, yaxlit operator sifatida keluvchi ‘{‘ va ‘}’ belgilari orasidagi operatorlar bajariladi va ekranga “Tenglama haqiqiy ildizlarga ega emas” xabari chiqadi va programma o‘z ishini tugatadi (“return 0;” operatorini bajarish orqali). Diskriminant noldan kichik bo‘lmasa, navbatdagi shart operatori uni nolga tengligini tekshiradi. Agar shart o‘rinli bo‘lsa, keyingi qatorlardagi operatorlar bloki bajariladi – ekranga “ Tenglama yagona ildizga ega:” xabari, hamda x1 o‘zgaruvchi qiymati chop qilinadi va programma shu yerda o‘z ishini tugatadi, aks holda, ya’ni D qiymatni noldan katta holati uchun else kalit so‘zidan keyingi operatorlar bloki bajariladi va ekranga “Tenglama ikkita ildizga ega: “ xabari, hamda x1 va x2 o‘zgaruvchilar qiymatlari chop etiladi. Shu bilan shart operatoridan chiqiladi va asosiy funksiyaning return ko‘rsatmasini bajarish orqali programma o‘z ishini tugatadi.

O‘z navbatida <operator₁> va <operator₂> ham shartli operator bo‘lishi mumkin. Ifodadagi har bir else kalit so‘zi, oldindagi eng yaqin if kalit so‘ziga tegishli hisoblanadi (xuddi ochiluvchi va yopiluvchi qavslardek). Buni inobatga olmaslik mazmunan xatoliklarga olib kelishi mumkin.

Masalan:

```
if (x==1)
```

```
if (y==1) cout<<"x=1 va y=1";
```

```
else cout <<"x<>1";
```

Bu misolda «x<>1 » xabari x qiymatini 1 ga teng va u qiymatini 1 ga teng bo‘lmagan holda ham chop etiladi. Quyidagi variantda ushbu mazmunan xatolik bartaraf etilgan:

```
If (x==1)
```

```
{
```

```
If (y==1) cout<<"x=1 va y=1";
```

```
}
```

```
else cout<<"x<>1";
```

Ikkinchi misol tariqasida uchta butun sonning maksimal qiymatini topadigan programma bo‘lagini keltirishimiz mumkin:

...

```

int x,y,z,max;
cin >>x>>y>>z;
if (x>y)
if (y<z) max=z;
else max=y;
else
if (x<z) max=z;
else max=x;
...

```

Shart operatorida e'lon qilish operatorlarini ishlatish man etiladi, lekin undagi bloklarda o'zgaruvchilarni e'lon qilish mumkin va bu o'zgaruvchilar faqat blok ichida amal qiladi. Quyidagi misolda bu holat bilan bog'liq xatolik ko'rsatilgan:

```
If (j>0) {int i; i=2*j;}
```

Else i=-j;//xato, chunki I blokdan tashqarida ko'rinmaydi

Masala. Berilgan to'rt xonali ishorasiz sonning boshidagi ikkita raqamining yig'indisi qolgan raqamlar yig'indisiga teng yoki yo'qligi aniqlansin (raqamlar yig'indisi deganda ularga mos son qiymatlarining yig'indisi tushuniladi). Sonning raqamlarini ajratib olish uchun butun sonlar arifmetikasi amallaridan foydalaniladi:

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

```
Int main()
```

```
{
```

```
Unsigned int n,a3,a2,a1,a0;//n=a3a2a1a0 ko'rinishida
```

```
cout<<"\nn-qiymatini kiriting:";
```

```
cin>>n;
```

```
If(n<1000| n>9999)
```

```
{
```

```
cout<<"kiritilgan son 4 xonali emas!";
```

```
return 1;
```

```
}
```

```
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 ishorasiz butun son kiritishni taklif qiladi. Agar kiritilgan son 4 xonali bo‘lmasa ($n < 1000$ yoki $n > 9999$), bu haqda xabar beriladi va programma o‘z ishini tugatadi. Aks holda n sonining raqamlari ajratib olinadi, hamda boshidagi ikkita raqamning yig‘indisi – (a_3+a_2) qolgan ikkita raqamlar yig‘indisi – (a_1+a_0) bilan solishtiriladi va ularning teng yoki yo‘qligiga qarab mos javob chop qilinadi.

?: shart amali: Agar tekshirilayotgan shart nisbatan sodda bo‘lsa, shart amalining $\langle\langle ? \rangle\rangle$ ko‘rinishini ishlatish mumkin:

$\langle\text{shart ifoda}\rangle ? \langle\text{ifoda}_1\rangle : \langle\text{ifoda}_2\rangle ;$

Shart amali if shart operatoriga o‘xshash holda ishlaydi: agar $\langle\text{shart ifoda}\rangle > 0$ qiymatidan farqli yoki true bo‘lsa, $\langle\text{ifoda}_1\rangle$, aks holda $\langle\text{ifoda}_2\rangle$ bajariladi. Odatda ifodalar qiymatlari birorta o‘zgaruvchiga o‘zlashtiriladi.

Misol tariqasida ikkita butun son maksimumini topish ko‘raylik.

```

#include <iostream.h>
int main()
{
int a,b,c;
cout<<"a va b sonlar maksimumini topish programmasi."";
cout<<"\n a- qiymatni kiriting: ";
cin>>a;
cout<<"\n b- qiymatni kiriting: ";
cin>>b;
c=a>b?a:b;
cout<<"\nSonlar maksimumi: "<<c;
return 0;
}

```

```
}
```

Programmada shart operatori qiymat berish operatorining tarkibiga kirgan bo'lib, a o'zgaruvchining qiymati b o'zgaruvchining qiymatidan kattaligi tekshiriladi, agar shart rost bo'lsa c o'zgaruvchiga a o'zgaruvchi qiymatini, aks holda b o'zgaruvchining qiymatini va c o'zgaruvchisining qiymati chop etiladi.

?: amalining qiymat qaytarish xossasidan foydalangan holda, uni bevosita cout ko'rsatmasiga yozish orqali ham qo'yilgan masalani yechish mumkin:

```
#include <iostream.h>

int main()
{
    int a,b;

    cout<<"a va b sonlar maksimumini topish programmasi.";
    cout<<"\n a- qiymatni kiriting: ";
    cin>>a;
    cout<<"\n b- qiymatni kiriting: ";
    cin>>b;
    c=a>b?a:b;
    cout<<"\nSonlar maksimumi: "<<(a>b) ?a:b;
    return 0;
}
```

Switch operatori: Shart operatorining yana bir ko'rinishi switch tarmoqlanish operatori bo'lib, uning sintaksisi quyidagicha:

```
switch (<ifoda>)
{
    case <o'zgarmas ifoda1> : <operatorlar guruhi1>; break;
    case <o'zgarmas ifoda2> : <operatorlar guruhi2>;break;
    ...
    case <o'zgarmas ifodan> : <operatorlar guruhin>; break;
    default:: <operatorlar guruhin+1>;
}
```

Bu operator quyidagicha amal qiladi: birinchi navbatda <ifoda> qiymati hisoblanadi, keyin bu qiymat case kalit so‘zi bilan ajratilgan <o‘zgarmas ifoda_i> bilan solishtiriladi. Agar ular ustma-ust tushsa, shu qatordagi ‘.’ belgisidan boshlab, toki break kalit so‘zigacha bo‘lgan <operatorlar guruhi_i> bajariladi va boshqaruv tarmoqlanuvchi operatoridan keyingi joylashgan operatorga o‘tadi. Agar <ifoda> birorta ham <o‘zgarmas ifoda_i> bilan mos kelmasa, qurilmaning default qismidagi <operatorlar guruhi_{n+1}> bajariladi. Shuni qayd etish kerakki, qurilmada default kalit so‘zi faqat bir marta uchrashi mumkin.

Misol uchun, kirish oqimidan “Jarayon davom etilsinmi?” so‘roviga foydalanuvchi tomonidan javob olinadi. Agar ijobiy javob olinsa, ekranga “Jarayon davom etadi!” xabari chop etiladi va programma o‘z ishini tarmoqlanuvchi operatoridan keyingi operatorlarni bajarish bilan davom ettiladi, aks holda “Jarayon tugadi!” javobi beriladi va programma o‘z ishini tugatadi. Bunda, foydalanuvchining ‘y’ yoki ‘Y’

javoblari jarayonni davom ettirishni bildiradi, boshqa belgilar esa tugatishni anglatadi.

```
#include <iostream.h>

int main()
{
    char javob=' ';
    cout<<"jarayon davom etsinmi?('y','Y'):";
    cin>>javob;
    switch(javob)
    {
        case 'y':
        case 'Y':
            cout<<"jarayon davom etadi!\n";
            break;
        default:
            cout<<"jarayon tugadi!\n";
    }
    return 0;
}
```

```
...//jarayon
return 0;
}
```

Umuman olganda, tarmoqlanuvchi operatorlarda break va default kalit ishlatish shart emas. Lekin bu holda operatorning mazmuni mumkin. Masalan, default qismi bo‘lmagan holda, agar <ifoda> birorta <o‘zgarmas ifoda_i> bilan ustma-ust tushmasa, operator hech qanday amal bajarmasdan boshqaruv tarmoqdanuvchi operatorlardan keyingi operatorga o‘tadi. Agar break bo‘lmasa, <ifoda> birorta <o‘zgarmas ifoda_i> bilan ustma-ust tushgan holda, unga mos keluvchi operatorlar guruhini bajaradi va «to‘xtamasdan» keyingi qatordagi operatorlar guruhini bajarishga o‘tib ketadi. Masalan, yuqoridagi misolda break operatori bo‘lmasa va jarayonni davom ettirishni tasdiqlovchi ('Y') javob bo‘lgan taqdirda ekranga

Jarayon davom etadi!

Jarayon tugadi!

Xabarlarini chiqadi va programma o‘z ishini tugatadi (return operatorining bajarilishi natijasida).

Tarmoqlanuvchi operator sanab o‘tiluvchi turdagi o‘zgarmaslar bilan birgalikda ishlatilganda samara beradi. Quyidagi programmada ranglar gammasini toifalash masalasi yechilgan.

```
#include <iostream.h>
int main()
{
enum Ranglar{qizil, tuq_sariq,sariq,yashil,kuk,zangori,binafsha};
Ranglar rang;
//...
switch(rang)
{
case qizil:
case tuq_sariq:
case sariq:
cout<<"Issiq gamma tanlanadi. \n";
break;
```

```

case yashil:
case kuk:
case zangori:
case binafsha:
cout<<"Sovuq gamma tanlandi.\n";
break;
default:
cout<<"Kamalak bunday rangga ega emas. \n";
}
return 0;
}

```

Programma bajarilishida boshqaruv tarmoqlanuvchi operatorga kelganda, rang qiymati qizil yoki tuq_sariq yoki sariq bo'lsa, "Issiq gamma tanlandi" xabari, agar rang qiymati yashil yoki kuk yoki zangori yoki binafsha bo'lsa, ekranga "Sovuq gamma tanlandi" xabari, agar rang qiymati sanab o'tilgan qiymatlardan farqli bo'lsa, ekranga "Kamalak bunday rangga ega emas" xabari chop etiladi va programma o'z ishini tugatadi.

Switch operatorida e'lon operatorlari ham uchrashi mumkin. Lekin switch operatori bajarilishida «sakrab o'tish» holatlari bo'lishi hisobiga blok ichidagi ayrim e'lonlar bajarilmasligi va buning oqibatida programma ishida xatolik ro'y berishi mumkin:

```

//...
int k=0,n=0;
cin>>n;
switch (n)
{
int=10; //xato,bu operator hech qachon bajarilmaydi
case 1:
int j=20; //agar n=2 bo'lsa,bu e'lon bajarilmaydi
case 2:
k+=i+j; //xato, chunki i,j o'zgaruvchilar noma'lum

```

```
}  
cout<<k;  
//...
```

Masala. Quyida, sanab o‘tiluvchi turlar va shu turdagi o‘zgaruvchilar e’lon qilingan:

```
enum Birlik{detsimetr, kilometr, metr, millimetr, santimetr}  
float y; Birlik r;  
Birlikda berilgan x o‘zgaruvchisining qiymat metrlarda chop qilinsin.
```

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



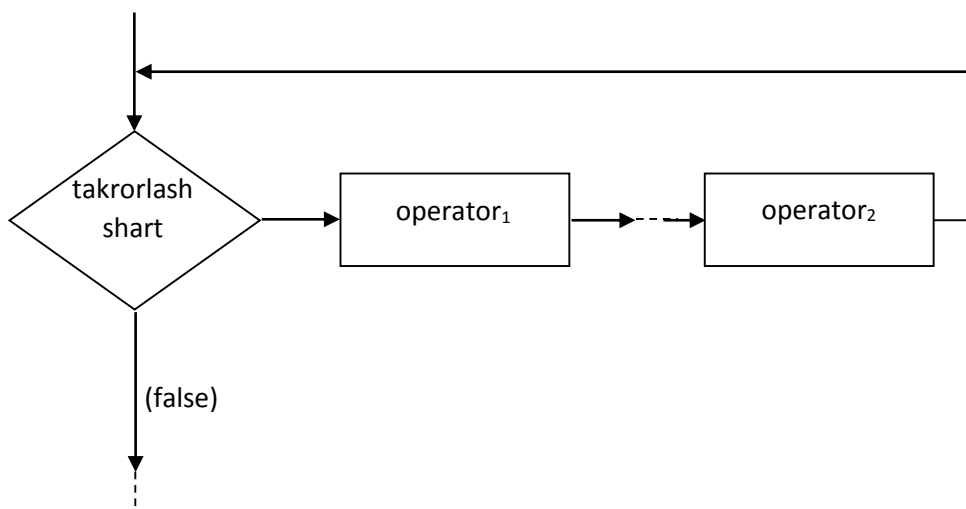
```

