

ЎЗБЕКИСТОН РЕСПУБЛИКАСИ ОЛИЙ ВА ЎРТА МАХСУС
ТАЪЛИМ ВАЗИРЛИГИ

ТЕРМИЗ ДАВЛАТ УНИВЕРСИТЕТИ

ФИЗИКА МАТЕМАТИКА ФАКУЛЬТЕТИ

Амалий математика ва информатика кафедраси

Менглиев Ш.А.

“Паскаль дастурлаш тили”
фанидан маърузалар матнлари

Термиз – 2012й.

Ушбу маруза матни учун кўрсатма илмий-услубий кенгашнинг _____ йил
« ____ » _____ бўлиб ўтган ____-сонли мажлисида кўриб чиқилди ва чоп
этишга тавсия этилди.

“Паскаль дастурлаш” фанидан маруза матни / ТерДУ ____ б. Термиз 20_ й.

Такризчилар:

ф.м.-ф.н. Нормуродов Ч. Амалий математика ва информатика кафедраси
кафедра мудири

и.ф.н О. Садатов Амалий математика ва информатика кафедраси доценти

Тузувчи: Амалий математика ва информатика кафедраси ўқитувчиси
Менглиев Ш.А.

МАЪРУЗА № 1

АЛГОРИТМ ВА ДАСТУР ТУШУНЧАЛАРИ

Маъруза режаси:

1. Кириш.
2. Алгоритм ва унинг хоссалари.
3. Алгоритм турлари.
4. Алгоритмик тиллар.
5. Масалаларни ечиш босқичлари.
6. Дастурлаш воситалари
7. Маъруза бўйича саволлар.

А д а б и ё т л а р

1. Арипов М., Ҳайдаров А. Информатика асослари, Ўқув қўлланма, Тошкент, Ўқитувчи” 2002й.
2. Шохамидов Ш., Неъматов А. ва бошқалар. “Информатика” фанидан маърузалар матнлари. ТТЕСИ, Тошкент-2003й.

1. Кириш

Алгоритм сўзи буюк математик Ал-Хоразмийнинг номи билан боғлиқ бўлиб, у биринчи бўлиб араб рақамларидан фойдаланган ҳолда арифметик амалларни бажариш қоидадини баён этди.

Ҳар қандай қўйилган масалани компьютерда ечиш учун олдин унинг ечиш усулини танлаб, кейин унинг алгоритмини ишлаб чиқиш керак бўлади. Демак, ҳеч бир масала йўқки унинг ечилиш йўллари билмасдан ва алгоритмини тасаввур қилмасдан туриб уни компьютерда ечиб бўлмайди.

2. Алгоритм ва унинг хоссалари

Электрон ҳисоблаш машиналарининг вужудга келишига қадар алгоритмга ҳар хил таъриф бериб келинди. Лекин уларнинг барчаси маъно жиҳатдан бир-бирига жуда яқин бўлиб, бу таъриф ҳозирги кунда қуйидагича талқин қилинади.

Таъриф. Алгоритм деб, қўйилган масалани ечиш учун маълум қоидага биноан бажариладиган амалларнинг чекли қадамлар кетма-кетлигига айтилади.

Ҳар қандай алгоритм маълум кўрсатмаларга биноан бажарилади ва бу кўрсатмаларга буйруқ дейилади.

Алгоритм қуйидаги муҳим хоссаларга эга:

Аниқлик ва тушунарлилик - деганда алгоритмда ижрочиға берилаётган кўрсатмалар аниқ мазмунда бўлиши тушунилади.

Оммавийлик - деганда ҳар бир алгоритм мазмунига кўра бир турдаги масалаларнинг барчаси учун ҳам ўринли бўлиши тушунилади.

Натижавийлик - деганда алгоритмда чекли қадамлардан сўнг албатта натижа бўлиши тушунилади.

Дискретлик - деганда алгоритмларни чекли қадамлардан ташкил қилиб бўлаклаш имконияти тушунилади.

3.Алгоритм турлари

Алгоритмнинг учта тури бор: чизиқли, тармоқланувчи ва такрорланувчи.

Чизиқли алгоритм - деб ҳеч қандай шартсиз фақат кетма-кет бажариладиган жараёнларга айтилади.

Тармоқланувчи алгоритм - деб маълум шартларга мувофиқ бажариладиган кўрсатмалардан тузилган алгоритмга айтилади.

Такрорланувчи алгоритм - деб бирон бир шарт текширилиши ёки бирон параметрнинг ҳар хил қийматлари асосида алгоритмда такрорланиш юз берадиган жараёнларга айтилади.

Алгоритмларни турли усулларда тасвирлаш мумкин. Масалан: сўз билан ифодалаш; формулаларда бериш; блок-схемаларда тасвирлаш; дастур шаклида ифодалаш ва бошқалар.

Алгоритмларни блок-схема кўринишда тасвирлаш қулай ва тушунарли бўлгани учун кўп ишлатилади. Бунда алгоритмдаги ҳар бир кўрсатма ўз шаклига эга. Масалан: параллелограмм кўринишдаги белги маълумотларни киритиш ва чиқариш; тўғри тўртбурчак белгиси ҳисоблаш жараёнини; ромб белгиси шартларнинг текширилишини билдиради.