case millimetr: y=x/1000; break;
case santimetr: y=x/100; break;
default:
    cout<<"uzunlik birligi noto'g'ri kiritildi!";
    return 0;
}
cout<<y<<"metr";
return 0;
}

```

Programma bajarilishini boshqarishning boshqa bir kuchli mexanzmlaridan biri – takrorlash operatorlari hisoblanadi.

Takrorlash operatori «takrorlash sharti» deb nomlanuvchi ifodaning rost qiymatida programmaning ma’lum bir qismidagi operatorlarni (takrorlash tanasini) ko‘p marta takror ravishda (itarativ jarayon).



9-rasm. Takrorlash operatorining blok sxemasi

Takrorlash o‘zining kirish va chiqish nuqtalariga ega, lekin chiqish nuqtasining bo‘lmasligi mumkin. Bu holda takrorlashga cheksiz takrorlash deyiladi. Cheksiz takrorlash uchun takrorlashni davom ettirish sharti doimo rost bo‘ladi.

Takrorlash shartini tekshirish takrorlash tanasidagi operatorlarni bajarishdan oldin tekshirilishi mumkin (for, while takrorlashlari) yoki takrorlash tanasidagi operatorlari bir marta bajarilgandan keyin tekshirilishi mumkin (do-while).

Takrorlash operatorlari ichma-ich joylashgan bo‘lishi mumkin.

for takrorlash operatori: *for* takrorlash operatorining sintaksisi quyidagi ko‘rinishga ega:

for (<ifoda₁>; <ifoda₂>;<ifoda₃>) <operator yoki blok>;

Bu operator o'z ishini <ifoda₁> ifodasini bajarishdan boshlaydi. Keyin takrorlash qadamlari boshlanadi. Har bir qadamda <ifoda₂> bajariladi, agar natija 0 qiymatidan farqli yoki true bo'lsa, takrorlash tanasi - <operator yoki blok> bajariladi va oxirida <ifoda₃> bajariladi. Agar <ifoda₂> qiymati 0 (false) bo'lsa, takrorlash jarayoni to'xtaydi va boshqaruv takrorlash operatoridan keyingi operatorga o'tadi. Shuni qayd qilish kerakki, <ifoda₂> ifodasi vergul bilan ajratilgan bir nechta ifodalar birlashmasidan iborat bo'lishi mumkin, bu holda oxirgi ifoda qiymati takrorlash sharti hisoblanadi. Takrorlash tanasi sifatida bitta operator, jumladan bo'sh operator bo'lishi yoki operatorlar bloki kelishi mumkin.

Misol uchun 10 dan 20 gacha bo'lgan butun sonlar yig'indisini hisoblash masalasini ko'raylik.

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

Programmada takrorlash operatori o'z ishini, I takrorlash parametriga (takrorlash sanagichiga) boshlang'ich qiymat – 10 sonini berishdan boshlaydi va har bir takrorlash qadamidan (itaratsiyadan) keyin uning qiymati bittaga oshadi (qavs ichidagi uchinchi operator bajarilishini hisobiga). Har bir takrorlash qadamida takrorlash tanasidagi operator bajariladi, ya'ni Summa o'zgaruvchisiga 1 qiymati qo'shiladi. Takrorlash sanagichi i qiymati 21 bo'lganda "i<=20" takrorlash sharti (0-qiymati) bo'ladi va takrorlash tugaydi. Natijada boshqaruv takrorlash operatoridan keyingi cout operatoriga o'tadi va ekranga yig'indi chop etiladi.

Yuqorida keltirilgan misolga qarab takrorlash operatorlarining qavs ichidagi ifodalariga izoh berish mumkin:

<ifoda₁> - takrorlash sanagichi vazifasini bajaruvchi o'zgaruvchisiga boshlang'ich qiymat berishga xizmat qiladi va u takrorlash jarayoni boshida faqat bir marta hisoblanadi. Ifodada o'zgaruvchi e'loni uchrash mumkin va bu o'zgaruvchi takrorlash operatori tanasida amal qiladi va takrorlash operatoridan tashqarida «ko'rinmaydi» (C++ Builder kopiilyatori uchun);

<ifoda₂> - takrorlashni bajarish yoki yo‘qligini aniqlab beruvchi mantiqiy ifoda, agar shart rost bo‘lsa, takrorlash davom etadi, aks holda yo‘q. Agar bu ifoda bo‘sh bo‘lsa, shart doimo rost deb hisoblanadi;

<ifoda₃> - odatda takrorlash sanagichning qiymatini oshirish (kamaytirish) uchun xizmat qiladi yoki unda takrorlash shartiga ta’sir boshqa amallar bo‘lishi mumkin.

Takrorlash operatorida qavs ichidagi ifodalar bo‘lmasligi mumkin, lekin sintaksis ‘;’ bo‘lmasligiga ruxsat bermaydi. Shu sababli sodda ko‘rinishdagi takrorlash operatori quyidagicha bo‘ladi:

```
for(;;) cout <<"Cheksiz takrorlash...";
```

Agar takrorlash jarayonida bir nechta o‘zgaruvchilarning qiymati sinxron ravishda o‘zgarishi kerak bo‘lsa, <ifoda₁> va <ifoda₃> ifodalarida zarur operatorlarni ‘,’ bilan yozish orqali bunga erishish mumkin:

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

Takrorlash operatorining har bir qadamida j va I o‘zgaruvchi qiymatlari mos ravishda o‘zgarib boradi.

For operatorida takrorlash tanasi bo‘lmasligi ham mumkin. Masalan, programma bajarilishini ma’lum bir muddatga «to‘xtab» turish zarur bo‘lsa, bunga takrorlashni hech qanday qo‘shimcha ishlarni bajarmasdan amal qilishi orqali erishish mumkin:

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

Yuqorida keltirilgan 10 dan 20 gacha bo‘lgan sonlar yig‘indisini bo‘sh tanali (bo‘sh operatorli) takrorlash operatori orqali hisoblash mumkin:

```

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

```

Takrorlash operatori tanasi sifatida operatorlar bloki ishlatishini faktorialni hisoblash misolida ko'rsatish mumkin:

```

#include <iostream.h>
int main()
{
int a;
unsigned long fact=1;
cout<<"butun sonni kiriting: _";
cin>>a;
if ((a>=0)&&(a<33))
{
for (int i=1; i<=a; i++) fact*=i;
cout<<a<<"!="<<fact<<"\n";
}
return 0;
}

```

Programma foydalanuvchi tomonidan 0 dan 33 gacha oraliqdagi son kiritilganda amal qiladi, chunki 34! Qiymati unsigned long uchun ajratilgan razryadlarga sig'maydi.

Masala. Takrorlash operatorining ichma-ich joylashuviga misol sifatida raqamlari bir-biriga o'zaro teng bo'lmagan uch xonali natural sonlarni o'sish tartibida chop qilish masalasini ko'rishimiz mumkin:

```

#include <iostream.h>
int main()
{
unsigned char a2,a1,a0; // uch xonali son raqamlari
for (a2='1'; a2<='9'; a2++) //sonning 2-o'rindagi raqami
for (a1='0'; a1<='9'; a1++) //sonning 1-o'rindagi raqami
for (a0='0'; a0<='9'; a0++) //sonning 0-o'rindagi raqami

```

```

// raqamlarni o'zaro teng emasligini tekshirish
if(a0!=a1 && a1!=a2 && a0!=a2) //o'zaro teng emas
cout<<a2<<a!<<a0<<'\n' ;
return 0;
}

```

Programmada uch xonali sonning har bir raqami takrorlash operatorlari yordamida hosil qilinadi. Birinchi, tashqi takrorlash operatori bilan 2-xonadagi raqam (a2 takrorlash parametri) hosil qilinadi. Ikkinchi, ichki takrorlash operatorida (a1 takrorlash parametri) son ko'rinishining 1-xonasidagi raqam va nihoyat, unga nisbatan ichki bo'lgan a0 parametrli takrorlash operatorida 0-xonadagi raqamlar hosil qilinadi. Har bir tashqi takrorlashning bir qadamiga ichki takrorlash operatorining to'liq bajarilishi to'g'ri keladi.

While takrorlash operatori: *while* takrorlash operatori, operator yoki blokni takrorlash sharti yolg'on (false yoki 0) bo'lguncha takror bajaradi. U quyidagi sintaksisga ega:

```
while (<ifoda>) <operator yoki blok>;
```

Agar <ifoda> rost qiymatli o'zgarmas ifoda bo'lsa, takrorlash cheksiz bo'ladi. Xuddi shunday, <ifoda> takrorlash boshlanishida rost bo'lib, uning qiymatiga takrorlash tanasidagi hisoblash ta'sir etmasa, ya'ni uning qiymati o'zgarmasa, takrorlash cheksiz bo'ladi.

While takrorlash shartini oldindan tekshiruvchi takrorlash operatori hisoblanadi. Agar takrorlash boshida <ifoda> yolg'on bo'lsa, while operatori tarkibidagi <operator yoki blok> qismi bajarilmasdan cheklab o'tiladi.

Ayrim hollarda <ifoda> qiymat berish operatori ko'rinishida kelishi mumkin. Bunda qiymat berish amali bajariladi va natija 0 solishtiriladi. Natija noldan farqli bo'lsa, takrorlash davom ettiriladi.

Agar rost ifodaning qiymati noldan farqli o'zgarmas bo'lsa, cheksiz takrorlash ro'y beradi. Masalan:

```
While(1); // cheksiz takrorlash
```

Xuddi for operatoridek, ',' yordamida <ifoda> da bir nechta amallar sinxron ravishda bajarish mumkin. Masalan, son va uning kvadratlarini chop qiladigan programmada ushbu holat ko'rsatilgan:

```

#include <iostream.h>
int main()
{
int n, n2;

```

```

cout<<"sonni kiriting(1..10):_";
cin>>n;
n++;
while(n--,n2=n*n, n>0)
cout<<"n="<<n<<" n^2="<<n2<<endl;
return 0;
}

```

Programmada takrorlash operatori bajarilishida n soni 1 gacha kamayib boradi. Har bir qadamda n va uning kvadrati chop qilinadi. Shunga e'tibor berish kerakki, shart ifodasida operatorlarni yozilish ketma-ketligining ahamiyati bor, chunki, eng oxirgi operator takrorlash sharti sifatida qaraladi va n qiymati 0 bo'lganda takrorlash tugaydi.

Keyingi programmada berilgan o'nlik sonning ikkilik ko'rinishini chop qilish masalasini yechishda while operatorini qo'llash ko'rsatilgan.

```

#include <iostream.h>
int main()
{int sanagich=4;
short son10, jarayon=1;
while (jarayon) // cheksiz takrorlash
{cout<<"o'nlik sonni kiriting(0..15)_";
cin>>son10;
cout<<'\n'<<son10<<"sonining ikkilik ko'rinishi:";
while (sanagich)
{if(son10&8) //son10&00001000
cout<<'1';
else cout<<'0';
son10=son10<<1 //razryadlarni 1o'rin chapga surish
sanagich--;
}
cout<<'\n';
cout<<"jarayonni to'xtatish(0), davom ettirish(1):_";
cin>>jarayon;
}
}

```

```

    sanagich=4;
}
return 0;
}

```

Programmada ichma-ich joylashgan takrorlash operatorlari ishlatilgan. Birinchisi, sonning ikkilik ko‘rinishini chop qilish jarayonini davom ettirish sharti bo‘yicha amal qiladi. Ichki joylashgan, ikkinchi takrorlash operatoridagi amallar har qanday, 0 dan 15 gacha bo‘lgan sonlar to‘rtta razryadli ikkilik son ko‘rinishida bo‘lishiga asoslangan. Unda kiritilgan sonning ichki, ikkilik ko‘rinishida uchinchi razryadida 0 yoki 1 turganligi aniqlanadi (“son10& 8”). Shart natijasi natija 1 (rost) bo‘lsa, ekranga ‘1’, aks holda ‘0’ belgisi chop etiladi.

Keyingi qadamda son razryadlari chapga bittaga suriladi va yana uchinchi razryaddagi raqam chop etiladi. Takrorlash sanagich qiymati 0 bo‘lguncha davom etadi (to‘rt marta) va boshqaruv ichki takrorlash operatoridan chiqadi.

While takrorlash operatori yordamida samarali programma kodi yozishga bir misol bu – ikkita natural sonlarning eng katta umumiy bo‘luvchisi (EKUB) Evklid algoritmi bilan topish masalasini ko‘rishimiz mumkin:

```

int main()
{int a,b;
cout<<"A va B natural sonlar EKUBini topish.\n";
cout<<"A va B natural sonlarni kiriting:"
cin>>a>>b;
while (a!=b) a>b? a-=b:b-=a;
cout<<"bu sonlar
cin<<" bu sonlar EKUBi="<<a;
return 0;
}

```

Butun turdagi a va b qiymatlari oqimdan o‘qilgandan keyin ular qiymatlari toki o‘zaro teng bo‘lmaguncha takrorlash jarayoni ro‘y beradi. Takrorlashning har bir qadamida a va b sonlarining kattasidan- kichigi ayriladi va ular tengligi tekshiriladi. Takrorlashdan keyingi ko‘rsatma vositasida a o‘zgaruvchisining qiymati natija sifatida chop etiladi.

Do-while takrorlash operatori: *do-while* takrorlash operatori while operatoridan farqli ravishda oldin operator yoki blokni bajaradi, keyin takrorlash

shartini tekshiradi. Bu qurilma takrorlash tanasini kamida bir marta bajarilishini ta'minlaydi. Do-while takrorlash operatori quyidagi sintaksisga ega:

```
do <operator yoki blok>; while (<ifoda>);
```

Bunday takrorlash operatorining keng qo'llaniladigan holatlari bu takrorlashni boshlamasdan turib, takrorlash shartini tekshirishning iloji bo'lmagan holatlar hisoblanadi. Masalan, birorta jarayonni davom ettirish yoki to'xtatish haqidagi so'rovga javob olish va uni tekshirish zarur bo'lsin. Ko'rinib turibdiki, jarayonni boshlamasdan oldin bu so'rovni berishning ma'nosi yo'q. Hech bo'lmaganda takrorlash jarayonining bitta qadami amalga oshirilgan bo'lishi kerak:

```
#include <iostream.h>

int main()
{
    char javob;
    do
    {
        ...// programma tanasi
        cout<< " jarayonni to'xtashish(N):_ ";
        cin>>javob;
    }
    while(javob !=N)
    return 0;
}
```

Programma toki " jarayonni to'xtashish (N):_"so'roviga (N) javobi kiritilmaguncha davom etadi.

Bu operator ham cheksiz takrorlanishi mumkin:

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

Masala. Har qanday 7 katta butun sondagi pul miqdorini 3 va 5 so'mliklarda berish mumkinligi isbotlansin. Qo'yilgan masala $p=3n+5m$ tenglamasi qanotlantiruvchi m,n sonlar juftliklarini topish masalasidir (p -pul miqdori). Bu shartning bajarilishini m va n o'zgaruvchilarining mumkin bo'lgan qiymatlarining barcha kombinatsiyalarida tekshirish zarur bo'ladi.

```
#include <iostream.h>

int main()
```



```

{
    unsigned int pul; //pul- kiritiladigan pul miqdori
    unsigned n3,m5; //n-3 so'mliklar , m-5 so'mliklar soni
    bool xato=false; //pul qiymatini kiritilgandagi xatolik
    do
    {
        if (xato) cout<<"kiritilgan pul qiymati 7 dan kichik !";
        xato=true ; //keyingi takrorlash xato hisoblanadi
        cout<<"\npul qiymatini kiriting (>7):";
        cin>>pul;
    }
    while (pul<=7); // toki 7 sonidan katta sopl kiritulguncha
    n3=0 ; //birorta ham 3 so'mlik yo'q
    do
    {
        m5=0; // birorta ham 5 so'mlik yo'q
        do
        {
            if (3*n3+5*m5==pul)
                cout<<n3<<"ta 3 so'mlik + " <<m5<<" ta 5 so'mlik\n";
            m5++ //5so'mliklar bittaga oshiriladi
        }
        while(3*n3+5*m5<=pul);
        n3++; //3 so'mliklar bittaga oshiriladi
    }
    while(3*n3<=pul);
    return 0;
}

```

Programma pul qiymatini kiritishni so'raydi (pul o'zgaruvchiga). Agar pul qiymati 7 sonidan kichik bo'lsa, bu haqda xabar beriladi va takror ravishda qiymat kiritish talab qilinadi. Pul qiymati 7 dan katta bo'lganda, 3 va 5 so'mliklarning

mumkin bo'lgan to'la kombinatsiyasini amalga oshirish uchun ichma-ich takrorlashlar amalga oshiriladi. Tashqi takrorlash n_3 (3 so'mliklar miqdori) bo'yicha, ichki takrorlash esa m_5 (5 so'mliklar miqdori) bo'yicha, toki bu miqdordagi pullar qiymati pul qiymatidan oshib ketmaguncha davom etadi. Ichki takrorlashda m_5 o'zgaruvchisining har bir qiymatida « $3 \cdot n_3 + 5 \cdot m_5 = \text{pul}$ » sharti tekshiriladi, agar u o'rinli bo'lsa, yechim varianti sifatida n_3 va m_5 o'zgaruvchilar qiymatlari chop etiladi.

Pul qiymati 30 so'm kiritilganda, ekranga

0 ta 3 so'mlik +6 ta 5 so'mlik chop etiladi.

5 ta 3 so'mlik +6 ta 5 so'mlik

10 ta 3 so'mlik +0 ta 5 so'mlik

echim variantlari chop etiladi.

Berilganlar massivi tushunchasi: Xotirada ketma-ket (regular) joylashgan bir xil turdagi qiymatlarga massiv deyiladi.

Odatda massivlarga zarurat, katta hajmdagi, lekin cheklangan miqdordagi va tartiblangan qiymatlarni qayta ishlash bilan bog'liq masalalarni yechishda yuzaga keladi. Faraz qilaylik, talabalar guruhining reyting ballari bilan ishlash masalasi qo'yilgan. Unda guruhning o'rtacha reytingini aniqlash, reytinglarni kamayishi bo'yicha tartiblash, konkret talabaning reytingi haqida ma'lumot berish va boshqa masala ostilarini yechish zarur bo'lsin. Qayd etilgan masalalarni yechish uchun berilganlarning (reytinglarning) tartiblangan ketma-ketligi zarur bo'ladi. Bu yerda tartiblanganlik ma'nosi shundaki, ketma-ketlikning har bir qiymati o'z o'rniga ega bo'ladi (birinchi talabaning reytingi massivda birinchi o'rinda, ikkinchi talabaniki – ikkinchi o'rinda va hakoza). Berilganlar ketma-ketligini ikki xil usulda hosil qilish mumkin. Birinchi yo'l – har bir reyting uchun alohida o'zgaruvchi aniqlash: Reyting₁, Reyting₂,... Reyting_N. Lekin ,guruhdagi talabalar soni yetarlicha katta bo'lganda, bu o'zgaruvchilar qatnashgan programmani tuzish katta qiyinchiliklarni yuzaga keltiradi. Ikkinchi yo'l – berilganlar ketma-ketligini yagona nom bilan aniqlab, uning qiymatlariga murojaatni, shu qiymatlarning ketma-ketlikda joylashgan o'rnining nomeri (indeksi) orqali amalga oshirishdir. Reytinglar ketma-ketligini Reyting deb nomlab, undagi qiymatlariga Reyting₁, Reyting₂,... Reyting_N ko'rinishida murojaat qilish mumkin. Odatda berilganlarning bunday ko'rinishiga massivlar deyiladi. Massivlarni matematikadagi sonlar vektoriga o'xshatish mumkin, chunki vektor ham o'zining individual nomiga ega va u fiksirlangan miqdordagi bir turdagi qiymatlardan – sonlardan iboratdir.

Demak, massiv – bu fiksirlangan miqdordagi ayrim qiymatlarning (massiv elementlarining) tartiblangan majmuasidir. Barcha elementlar bir xil turda bo'lishi kerak va bu tur element turi yoki massiv uchun tayanch tur deb nomlanadi. Yuqoridagi keltirilgan misolda Reyting – haqiqiy turdagi vektor deb nomlanadi.

Programmada ishlatiladigan har bir konkret massiv o'zining individual nomiga ega bo'lishi kerak. Bu nomni to'liq o'zgaruvchi deyiladi, chunki uning qiymati to'liq bir massivning o'zi bo'ladi. Massivning har bir elementi massiv nomi, hamda kvadrat qavsga olingan va element selektori deb nomlanuvchi indeksni ko'rsatish orqali oshkor ravishda belgilanadi. Murojaat sintaksisi:

<massiv nomi >[<indeks>]

Bu ko'rinishga xususiy o'zgaruvchi deyiladi, chunki uning qiymati massivning alohida elementidir, Bizning misolda Reyting massivining alohida komponentalariga Reyting[1], Reyting[2],..., Reyting[N] xususiy o'zgaruvchilar orqali murojaat qilish mumkin. Boshqacha bu o'zgaruvchilarni indeksli o'zgaruvchilar deyiladi.

Umuman olganda indeks sifatida ifoda ishlatilishi mumkin. Ifoda qiymati massiv elementi nomerini aniqlaydi. Ifoda sifatida o'zgaruvchi ham olinishi mumkinki, o'zgaruvchining qiymati o'zgarishi bilan murojaat qilinayotgan massiv elementini aniqlovchi indeks ham o'zgaradi. Shunday qilib, programmadagi bitta indeksli o'zgaruvchi orqali massivning barcha elementlarini belgilash (aniqlash) mumkin. Masalan, Reyting[1] o'zgaruvchisi orqali I o'zgaruvchining qiymatiga bog'liq ravishda Reyting massivining turli (barcha) elementlariga murojaat qilish imkoni mavjud. Shuni qayd qilish kerakki, massiv indeksi sifatida butun son qo'llaniladi.

Haqiqiy turdagi (float,double) qiymatlar to'plami cheksiz bo'lganligi sababli ular indeksi sifatida ishlatilmaydi.

C++ tilida indeks doimo 0 dan boshlanadi, uning eng katta qiymati massiv e'lonidagi uzunlikdan bittaga kam bo'ladi.

Massiv e'loni quyidagicha bo'ladi:

<tur> <nom> [<uzunlik>]={boshlang'ich qiymatlar}

Bu yerda <uzunlik> - o'zgarimas ifoda. Misollar:

```
int m[6]={ 1,4,-5,2,10,3} ;
```

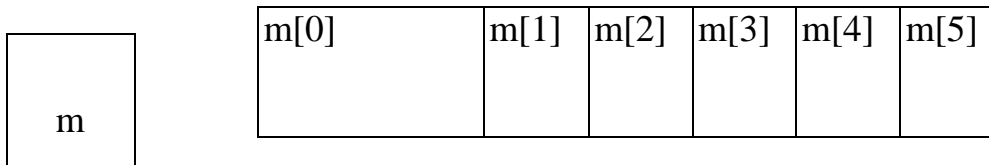
```
float a[4] ;
```

Massiv statik va dinamik bo'lishi mumkin, Statik massivning uzunligi oldindan ma'lum bo'lib, u xotirada ma'lum adresdan boshlab ketma-ket joylashadi. Dinamik massivni uzunligi programma bajarilish jarayonida aniqlanib, u dinamik xotiradagi ayni paytda bo'sh bo'lgan adreslarga joylashadi. Masalan,

```
int m[6] ;
```

ko'rinishida e'lon qilingan bir o'lchamli massiv elementlari xotirada quyidagicha joylashadi:

Adres	Qiymatlar
-------	-----------



10-rasm. Bir o‘lchamli massivning xotiradagi joylashuvi

Massivning i - elementiga $m[i]$ yoki $*(m+i)$ – vositali murojaat qilish mumkin. Massiv uzunligini `sizeof(m)` amali orqali aniqladi.

Massiv e‘lonida uning elementlariga boshlang‘ich qiymatlar olish (initsializatsiyalash) mumkin va uning bir nechta variantlari mavjud.

1) O‘lchami ko‘rsatilgan massiv elementlarini to‘liq initsializatsiyalash:

```
int t[5] = {-10, 5, 15, 4, 3};
```

Bunda 5 ta elementdan iborat bo‘lgan t nomli bir o‘lchamli massiv e‘lon qilingan va uning barcha elementlariga boshlang‘ich qiymatlar berilgan. Bu e‘lon quyidagi e‘lon bilan ekvivalent:

```
int t[5];
```

```
t[0] = -10; t[1] = 5; t[2] = 15; t[3] = 4; t[4] = 3;
```

1) O‘lchami ko‘rsatilgan massiv elementlarini to‘liqmas initsializatsiyalash:

```
int t[5] = {-10, 5, 15};
```

Bu yerda faqat massiv boshidagi uchta elementga boshlang‘ich qiymatlar berilgan. Shuni aytib o‘tish kerakki, massivning boshidagi yoki o‘rtadagi elementlariga qiymatlar bermasdan, uning oxiridagi elementlarga boshlang‘ich qiymat berish mumkin emas. Agarda massiv elementlariga boshlang‘ich qiymat berilmasa, unda kelishuv bo‘yicha `static` va `extern` modifikatori bilan e‘lon qilingan massiv uchun elementlarining qiymati 0 soniga teng deb, `automatic` massivlar elementlarining boshlang‘ich qiymatlari noma‘lum hisoblanadi.

3) O‘lchami ko‘rsatilmagan massiv elementlarini to‘liq initsializatsiyalash:

```
int t[] = {-10, 5, 15, 4, 3};
```

Bu misolda massivni barcha elementlariga qiymatlar berilgan hisoblanadi, massiv uzunligi kompilyator tomonidan boshlang‘ich qiymatlar soniga qarab aniqlanadi. Agarda massivni uzunligi berilmasa, boshlang‘ich qiymati berilishi shart.

Massivni e‘lon qilish misollari:

```
char ch[4] = { 'a', 'b', 'c', 'd' }; // belgilar massivi
```

```
int in[6] = { 10, 20, 30, 40 }; // butun sonlar massivi
```

```
char str[] = "abcd"; // satr uzunligi 5 ga teng, chunki
```

```
//uning oxiriga '\0' belgisi qo'shiladi
char str[]={ 'a', 'b', 'c', 'd'}; // yuqoridagi satrning
// boshqacha yozilishi
```

Masala. Bir oy ichidagi kundalik harorati berilgan, oy uchun o'rtacha haroratni hisoblash programmasi tuzilsin.

Programma matni:

```
void main()
{const int n=30;
int temp[n];
int I,s,temp_urtacha;
cout<<"Kunlik haroratni kiriting:\n"
for(i=0; i<n; i++)
{cout<<"\n temp[ "<<i<<" ]=";
cin>>temp[i]; }
for (i=0; i<n; i++)s+=temp[i];temp_urtacha=s/n;
cout<<"Kunlik harorat:\n";
for(i=0; i<n; i++) cout<<"\temp[ "<<i<<" ]="<<temp[i];
cout<<"Oydagi o'rtacha harorat="<<temp_urtacha;
return;
}
```

Ko'p o'lchamli statik massivlar: C++ tilida massivlar elementining turiga cheklovlar qo'yilmaydi, lekin bu turlar chekli o'lchamdagi obyektning turi bo'lishi kerak. Chunki kompilyator massivning xotiradan qancha joy (bayt) egallashini hisoblay olishi kerak. Xususan, massiv komponentasi massiv bo'lishi mumkin, ya'ni «vektorlar-vektori» natijada matritsa deb nomlanuvchi ikki o'lchamli massiv hosil bo'ladi.

Agar matritsaning elementi ham vektor bo'lsa, uch o'lchamli massivlar – kub hosil bo'ladi. Shu yo'l bilan yechilayotgan masalaga bog'liq ravishda ixtiyoriy o'lchamdagi massivlarni yaratish mumkin.

Ikki o'lchamli massivning sintaksisi quyidagi ko'rinishda bo'ladi:

$$\langle \text{tur} \rangle \langle \text{nom} \rangle [\langle \text{uzunlik} \rangle] [\langle \text{uzunlik} \rangle]$$

Masalan, 10 x 20 o'lchamli haqiqiy sonlar massivining e'loni:

```
float A[10] [20] ;
```

E'lon qilingan A matritsani ko'rinishi 11-rasmda keltirilgan.

J

$$\begin{aligned}
 a_0 &: (a_{00}, a_{02} \dots \dots a_{018}, a_{019}), \\
 a_1 &: (a_{10}, a_{11} \dots \dots a_{118}, a_{119} \dots), \\
 &\dots \\
 i \ a_i &: (\dots, \dots, \dots a_{ij} \dots, \dots, \dots), \\
 &\dots \\
 a_9 &: (a_{90}, a_{91} \dots \dots a_{918} \dots, a_{919} \dots).
 \end{aligned}$$

11-rasm. Ikki o'lchamli massivning xotiradagi joylashuvi

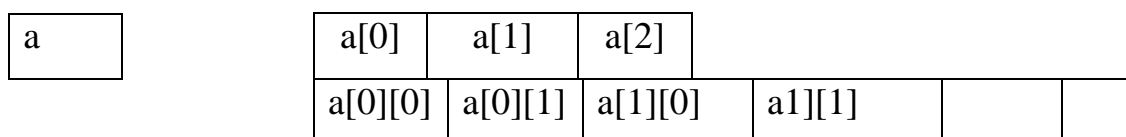
Endi adres nuqtai – nazaridan ko'p o'lchamli massiv elementlariga murojaat qilishni ko'raylik. Quyidagi e'lonlar berilgan bo'lsin:

```
int a[3][2];
```

```
float b[2][2][2];
```

Birinchi e'londa ikki o'lchamli massiv, ya'ni 2 satr va 3 ustundan iborat matritsa elon qilingan, ikkinchisida uch o'lchamli – 3 ta 2x2 magritsadan iborat bo'lgan massiv e'lon qilingan. Uning elementlariga murojaat sxemasi:

Ades Ko'rsatkichlar massivi Qiymatlar



11- rasm. Ikki o'lchamli elementlariga murojaat

Bu yerda a[i] ko'rsatkichda i-chi satrning boshlang'ich adresi joylashadi, massiv elementiga a[i][j] ko'rinishidagi asosiy murojaatdan tashqari vositali murojaat qilish mumkin: *((a+i)+j) yoki *(a[j]+j).

Bitiruv malakaviy ishda ko'zlangan dastur yaratish uchun zarur bo'lgan algoritmlarning bir qismi shulardan iborat.

1.2 Chop etish jarayonini boshqaruvchi dastur yaratish davomida qo'llaniladigan C++ Builder komponentalari

C++Builder 32 razryadli takomillashtirilgan Vizual Komponentalar Kutubxonasi VCL (Visual Component Library) bilan birgalikda etkazib beriladi. Bu kutubxona eng murakkab ilovalarni qurish uchun mo'ljallangan 100 dan ortiq takroran qo'llanadigan komponentalardan iborat. Kutubxonaning asosiy komponentalari Palitralar komponentalarining instrumental Panelida berilgan. Komponentalar belgilari dasturingiz shakliga olib o'tiladi. Quyida krill va lotin imlosi yozuvlarini o'giruvchi dastur yaratish jarayonida qo'llaniladigan komponentalar keltirilgan.

Kutubxona Windows va Windows 95 operatsiya tizimlaridagi Foydalanuvchi Grafik Interfeysi standart interfeys ob'ektlarining to'liq inkapsulalanishini o'z ichiga oladi. Ular orasida, ixtisoslashgan komponentalar bilan bir qatorda, relyatsion ma'lumotlar bazasini boshqarish uchun mo'ljallangan komponentalar alohida o'rin egallaydi. Ishonchli va samarali dasturlarni yaratishda C++Builder ob'ektga mo'ljallangan dasturlash (OMD) imkoniyatlaridan to'liq foydalanadi. C++ Builder bu OMD ekan, OLE (OCX) boshqaruvchi elementlarni kiritish uncha qiyinchilik tug'dirmaydi. O'z masalalaringiz talablarini kerakli darajada qondirish uchun, Kutubxonaning mavjud komponentalaridan foydalaning, hosila komponentalar imkoniyatlarini kengaytiring.

C++Builder bosh xususiyati avvalambor uning dasturni vizual ishlash jarayonida nafaqat tayyor komponentalardan foydalanish, balki yangi komponentalarni yaratish qobiliyatida ham namoyon bo'ladi. Yangi komponentalar, dastlabki komponentalar kabi, sodda bo'lishi mumkin, bunda ularning funksional imkoniyatlari sal-pal kengaytirilgan yoki o'zining mutlaqo o'ziga xos ko'rinishi, xulq-atvori va kodining mazmuni bilan farqlanadigan bo'ladi. Komponentalarning yaratilishi OMD ning vorislik mexanizmiga tayanadi, cheklanishlarga deyarli ega bo'lmaydi hamda quyidagi bosqichlardan o'tadi:

- * mavjud kponenta turiga vorislik;
- * yangi xususiyatlar, metodlar va voqealarni aniqlash;
- * yaratilgan komponentani qayd etish.

Qidirish oson bo'lishi uchun, Palitra funksional jihatdan o'xshash komponentalarni birlashtiradigan qo'shimcha ilovalar bilan bo'lingan. Tanlab olingan komponentaning kontekst menyusini unga sichqonchanning o'ng tugmasini bosib ochish mumkin.

Standart komponentalar: Komponentalar palitrasining Standard qo'shimcha ilovalari komponentalari sizning dasturingizga Windows standart interfeysli elementlarning 14 tasining ulanishini amalga oshiradi.

TMainMenu

Bosh menyu komandolari panelini va ularga mos keladigan tushib qoladigan menyularni yaratadi. Barcha menyu komandalarining identifikatorlari menyuning har qanday konkret komandasiga kirish huquqiga ega bo'lgan Items xususiyati bilan aniqlanadi, AutoMerge xususiyati Merge va Unmerge metodlari bilan birgalikda turli shakldagi menyularning birlashish jarayonini boshqaradi.

TpopUpMenu

Shakl yoki bironta boshqa komponenta uchun maxsus menyu yaratadi. E'tiborga oling, aynan shu maqsad uchun har qanday boshqa komponenta PopUpMenu xususiyatiga ega bo'lib, bu xususiyatda siz uning bilan bog'liq menyuga iqtibos qilishingiz mumkin.

Agar siz sichqonchanning o'ng tugmasini shaklga yoki berilgan komponenta mansub bo'lgan biron boshqa elementga bosish bilan maxsus menyu ekranda paydo bo'lishini xohlasangiz, AutoPopup xususiyatining true qiymatini o'rnatib. Voqea qayta ishlatgichi – OnPopup yordamida bevosita maxsus menyuning paydo bo'lishi oldindan bajariladigan protsedurani aniqlash mumkin.

Tlabel

Shaklda tahrir qilib bo'lmaydigan statik matnning to'rtburchak sohasini aks ettiradi. Odatda matn boshqa komponenta nomidan iborat bo'ladi.

Nom matni Caption xususiyatining qiymatidir. Alignment xususiyati matni tekislash usulini aniqlaydi. Shrift o'lchami avtomatik tarzda sohaning maksimal to'ldirilishiga mos kelishi uchun, AutoSize xususiyatining true qiymatini o'rnatib. Kalta soha ichida matnning hammasini ko'rish imkoniga ega bo'lish uchun, WordWrap xususiyatining true qiymatini bering. Transparent xususiyatining true qiymatini o'rnatib, boshqa komponentaning bir qismini to'g'ri uning ustida joylashtirilgan nom orasidan ko'rinib turadigan qilishingiz mumkin.

Tedit

Axborot yakka satrining tahrir qilinayotgan kiritishidagi to'rtburchak sohani shaklda aks ettiradi. Tahrir sohasining ichidagi boshlang'ich narsalarni Text xususiyatining qiymati bo'lgan satr aniqlaydi.

Tedit komponentasi TcustomEdit sinfining to'g'ridan-to'g'ri hosilasi bo'lib, uning barcha xususiyatlari, metodlari va voqelariga vorislik qiladi.

Tbutton

Yozuvli to'rtburchak tugmani yaratadi. Tugmacha bosilganda, dasturda biron-bir xatti-harakat nomlanadi (initsiallashtiriladi).

Tugmachalar ko'proq dialogli darchalarda qo'llanadi. Default xususiyatining true qiymati tomonidan tanlab olingan yashirin tugmacha, dialog darchasida har gal

Enter klavishchasi bosilganda, `OnClick` voqea qayta ishlatgichini ishga tushiradi. `Cancel` xususiyatining `true` qiymati tanlab olgan uzish tugmachasi, dialog darchasida har gal `Escape` klavishchasi bosilganda, `OnClick` voqea qayta ishlatgichini ishga tushiradi.

`Tbutton` komponentasi `TbuttonControl` sinfining hosilasi hisoblanadi.

TcheckBox

Ikkita holatga hamda tavsifiy matnga ega bo'lgan kvadrat chek-boksni yaratadi (bunda tavsifiy matn chek-boksning vazifasini spetsifikatsiya qiladi).

Boks holatini bildiruvchi «check» biron-bir variantning tanlanishiga mos keladi (boks ustidan tortilgan check bilan belgilanadi), «unchecked» holati esa tanlov olib tashlanishiga mos keladi – bunda `Checked` komponentasining xususiyati mos ravishda o'zgaradi hamda `OnClick` voqeasi yuzaga keladi. Tavsifiy matn `Caption` xususiyatida saqlanadi. `AllowGrayed` xususiyatining `true` qiymatini o'rnatib, boksni ko'qroq rangli (masalan, kulrang) qilish mumkin. `State` xususiyati joriy holatni va boks rangini aks ettiradi.

`TcheckBox` komponentasi `TbuttonControl` sinfining hosilasidir.

Tpanel

Boshqa komponentlarni o'z ichiga olishi mumkin bo'lgan bo'sh panelni yaratadi. Siz `Tpanel` dan o'z shaklingizda instrumentlar paneli yoki holatlar satrlarini yaratish uchun foydalanishingiz mumkin.

`Tpanel` panel komponentasi `TcustomPanel` sinfining hosilasi bo'lib, uning barcha xususiyatlar, metodlari va voqealari to'liq vorislik qiladi.

Windows komponentalari

Windows komponentalari sizning dasturingizga Windows ning 12 ta interfeys elementlarining ulanishini amalga oshiradi.

TOpenDialog* va *TSaveDialog

Juda ko'p dasturlarda ***TOpenDialog*** (faylni ochish) va ***TSaveDialog*** (faylni saqlash) komponentalaridan foydalaniladi. Bu komponentalarning xususiyatlari birxil faqat bajarish vazifalari turlicha faylni ochish va faylni saqlash.

Foydalanuvchi tomonidan tanlangan faylni satr tipida qaytarish uchun `FileName` xususiyati tanlanadi. Shuningdek ushbu xususiyatning qiymati dialogga kirishdan oldin ham o'rnatilishi mumkin. Shundan so'ng dialog oynasida Fayl nomi ko'rsatiladi.

`Filter` xususiyati orqali qidirish oynasida mavjud bo'lgan fayllardan faqat tanlangan tipdagi fayllarni ko'rinishini taminlaydi. `FilterIndex` xususiyati dialog oynasi ochilganda foydalanuvchiga ko'rsatiladigan filtr sonini bildiradi. Misol uchun, `FilterIndex` = 1 qiymati standart birinchi filtni o'rnatadi.

InitialDir xususiyati foydalanuvchi dialog oynasini ochganda odatda qaysi katalogni ochish kerakligini bildiradi. Agar foydalanuvchi tomondan xususiyatga qiymat berilmagan bo'lsa dialog oynasi ochilganda foydalanuvchi tomonidan oxirgi marta ochilgan katalogga murojat qilinadi.

DefaultExt xususiyati standart fayl kengaytmasi qiymatini belgilaydi. Agar foydalanuvchi ushbu xususiyatga qiymat bermasa muloqot oynasida faylning to'liq nomi va tupini ko'rsatishi lozim.

Title xususiyati muloqot oynasi sarlavhasini nomlashga imkon beradi. Agar bu xususiyatga qiymat berilmagan bo'lsa muloqot oynasi operatsion sistemaga mos sarlovha bilan ochiladi.

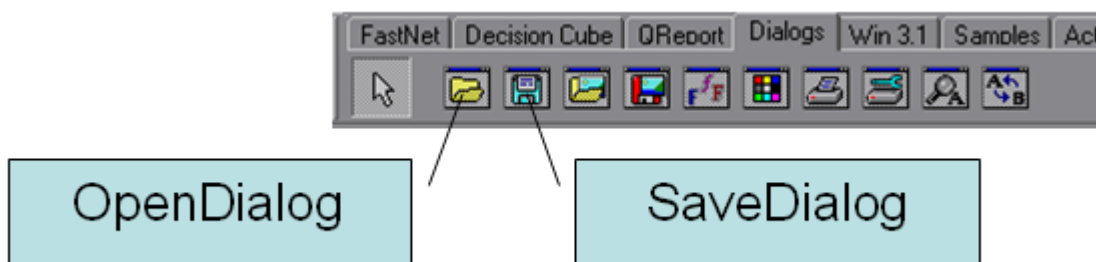
Options xususiyati faylni tanlash shartlarini belgilaydi. Ko'p foydalaniladigan xususiyatlarni o'z ichiga oladi, ular quidagilar:

3-jadval

ofAllowMultiSelect	Foydalanuvchiga bir nechta faylni tanlash imkonini beradi.
ofCreatePrompt	Agar foydalanuvchi mavjud bo'lmagan fayl nomini kiritisa, kiritilgan nom bilan yangi fayl hosil qilish haqida so'rov paydo bo'ladi.
ofEnableIncludeNotify	Muloqot oynasiga habar yuborishga ruhsat beradi.
ofEnableSizing	Foydalanuvchiga muloqot oynasi o'lchamlarini o'zgartirishga ruhsat beradi.
ofExtensionDifferent	Foydalanuvchi tomonidan tanlangan fayl kengaytmasi DefaultExt dan farq qilishini ko'rsatuvchi bayroqcha.
ofFileMustExist	Agar foydalanuvchi mavjud bo'lmagan fayl nomini kiritisa xatolik haqida habar beradi.
ofHideReadOnly	Muloqot oynasidan "faqat o'qish uchun" indikatorini o'chiradi.
ofNoChangeDir	OK tugmasi bosilgandan keyin faylni qidirish jarayonida ochilgan kataloglarni tiklaydi.
ofNoDereferenceLinks	Muloqot oynasida "tezkor tugmacha"lardan foydalanishni taqiqlaydi.
ofNoLongNames	Ismning faqat 8 ta belgisi va kengaytmaning uchta belgisi ko'rsatiladi.
ofNoNetworkButton	Muloqot oynasidan qidirish tugmasini olib tashlaydi. ofOldStyleDialog bayroqchasi o'rnatilgan holdagina ishlaydi.
ofNoReadOnlyReturn	Agar foydalanuvchi faylni faqat o'qish uchun tanlagan bo'lsa, xato xabari paydo bo'ladi.
ofNoTestFileCreate	Faylni saqlashda tarmoqdagi himoyalangan fayllarni va mavjud bo'lmagan disklarni tanlashni oldini oladi.

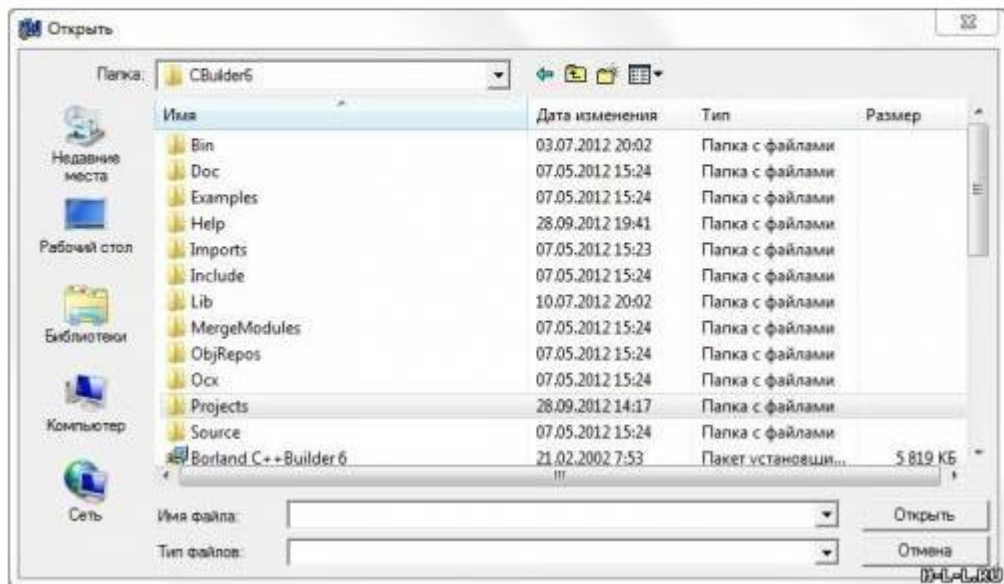
ofNoValidate	Xatlanmaydigan belgilar fayl nomlariga yozilishiga ruxsat bermaydi, ammo siz hal qilinmagan belgilar bilan fayllarni tanlashingizga to'sqinlik qilmaydi
ofOldStyleDialog	Eski uslubda fayl tanlovi dialogini hosil qiladi.
ofOverwritePrompt	Agar faylni saqlashda foydalanuvchi mavjud fayl nomini yozgan bo'lsa, unda o'sha faylga ega bo'lgan fayl bor va foydalanuvchi mavjud faylni qayta yozish istagi so'raladi.
ofPathMustExist	Agar foydalanuvchi fayl nomidagi mavjud bo'lmagan katalogni ko'rsatgan bo'lsa, xato xabari yaratadi.
ofReadOnly	Odatiy bo'lib, dialog oynasi ochilganda Read-Only indikatorini ochiladi
ofShareAware	Kollektiv kirish shartlarining buzilishidagi xatolar inobatga olinmaydi va ularga qaramasdan faylni tanlashga imkon beradi.
ofShowHelp	Muloqot oynasidagi Yordam tugmasini ko'rsatadi.

Windows operatsiya tizimida fayllarni qidirish uchun ularni saqlash va ochishning universal dialogli darchalari ko'zda tutilgan bo'lib, ulardan foyladanishda Dialog qo'shimcha ilovadan tegishli komponentalarni shaklda joylashtirish kerak.



12-rasm.

Bu komponentalarni shaklda joylashtirgach, fayllar bilan ishlashning standart dialogli darchalarini chaqirib olish mumkin.

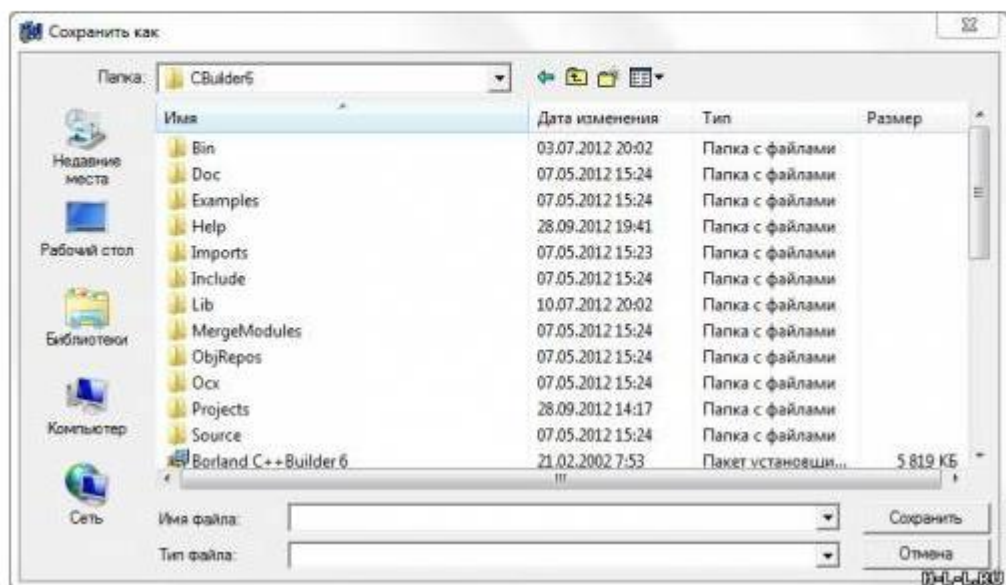


13-rasm.

```

if(OpenDialog1->Execute())
{
.....
}

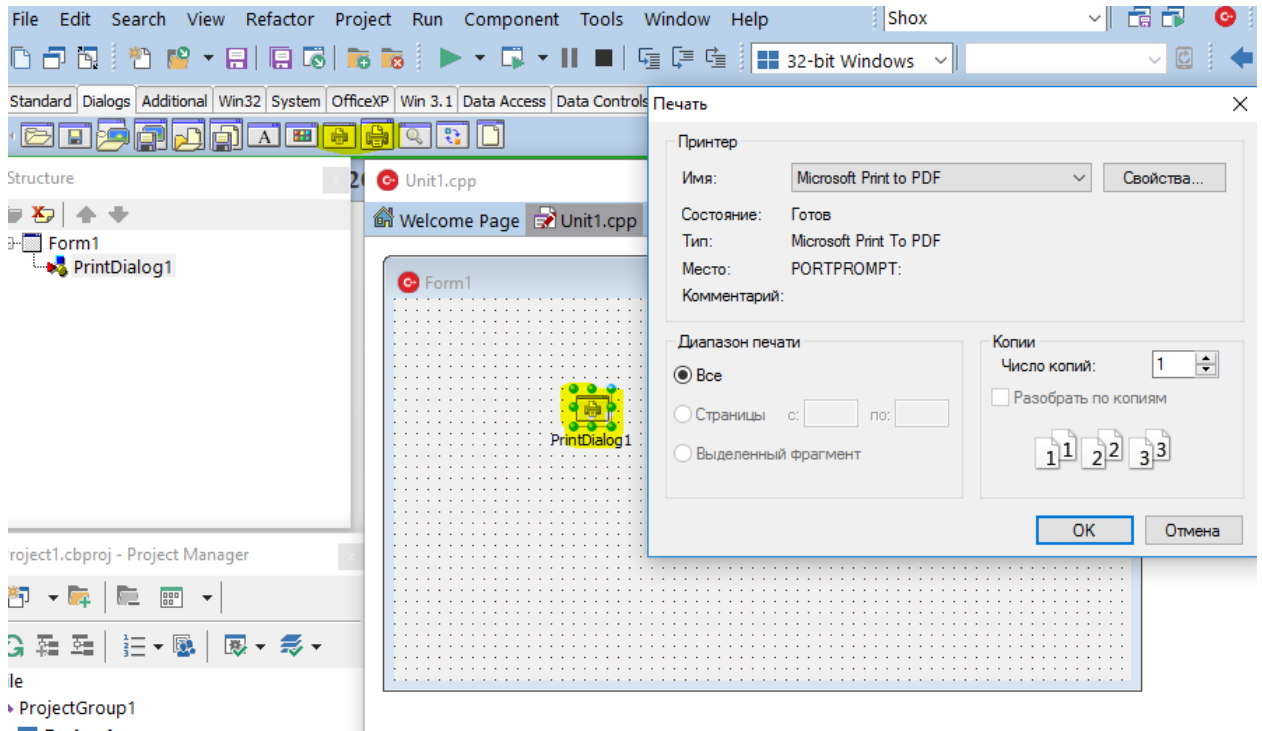
```



14-rasm.

```
if( SaveDialog1->Execute() )  
{  
.....  
}
```

TprintDialog komponentasi chop etish jarayonini boshqaruvchi asosiy komponentimiz hisoblanadi.



15-rasm

```
if( PrintDialog1->Execute() )  
{  
.....  
}
```

II bob. Borland C++ Builder dasturlash muhitida chop etish jarayonini boshqaruvchi dasturini yaratish texnologiyasi.

Bitiruv malakaviy ishda tayyorlangan dastur loihadini tuzushda Uorner metodologiyasidan foydalanildi.

Masalaning qo'yilishi: Programmalarni loyihalashtirish va ishlab chiqishning birinchi bosqichi masalaning qo'yilishini ishlab chiqishdir. Kompyuter programmalari uchun masalani qo'yilishini ishlab chiqishda quyidagi ishlar bajarilishi lozim:

1. Muammoni yechish yoki maqsadga erishish uchun zarur bo'lgan talablarni (xususiyatlar, sifat va imkoniyatlar) ishlab chiqish;
2. O'z ichiga quyidagilarni oluvchi spetsifikatsiyalarni ishlab chiqish:
 - programmaning maqsadi;
 - sistema funktsiyalarini tavsiflash;
 - kiritiladigan va chiqariladigan ma'lumotlarning spetsifikatsiyalari;
 - nazorat talablari (test holatlarini belgilash);
 - hujjatlar tipi va soni.

Bu jarayonda, sistema tayyor holda ega bo'lishi lozim bo'lgan xususiyatlar aniqlanadi, sistemaning funktsiyalari va interfeysning xarakteristikalarini tavsiflanadi.

Masalani yechishni boshlashdan avval uni aniq ta'riflash lozim. Birinchi navbatda - bu boshlang'ich va natijaviy ma'lumotlarni aniqlash hamda nima berilgan; nimani aniqlash lozim, degan savollarga javob topishni anglatadi.

Masalani qo'yilishini yanada mufassalroq ko'rib chiqishda quyidagi savollarga javob berish lozim:

- **yechimni qanday aniqlash kerak?**

Krill va lotin alifbosi orasidagi munosabatlarni to'g'ri amalga oshiruvchi dasturning yaratilishi uchun va yechimning aniqlanishi uchun dastlab shu turga mansub bo'lgan dasturlar va ulardagi kamchiliklar o'rganilishi kerak.

- **qanday ma'lumotlar yetishmayapti va ularning hammasi lozimmi;**

yaratilgan dasturiy vosita yangicha dasturlarga (mavjud dasturlarning yangi avlodlariga) mutunosibligiga etibor berish zarur.

Programmani loyihalashtirish: Dastavval, programm sistemaning arxitekturasi loyihalashtiriladi. Bu loyihalashtirishning boshlang'ich (umumiy) bosqichini ko'zda tutadi va sistema strukturasiga qo'yiladigan talablarni sinchiklab o'rganish bilan tugallanadi. Odatda, modullar pog'onasida har bir modulga talablar majmui ishlab chiqiladi:

- modulga nom beriladi, hamda formal parametrli modullar funktsiyalari haqida taklif kiritiladi;
- modul bajaradigan ishlar obzori beriladi;

- qaysi modullar shu modulga murojaat qiladi va mazkur modul qaysi modullarga murojaat qiladi;
 - formal va haqiqiy parametrlar, global, lokal bir-biri bilan bog'liq bo'lgan (bir qancha modul uchun umumiy bo'lgan) o'zgaruvchilar;
 - moduldagi ko'rsatilishi foydali bo'lgan kommentariylar.

Keyingi bajaradigan ishimiz – sinchiklab loyihalashtirishdir. Bu bosqichda programma protseduralar bo'yicha tavsiflanadi, hamda har bir modulni amalga oshirish uchun algoritmi tanlash va baholash ishlari bajariladi. Loyihalashtirish uchun sistemaga bo'lgan talablar asos qilib olinadi.

Programmarni loyihalashtirishning turli metodlari mavjud. Zamonaviy metodlar dekompozitsiyaga asoslangan. Dekompozitsiya esa o'z navbatida abstraktsiyalarga asoslanadi. Dekompozitsiyaning maqsadi – aniq, oddiy qoidalarga asoslanib o'zaro munosabatda bo'ladigan modullarni yaratishdan iboratdir. Dekompozitsiya, programmani, keyin birlashtirilishi mumkin bo'lgan komponentlarga bo'lish uchun ishlatiladi.

Arxitekturani loyihalash metodlari ikki guruhga bo'linadi:

- ishlov berishga mo'ljallangan;
- ma'lumotlarga mo'ljallangan.

Ishlov berishga mo'ljallangan metodlar o'z ichiga quyidagilarni oladi:

- a) modulli programmalash;
- b) funktsional dekompozitsiya;
- v) ma'lumotlar oqimidan foydalanib loyihalash.
- g) loyihani strukturaviy tahlili texnologiyasi

Modulli programmalash. Asosiy kontseptsiyalari:

- har bir modul yagona mustaqil funktsiyani amalga oshiradi;
- yagona kirish chiqish nuqtasiga ega;
- modulning razmeri iloji boricha kichiklashtiriladi;
- har bir modul boshqa modullardan mustaqil holda ishlab chiqiladi;

Mazkur tamoyilga asoslansak, har bir modul alohida testlanadi, so'ngra kodlashtirilib va testlangandan so'ng ular birlashtiriladi va butun sistema testlanadi.

Funktsional dekompozitsiya. Har bir modul loyihachining sub'ektiv qarori bilan xarakterlanadi. Aloqa, yaxshi tashkil etilgan interfeyslar yordamida amalga oshiriladi.

Ma'lumotlar oqimidan foydalanib loyihalash. Ma'lumotlar oqimidan programmani loyihalashning asosi kabi foydalaniladi.

Yuqoridan pastga qadamma-qadam detallashtirish strukturali loyihalash elementlaridan iborat:

- ma'lumotlar oqimini ekspertiza qilish va ma'lumotlar oqimi grafigida aks ettirish;
- ma'lumotlar oqimining kiritiladigan markaziy va chiqariladigan elementini tahlil qilish;
- programmaning iyerarxik strukturasi tuzish;
- programma strukturasi detallashtirish va optimallashtirish.

Loyihani strukturaviy tahlili texnologiyasi. Bu, sistemaning ob'ektlari o'rtasidagi iyerarxik funktsional aloqalarni tuzishda maxsus grafik vositadan foydalanib, strukturaviy tahlil qilishga asoslangan. Ushbu usul sistemani yaratishning boshlang'ich bosqichlarida samarali hisoblanadi, chunki diagrammalar ham oddiy, ham tushunarli bo'ladi.

Ma'lumotlar strukturasiidan foydalanib loyihalash metodlari quyidagilardir:

- a) Jekson metodologiyasi;
- b) Uorner metodologiyasi;
- v) iyerarxik diagrammalar metodi;
- g) loyihalashning ob'ektga mo'ljallangan metodologiyasi.

Uorner metodologiyasi. Bu metodologiya ham oldingisiga o'xshash, lekin loyihalash ishlari ko'proq detallashtirilgan. Loyihani aks ettirishning quyidagi turlaridan foydalaniladi:

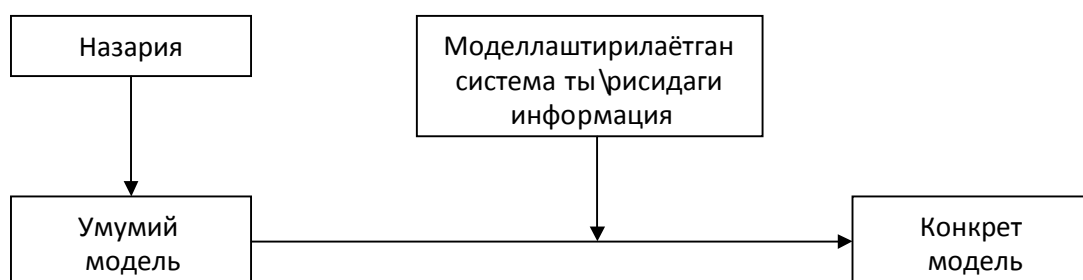
- ma'lumotlarni tashkil qilish diagrammalari (kiritiladigan va chiqariladigan ma'lumotlar);
- mantiqiy davom etish diagrammalari;
- ta'limotnomalar ro'yxati (loyihada ishlatiladigan komandalar);
- psevdokodlar (loyihaning tavsifi);
- sistemaning kiritiladigan ma'lumotlarini aniqlash;
- kiritiladigan ma'lumotlarni iyerarxik strukturasi tuzish;
- kiritiladigan faylning elementlari formatlarini batafsil ko'rib chiqish;
- chiqariladigan ma'lumotlar uchun ham xuddi shu ish bajariladi;
- programmaning spetsifikatsiyasi: o'qish, tarmoqlanish, hisoblash, chiqishlar, ostprogrammalarini chaqirish;

ta'limotnomalarni mantiqiy ketma-ketlikni ko'rsatuvchi blok-sxema tipidagi diagrammalar tuzish. **Modelni tuzish** Ko'pchilik hollarda modelni tuzish, murakkab masala hisoblanadi. Modellashtirishda tajriba orttirish uchun iloji boricha ko'p, ma'lum (mashhur) va muvaffaqiyatli chiqqan modellarni o'rganish lozim.

Modellarni tuzishda, odatda, ikkita tamoyildan foydalaniladi:

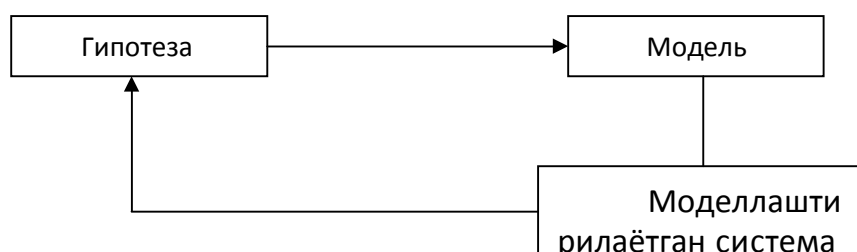
- deduktiv (umumiydan xususiyga);
- induktiv (xususiydan umumiyga).

Deduktiv usulda ma'lum fundamental modelning xususiy holi ko'rib chiqiladi (16-rasm). Bu yerda, berilgan farazlar asosida ma'lum model modellashtirilayotgan ob'ekt shart-sharoitlariga moslashtiriladi.



16-rasm. Deduktiv usulda modelni tuzish sxemasi.

Induktiv usulda (17-rasm) gipotezani taklif qilish, murakkab ob'ektni dekompozitsiya, tahlil, so'ngra sintez qilishdan iboratdir. Bunda sistema xattiharakatlaridagi qonuniyatlarni aniqlash maqsadida modellashtirish va xulosa chiqarish usullaridan keng foydalaniladi. Gipotezani oydinlashtirib olish



17-rasm. Induktiv usulda modelni tuzish sxemasi.

Induktiv usulda modelni tuzish texnologiyasi:

- 1) empirik, ya'ni tajribaga asoslangan bosqich:
 - a) xulosa chiqarish;
 - b) intuitsiya;
 - c) faraz qilish;
 - d) gipoteza.
- 2) modellashtirish uchun masalaning qo'yilishi;
- 3) baholashlar (miqdoriy va sifat tavsifi);
- 4) modelni tuzish.

Algoritmni ishlab chiqish. Algoritmni ishlab chiqish – eng murakkab va ko'p mehnat talab etuvchi, lekin ijodiy nuqtai nazardan qaraganda eng qiziqarli jarayondir.

Ishlab chiqish metodini tanlash, masalaning qo'yilishi va uning modeliga bog'liq. Ushbu bosqichda algoritmning to'g'riligini tekshirish lozim. Bu esa oson emas, ko'p mehnat talab etadi. Algoritm to'g'riligini isbotlashning keng tarqalgan protsedurasi – bu turli testlar majmuidan uni o'tkazishdir. Ammo bular ham programmaning to'g'ri ishlashiga kafolat bera olmaydi. Algoritm to'g'riligini isbotlashning umumiy metodikasiga asosan algoritm, qadamlar ketma-ketligi ko'rinishida tavsiflanadi. Har bir qadam uchun hamma kiritilishi (ushbu qadamgacha bo'lgan shartlar asosida) va chiqarilishi (ushbu qadamdan keyingi shartlar hisobga olingan holda) lozim bo'lgan ma'lumotlar asosida uning to'g'riligini isbotlovchi bironta isbot taklif etiladi. So'ngra yakuniy kiritiladigan va chiqariladigan ma'lumotlar bilan algoritmning tugallanganligi isboti taklif etiladi.

Algoritmni va uning murakkabligini tahlil qilish. Bu bosqich foydalaniladigan kompyuter resurslarini aniq ma'lumotlarga ishlov berish vaqtini baholash, hamda mahalliy tarmoq va telekommunikatsiyalarda ishlashga moslashish uchun zarurdir. Bundan tashqari ushbu masala uchun bironta miqdoriy kriteriy bo'lsa maqsadga muvofiq bo'ladi. Chunki bir necha algoritmlar solishtirilib ulardan eng oddiy va samaralisini tanlash mumkin bo'ladi. Programmadan foydalanishdan avval, u, sozlash va testlash bosqichidan o'tishi lozim.

Programmislarning eng muhim qoidasi – Har bir tayyor programm mahsulot uchun yordam (help) va matnli fayl (readme.txt) ko'rinishida hujjatlashtirilgan kuzatuv zarur.

2.1 Chop etish jarayonini boshqaruvchi dasturning algoritmlari.

Chopetish qurilmalari bilan ishlash hozirgi kun ish jarayonining ajralmas qismidir. Chopetish jarayonini boshqaruvchi dasturlarning qo'yilgan maqsadga mos ishlashi esa juda zarurdir. Chopetish jarayonini boshqaruvchi dasturning ishlab chiqilishi uchun bizga mavjud dasturlarning xatolarini to'g'irleydigan hamda kamchiliklarini to'ldiradigan algoritim zarur. Bitiruv malakaviy ish yozilishi davomida yaratilgan dastur huddi shu hususiyatlarni o'z ichiga oladi.

- Dastur foydalanuvchini o'z zimmasiga yuklatilgan murakkab hisoblashlardan ozod qiladi.
- Foydalanuvchi uchun dastlab tushunarsiz bo'lgan xabarlar chopetilmaydi.
- Foydalanuvchi bir dasturdan varaqlar indexini ko'chirib olib boshqa dasturning chopetish oynasiga joylashtirmaydi.
- Foydalanuvchi uchun ogohlantiruvchi habarlar yani "4 ga karrali son kiriting", "hujjatingizga 3 ta bo'sht list qo'shing" kabi habarlar o'rniga dastur bu vazifalarni o'z bajaradi.

Bitiruv malakaviy ish yozilishi davomida yaratilgan dasturning algoritmi quyidagicha. Dastur algoritimini tushunish uchun Codeblocks ilovasida lohihaning qismini keltiramiz:

```
int a;
cout << "a=" ; cin >> a;
if(a%4 == 0){
int b[a];
for(int i=1; i<=a; i++) b[i]=i; //1 2 3 4 5 6 7 8 9 10 11 12
for(int i=1; i<=a/2; i+=2){
    if(i!=a/2-1) cout << b[a-i+1] <<","<< b[i]<<",";
    else cout << b[a-i+1] <<","<< b[i];} //12,1,10,3,8,5
    cout << endl;
for(int i=1; i<=a/2; i+=2){
    if(i!=a/2-1) cout << b[a/2-i+1] <<","<< b[a/2+i]<<",";
    else cout << b[a/2-i+1] <<","<< b[a/2+i];} //6,7,4,9,2,11
}
```

Dastur C++ Builder (XE8) dastruri muhitida yoziladi, hamda alohida dsatur sifatida o'rnatiladi, tashqi interfeysi optimal holatda bo'lib ortiqcha,etiborni jalb qiladigan va chalg'itishi mumkin bo'lgan yozuvlar va shakllardan holi. Dastur asosan ko'rsatilgan hujjatni "kitobcha" ko'rinishida chopetishga mo'ljallangan bo'lib dastlab foydalanuvchi hujjatni aktiv holga olib kelishi (ochishi) yoki dastur yordamida hujjat manzilini ko'rsatishi kifoya. Chopetish uchun tanlangan hujjat aktivlashgandan(ochilgandan) so'ng dastur yodamida

hujjatning varaqlar soni hisoblanadi va yetishmagan varaqlarni o'zi avtomatik qo'shadi. Keyingi bosqichda chop etish jarayoni boshlanadi va kitobchani faqat bir tomoni chop etiladi. Chopetilgan varoqlar chopetish qurilmasi ga qaytadan qo'yilgach navbatdagi boshqichga o'tiladi. Chop etishgan varoqlarni malum qismidan buklagan holda kitobcha holiga keltiriladi.

Chopetish jarayonini avtomatlashtirishda C++ builder dasturida tuzulgan iliva orqali Word dasturidagi macros larni boshqarish orqali amalga oshiriladi. Bitiruv malakaviy ishning 2.2 bo'limida makroslardan foydalanish va makros kodi ham keltiriladi.

2.2 Borland C++ Builder dasturlash muhitida chop etish jarayonini boshqaruvchi dastur kod qismi

Bitiruv malakaviy ishni yozish jarayonida C++ Builder dasturi muhitida chopetish jarayonini boshqaruvchi dastur yartildi quyida uning kod qismini keltiramiz.

```
#include <vcl.h>
#pragma hdrstop
#include <string.h>

#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma link "Word_XP_srvr"
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    if(OpenDialog1->Execute())
        Edit1->Text = OpenDialog1->FileName;
        String str_errors;
        String str=OpenDialog1->FileName;
        Variant VarApp,VarDoc;

    try
    {
        VarApp=CreateOleObject("Word.Application");
```

```

VarDoc=VarApp.OlePropertyGet("Documents");
VarDoc.OleProcedure("Open", WideString(str));
VarApp.OlePropertySet("Visible",true);
}
catch(Exception &ex)
{
str_errors=ex.Message;
ShowMessage(str_errors);
}}
//-----

void __fastcall TForm1::Button2Click(TObject *Sender)
{
WideString macr;
macr="BIR";
WordApplication1->Run(macr.c_bstr());
}
//-----

void __fastcall TForm1::Button3Click(TObject *Sender)
{
WideString macr;
macr="IKKI";
WordApplication1->Run(macr.c_bstr());
}
//-----

```

Dastur yaratilishi jarayonida qoʻshimcha sifatida Word dasturida Normal.dotm shablonidan ham foydalanildi, quyida foydalanilgan makroslar kodini keltiramiz.

Sub BIR()

Dim x As Integer, y As Integer

Dim a As Integer

Dim i As Integer

ActiveDocument.Bookmarks("\EndOfDoc").Select

x = ActiveDocument.Range.Information(wdNumberOfPagesInDocument)

' MsgBox x

Do While Not (x Mod 4 = 0)

Selection.InsertBreak Type:=wdPageBreak

x = x + 1

Loop

a = x

Dim h As String

Dim BIR(1 To 1000) As Integer

For i = 1 To a

BIR(i) = i

Next

' 1 2 3 4 5 6 7 8 9 10 11 12

' 12,1,10,3,8,5

For i = 1 To a/2

h = h & CStr(BIR(a - i + 1)) & ","

If (BIR(i) = a/2 - 1) Then

h = h & CStr(BIR(i))

Else

h = h & CStr(BIR(i)) & ","

End If

i = i + 1

Next

```
Application.PrintOut FileName:="", Range:=wdPrintRangeOfPages, Item:= _  
    wdPrintDocumentWithMarkup, Copies:=1, Pages:=h, PageType:= _  
    wdPrintAllPages, Collate:=True, Background:=True, PrintToFile:=False, _  
    PrintZoomColumn:=2, PrintZoomRow:=1, PrintZoomPaperWidth:=0, _  
    PrintZoomPaperHeight:=0
```

End Sub

Sub IKKI()

'Ikki

Dim x As Integer, y As Integer

Dim a As Integer

Dim i As Integer

ActiveDocument.Bookmarks("\EndOfDoc").Select

x = ActiveDocument.Range.Information(wdNumberOfPagesInDocument)

Do While Not (x Mod 4 = 0)

Selection.InsertBreak Type:=wdPageBreak

x = x + 1

Loop

' MsgBox x

a = x

Dim h As String

Dim BIR(1 To 1000) As Integer

For i = 1 To a

BIR(i) = i

Next

' 1 2 3 4 5 6 7 8 9 10 11 12

' 6,7,4,9,2,11

For i = 1 To a/2

h = h & CStr(BIR(a/2 - i + 1)) & ", "

If (BIR(a/2 + i) = a - 1) Then

h = h & CStr(BIR(a/2 + i))

Else

h = h & CStr(BIR(a/2 + i)) & ", "

End If

i = i + 1

Next

Application.PrintOut FileName:="", Range:=wdPrintRangeOfPages, Item:= _

wdPrintDocumentWithMarkup, Copies:=1, Pages:=h, PageType:= _

wdPrintAllPages, Collate:=True, Background:=True, PrintToFile:=False, _

PrintZoomColumn:=2, PrintZoomRow:=1, PrintZoomPaperWidth:=0, _

PrintZoomPaperHeight:=0

End Sub

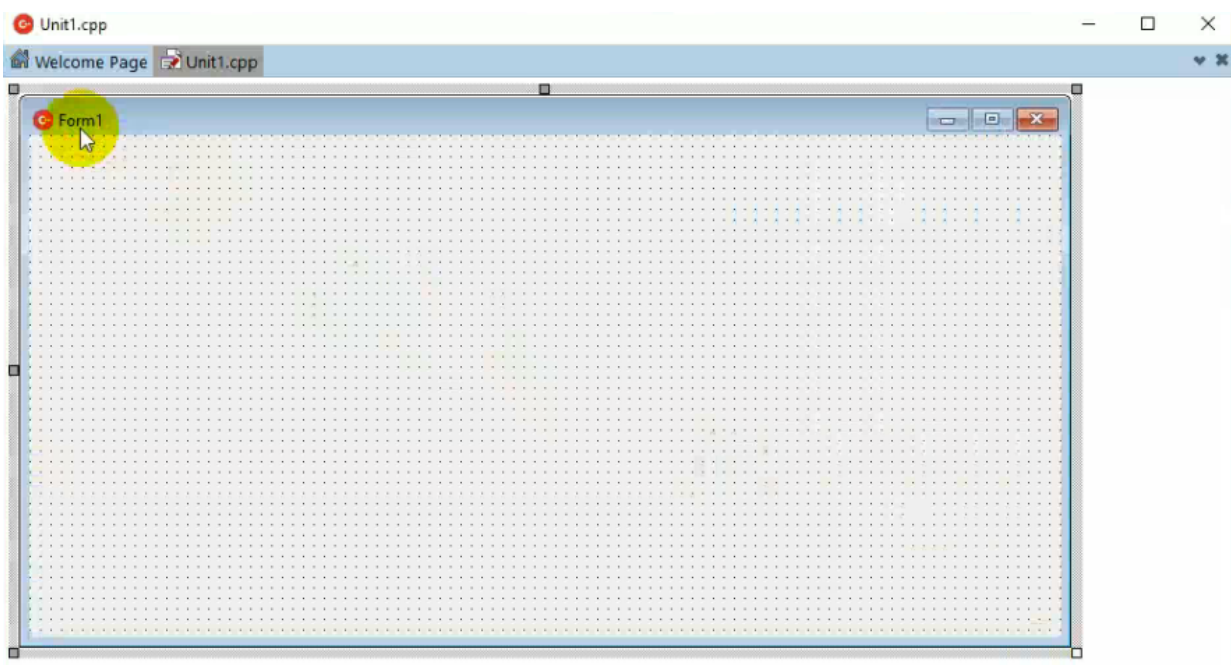
2.3 Dasturiy taminot interfeysini ishlab chiqish va amaliyotda qo'llash sohalari.

“Kitobcha” holatida chopetish jarayonini avtomatlashtirish uchun ishlab chiqilgan dastur interfeysi yaratilishini rasmlar orqali ko'rsatib o'tamiz.

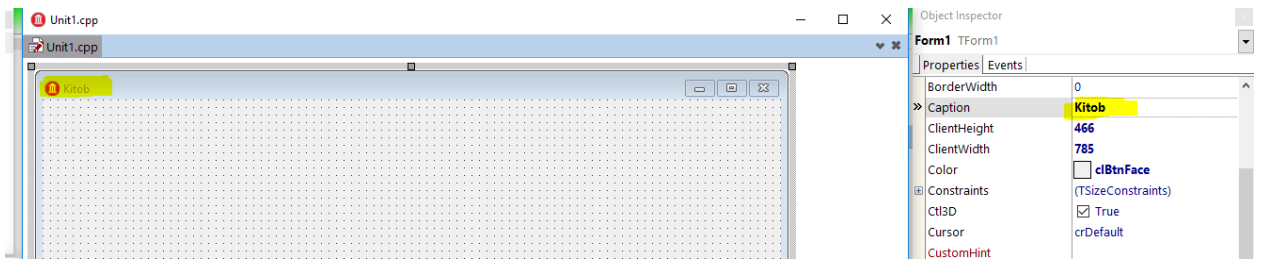
Dastlab C++ Builder dasturini yuklab forma hosil qilamiz.



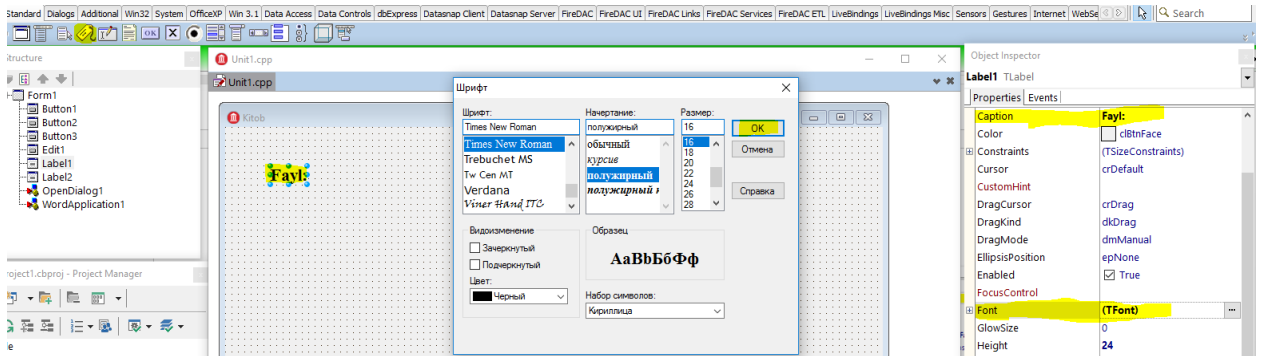
18-rasm. Yani forma hosil qilish.



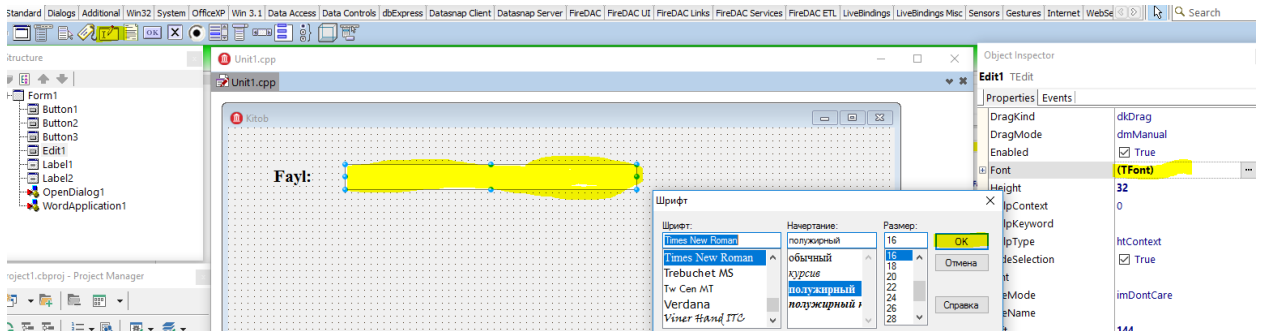
19-rasm.



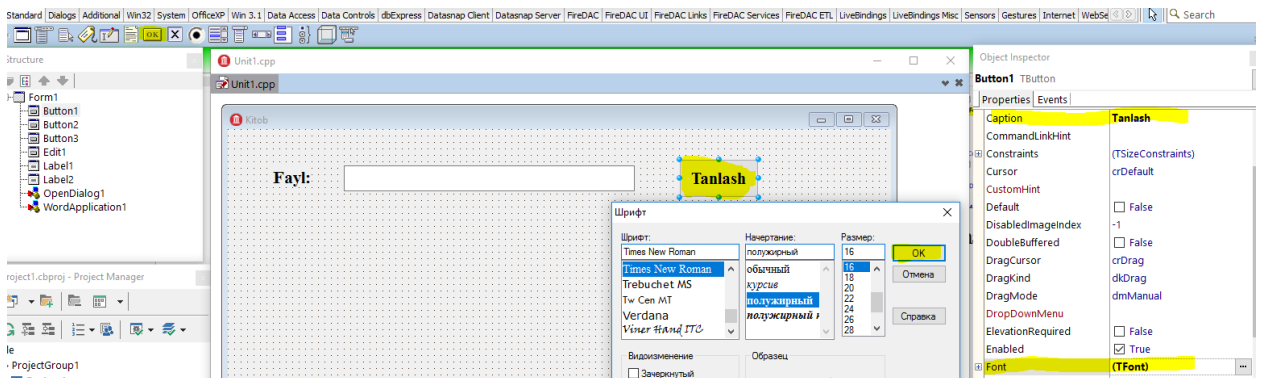
20-rasm. Forma nomini o'zgartiramiz.



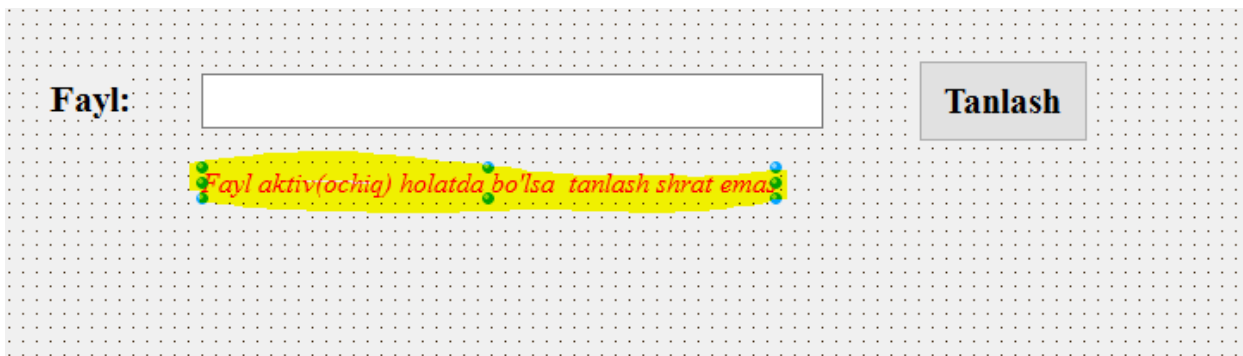
21-rasm. Label komponentasini formaga joylab uning nomini va o'lchamlarini o'zgartiramiz.



22-rasm. Edit komponentasini formaga joylashtirib uning text hususiyatini va o'lchamlarini o'zgartiramiz.



23-rasm. Button komponentasini formaga joylab, uning Caption xususiyatini va o'lchamlarini to'g'irlab olamiz.



24-rasm. Yanabir Label komponentasi orqali foydalanuvchilarga ogohlantiruvchi habar qoldiramiz.



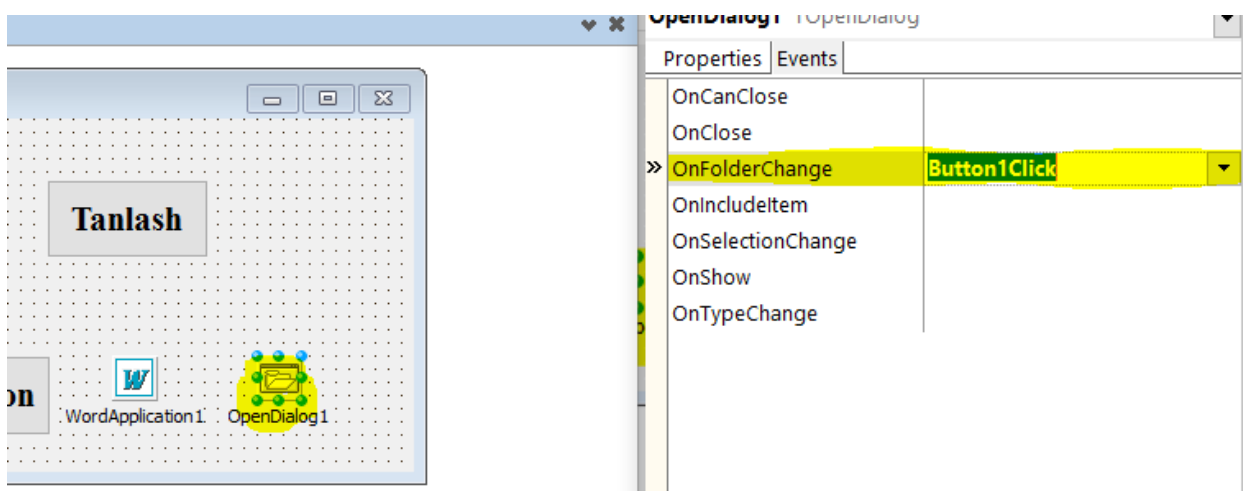
25-rasm. Chopetishni boshlovchi tugmachalarni o'rnatamiz.



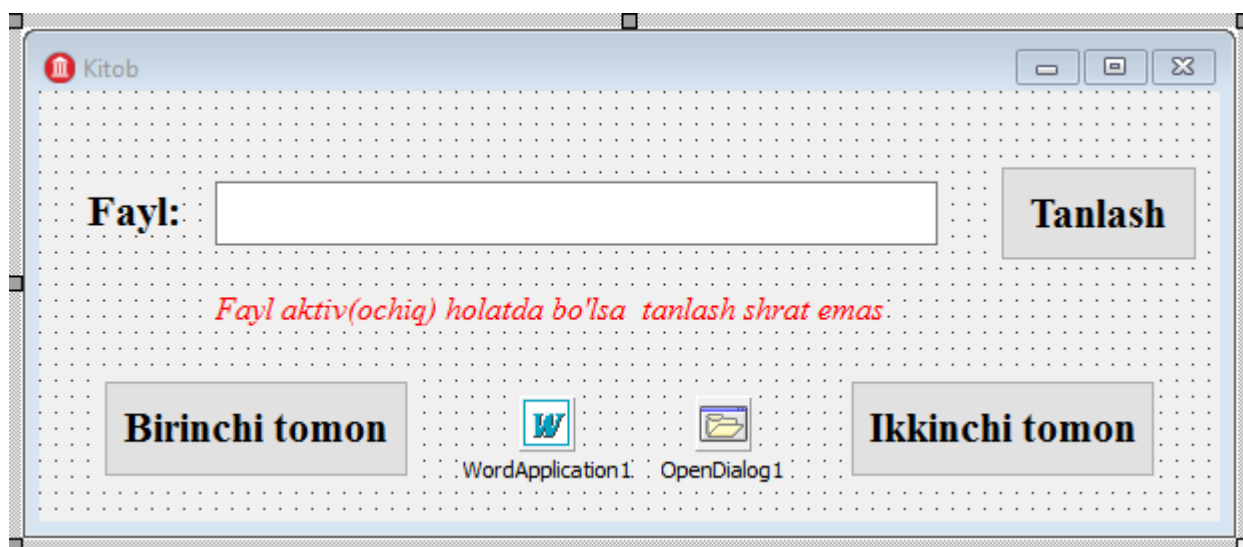
26-rasm. WordApplication komponentasini formaga joylashtirish orqali dasturning Word dasturi bilan aloqasini o'rnatamiz.



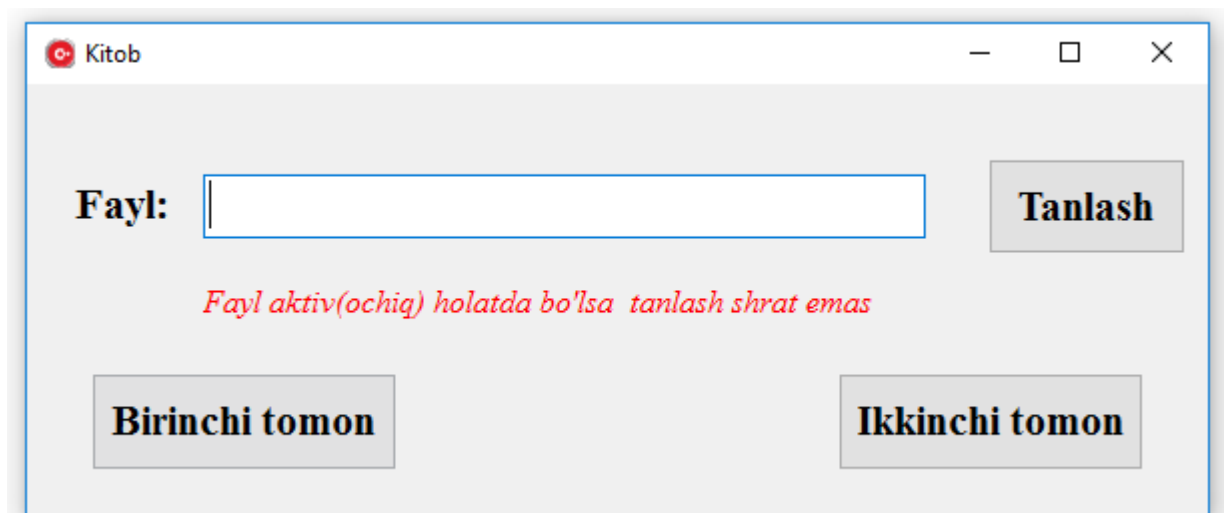
27-rasm. OpenFileDialog komponentasini formaga joylaymiz.



28-rasm. OpenFileDialog komponentasining hodisalaridan OnFolderChange hodisasiga Buutton1Clickni ko'rsatamiz. Bubilana biz "Tanlash tugmasi bosilganda" fayllar bilan ishlash uchun muloqot oynasi ochilishiga erishamiz.



29-rasm. Dasturning yakuniy interfeysi.



30-rasm. Dasturning ishga tushgan holati.

XULOSA

Hozirgi kunda axborot texnologiyalar hamda offis jihozlaridan foydalanmaydigan tashkilotlarning mavjud emasligini hisobga olgan holda offis jihozlaridan foydalanish hamda mehnat samaradorligini oshirish uchun amaliy dasturlarning o'zni beqiyosdir. Tashkilot hodimi mehnatini yengillashtiruvchi amaliy dasturlar sirasiga hujjatlar bilan ishlovchi amaliy dasturlar ham kiradi.

Shu bois biz bitiruv malakaviy ishimizda ishchi hodimlar va talabalar ko'p duch keladigan muommoli vaziyatlardan biri bo'lgan hujjatlarni "kitobcha" shaklida chopetishni avtomatlashtirish jarayonini ishlab chiqishni maqsad qilib qo'ygan edik. Ushbu bitiruv malakaviy ishni bajarish jarayonida

- Ta'lim muassasalarining (umumiy o'rta ta'lim maktablari, kasb-hunar kollejlari, akademik litseylar) o'quv jarayonida amaliy dasturiy vositalardan foydalanish holati o'rganilib, tahlil qilindi;

- Amaliy dasturiy vositalarning didaktik imkoniyatlari o'rganib chiqildi;

- Mavjud amaliy dasturiy vositalar o'rganilib, tahlil qilindi;

- Amaliy dasturiy vositalarni yaratishda foydalanuvchilarning psixofiziologik xususiyatlari inobatga olinishi o'rganildi;

- Amaliy dasturiy vositalar yaratish uchun mo'ljallangan dasturiy ta'minotlar o'rganildi va tavsifi keltirildi;

- Ishlab chiqilgan dasturiy vosital umumiy o'rta ta'lim maktablarida (2017-2018 o'quv yili davomida Termiz shahar **Qaysi maktabga amaliyotga brogan bo'lsangiz shuni yozing**) sinovdan o'tkazilib, o'quv jarayoni uchun quyidagi samaradorlik berishi aniqlandi:

- o'quv materialini o'rganish uchun vaqtni tejash;

- o'quvchiga mustaqil ishlash uchun imkoniyat yaratish;

- yo'l qo'yilgan xatolar orqali ortiqcha isrofgarchiliklarning oldini olish;

Ammo shu bilan bir qatorda bu sohada o'z yechimini kutayotgan bir qator muammolar mavjudligi ham aniqlandi. Amaliy dasturiy vositalarni yaratish texnologiyasini egallab olgan mutaxassislarning yetishmasligi, ular jumlasiga mutaxassis bo'lmagan o'qituvchilarning dasturchilar, texnik mutaxassislarsiz amaliy dasturiy vositalarni yaratishlari, muammolarga duch kelishlari, vaqtning ko'proq talab etilishi, dasturiy ta'minot va texnika vositalarini sozlash va foydalanishda vujudga keladigan muammolar, axborotlarni kompyuter ekranidan o'qish jarayonidagi qiyinchiliklar va boshqa jihatlar kiradi.

Foydalanilgan adabiyotlar

1. Закон Республики Узбекистан «Об образовании» // Ведомости Олий Мажлиса Республики Узбекистан. № 9, ст. 225, 1997 г.
2. O'zbekiston Respublikasi “Kadrlar tayyorlash milliy dasturi” to'g'risidagi qonuni.//O'zbekiston Respublikasi Oliy majlisining IX sessiyasi materiallari. – Toshkent, 29 avgust, 1997 y.
3. O'zbekiston Respublikasining “Axborotlashtirish to'g'risidagi” Qonuni.// O'zbekiston Respublikasi Oliy majlisining materiallari. –Toshkent, 11 dekabr, 2003 y.
4. O'zbekiston Respublikasi Vazirlar Mahkamasining 2001 yil 23 maydagi «2001-2005 yillarda kompyuter va axborot texnologiyalarini rivojlantirish, «Internet»ning xalqaro axborot tizimlariga keng kirib borishini ta'minlash dasturini ishlab chiqishni tashkil etish chora-tadbirlari to'g'risida»gi 230-sonli qarori.// – Toshkent. “Xalq so'zi”. 2001 yil, 24 may. № 101 (2663).
5. O'zbekiston Respublikasi Prezidentining 2002 yil 30 maydagi «Kompyuterlashtirishni yanada rivojlantirish va axborot-kommunikatsiya texnologiyalarini joriy etish to'g'risida»gi PF-3080-son Farmoni.// –Toshkent. “Xalq so'zi”. 2002 yil, 1 iyun. № 116 (2944).
6. O'zbekiston Respublikasi Vazirlar Mahkamasining 2002 yil 6 iyundagi “Kompyuterlashtirishni yanada rivojlantirish va axborot-kommunikatsiya texnologiyalarini joriy etish to'g'risida”gi 200-sonli Qarori.// –Toshkent “O'zbekiston ovozi”. 2002 yil, 8 iyun. № 72.
7. O'zbekiston Respublikasi Vazirlar Mahkamasining “Pedagog kadrlarni qayta tayyorlash va ularning malakasini oshirish tizimini yanada takomillashtirish to'g'risida”gi 25-sonli Qarori.// –Toshkent. 2006 yil 16 fevral.
8. O'zbekiston Respublikasi Oliy va o'rta maxsus ta'lim vazirligi Hay'atining “Oliy ta'lim tizimida o'quv jarayonini va o'quv uslubiy faoliyatni modernizatsiyalash hamda innovatsion ta'lim texnologiyalarini joriy etish to'g'risida”gi Qarori.// –Toshkent. 2010 yil 17 aprel.
9. O'zbekiston Respublikasi Prezidentining “2011-2015 yillar uchun axborot-kommunikatsion texnologiyalar bazasida axborot-kutubxona va axborot-resurs xizmat ko'rsatishni keyinchalik sifatli rivojlantirish bo'yicha chora-tadbirlar to'g'risida”gi PQ-1487-sonli Qarori.// –Toshkent. 2011 yil 23 fevral.
10. Xalq ta'limi Vazirligining “Umumiy o'rta ta'lim maktablarida faoliyat ko'rsatayotgan o'qituvchilarning zamonaviy axborot-kommunikatsiya texnologiyalaridan ta'lim jarayonida foydalanish bo'yicha malakasini oshirishni tashkil etish to'g'risida”gi 125-sonli buyrug'i.// –Toshkent. 2012-yil 21-aprel.
11. Ахборот-коммуникация технологияларини янада ривожлантиришга оид қўшимча чора – тадбирлар тўғрисида : ЎЗР Президентини 2005 йил 8

июлдаги ПҚ – 117- сон Қарори // Ўзб-н. Респ. Қонун хуж. Тўпл.-2005.-№27.- Б.4-5.

12. Ахборот технологиялари соҳасида кадрлар тайёрлаш тизимини такомиллаштириш тўғрисида : ЎзР Президентининг Қарори : 2005 йил 30 май // Ўзб-тон Респ-каси Қонуни хужжатлари тўплами.-2005.-№22.-Б.4-5. // Халқ сўзи.-2005.-3 июнь. // Маърифат.-2005.-июнь.

13. Б. Страуструп. Язык программирования С++. Специальное издание.-М.:ООО «Бином-Пресс», 2006.-1104 с.

14. Павловская Т.А. С++. Программирование на языке высокого уровня – СПб.: Питер. 2005.- 461 с.

15. Подбельский В.В. Язык СИ++.- М.; Финансы и статистика- 2003 562с.

16. Павловская Т.С. Щупак Ю.С. С/С++. Структурное программирование. Практикум.-СПб.: Питер,2002-240с

17. Павловская Т.С. Щупак Ю.С. С++. Объектно- ориентированное программ-ирование. Практикум.-СПб.: Питер,2005-265с

18. Глушаков С.В., Коваль А.В., Смирнов С.В. Язык программирования С++: Учебный курс.- Харьков: Фолио; М.: ООО «Издательство АСТ», 2001.-500с.

19.Юров В., Хорошенко С. Assembler: Учебный курс- СПб, “Питер”,2000.-672с.

20. 1997 йил 27 июлда “Ахборот тизимлари соҳасини қайта ташкил этиш ва бoshқарishни такомиллаштиришга oid chora-tadbirlar to‘g‘risida” gi PF-1823-sonli farmoni.

21. 2002-йил 30-майда “Компьютерlashtirishni yanada rivojlantirish va axborot kommunikatsiya texnologiyalarini joriy etish to‘g‘risida” gi PF – 3080 sonli farmoni.

22. 2005 йил 2 июнда “Ахборот texnologiyalari sohasida kadrlar tayyorlash tizimini takomillashtirish to‘g‘risida”gi PQ-91-sonli qarori.

23. 2005 йил 8 июлда “Ахборот-kommunikatsiya texnologiyalarini yanada rivojlantirishga oid qo‘shimcha chora-tadbirlar to‘g‘risida”gi PQ-117-sonli qarori.

24. 2005 йил 28 sentyabrda “O‘zbekiston respublikasining jamoat ta’lim axborot tarmog‘ini tashkil etish to‘g‘risida” gi PQ-191-sonli qarori.

25. 2007 йил 3 aprelda “O‘zbekiston Respublikasida axborotni kriptografik muhofaza qilishni tashkil etish chora-tadbirlari to‘g‘risida”gi PQ-614-sonli qarori.

26. 2012 йил 21 martda “Zamonaviy axborot-kommunikatsiya texnologiyalarini yanada joriy etish va rivojlantirish chora-tadbirlari to‘g‘risida”gi PQ-1730-sonli qarori.

27. 2013 йил 20 sentyabrda “Mamlakatimizning dasturiy ta’minot vositalari

ishlab chiquvchilarini rag‘batlantirishni yanada kuchaytirish chora-tadbirlari to‘g‘risida”gi PQ-2042-sonli qarori.

28. 2013 yil 25 sentyabda “Milliy geografik axborot tizimini tashkil etish investitsiya loyihasini amalga oshirish chora-tadbirlari to‘g‘risida”gi PQ-2045-sonli qarorilari.

29. Финогенов К.Г. Основы языка Assemblera.-М.: Радио и связь, 2001. - 288 с.

30. Пильшиков В.Н. Упражнения по языку Паскаль-М.: МГУ, 1986.

31. Абель П. Assembler для IBM PC и программирования. 1991. М.: “Высшая школа”, 1992.- 447 с.

32. Скенлон Л. Персональный ЭВМ IBM PC и XT. Программирование на языке Assemblera. -М.: Радио и связь. 1991.- 336 с.

33. Гофман В. Э., Хомоненко А.Д. Delphi 5. - СПб.: БХВ-Санкт-Петербург, 2000. -800с.

34. Немнюгин С.А. Turbo pascal, учебник. Изд. Питер., 2001, -496 с.

35. Поляков Д.Б., Круглов И.Ю. Программирование в среде Турбо-паскаль. (версия 5.5).М.:МАИ,1992.-576с.

36. Абрамов С.А., Гнезделова Капустина Е.Н.и др. Задачи по программированию. - М.: Наука, 1988.

37. Вирт Н. Алгоритмы + структуры данных = программа.-М.:Мир,1985.- 405с.

38. Информатика. Базовой курс. Учебник для Вузов., Санк-Петербург, 2001. под редакцией С.В.Симоновича.

39. Нортон П. Программно-аппаратная организация IBM PC.-М.:Мир,1991.- 327с.

40. Фигурнов В.Э. IBM PC для пользователя. М.: Финансы и статистика. Юнити. 1997.

41. Юров В. Assembler: практикум. -СПб.: Питер, 2002.- 400с.

42. Informatika va programmalsh. O‘quv qo‘llanma. Mualliflar: A.A.Xaldjigitov, Sh.F.Madraximov, U.E.Adamboev, O‘zMU, 2005 yil.

Internet resurslari ro'yxati:

<http://www.BeyondVBATutoria.com>
<https://www.planetaexcel.ru>
<http://on-line-teaching.com/index.html>
<http://www.xushnudbek.uz>
<http://www.CyberForum.ru>
www.intuit.ru
www.book.ru
www.kniga.ru
www.gov.uz
www.ziyonet.uz
www.ulugov.uz
www.aci.uz
www.ictp.uz
www.terdu.uz
www.infoman.uz
www.olam.uz
www.utube.uz
<http://www.uiniti.msk.ru>
<http://mcit.aris.ru>
<http://www.library.fa.ru>
<http://www.spb.runnet.ru> <http://sunny.ccas.ru/library.html>
<http://www.lrl.da.ru>
<http://www.infotel.ru>
<http://www.worldbank.org>
<http://www.oecdwash.org>
<http://www.syllabus.com/syllsub.html>
www.assist.ru
www.webmoney.ru
<http://sunny.ccas.ru/library.html>
<http://www.mesi.ru>
<http://www.CyberForum.ru>
<http://www.SQL.ru>