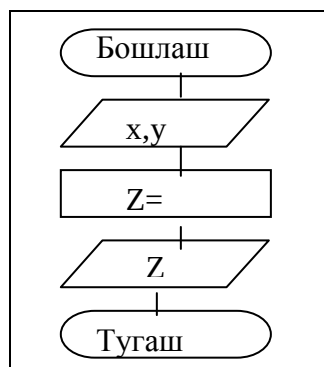
Мисоллар: Чизиқли алгоритмга доир: $Z = \frac{x^2 + \sin(2x - 1)}{\sqrt{x^2 + y^2 + 10}}$ ҳисоблаш

алгоритмини тузинг.

Сўзда берилиши:

- 1.Бошлаш.
- 2.x ва y қийматини киритиш.
- 3.z қийматини ҳисоблаш.
- 4.z қийматини чиқариш.
- 5.Тамом.

Блок-схемада:



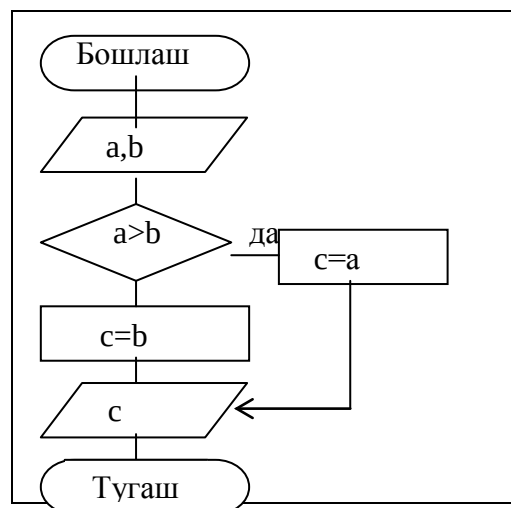
Тармоқланувчи алгоритмга доир:

Иккита a ва b сонлардан каттасини аниқлаш алгоритмини тузинг.

Сўзда берилиши:

Блок-схемада:

1. Бошлаш.
2. a ва b-қийматини киритиш.
3. агар $a > b$ бўлса, натижа a деб олиниб 5га ўтилсин.
4. натижа b деб олинсин.
5. Тамом.



Такрорланувчи алгоритмга доир:

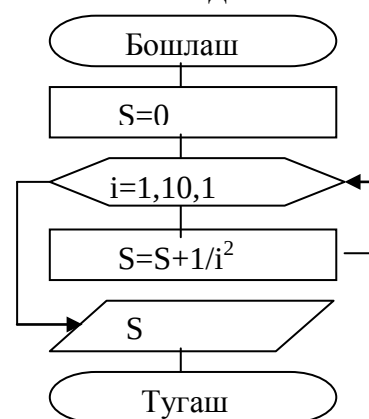
$$S = \sum_{i=1}^{10} \frac{1}{i^2} = 1 + \frac{1}{4} + \frac{1}{9} + \dots + \frac{1}{100}$$

йиғиндини ҳисоблаш алгоритмини тузинг.

Сўзда берилиши:

1. Бошлаш.
2. $S=0$
3. $i=1$
4. $S=1/i^2$
5. Агар $i < 100$ бўлса, у ҳолда $i=i+1$ ва 4 га ўтиш.
Акс ҳолда 6 га.
6. S қиймати чиқарилсин.
7. Тамом

Блок-схемада:



Ҳаётимизда алгоритмларни турли соҳаларда баъзан билган ҳолда баъзан эса билмаган ҳолда ишлатамиз. Алгоритмлар фақат математик характерга эга бўлмасдан уларни оддий ҳаётий турмушимизда ҳам кўп қўллаймиз. Масалан, овқат тайёрлаш, чой дамлаш, бирор берилган ишни бажариш ва бошқа. Бу ишларни бажаришда маълум бўлган аниқ кўрсатмаларни кетма кет бажарамиз. Агар бу кўрсатмалар аниқ бир кетма кетлик тартибида бажарилмаса керакли натижани олаолмаймиз. Мисол тариқасида математик характерга эга бўлмаган бутелброд тайёрлаш алгоритмини кўриб чиқайлик. Бунда бошланғич берилганлар: нон, колбаса ва пишлоқ. Натижа: бутелброд. Бутелброд тайёрлаш алгоритми:

- нон бўлагини кесиб олиш;
- колбаса ва пишлоқ бўлагини кесиб олиш;
- колбаса ва пишлоқ бўлагини нон бўлаги орасига қўйиш.

Агар бу жараённинг кетма кетлик ўринлари алмашса ёки бирор бир босқич амалга оширилмаса натижа бўлмайди.

4.Алгоритмик тиллар

Масалани ечиш алгоритми ишлаб чиқилгандан сўнг дастур тузилади. **Дастур** - бу берилган алгоритмга асосланган бирор бир алгоритмик тилда ёзилган кўрсатмалар, яъни буйруқлар ёки операторлар тўпламидир. **Дастурлаш** - эса бу дастур тузиш жараёни бўлиб, у қуйидаги босқичлардан иборатдир:

- дастурга бўлган талаблар;
- қўйилган масала алгоритмини танлаш ёки ишлаб чиқиш;
- дастур кодларини (матнлари, буйруқларни) ёзиш;
- дастурни тўғрилаш ва тест ўтказиш.

Ҳозирги кунда жуда кўплаб алгоритмик тиллар мавжуд. Уларга дастурлаш тиллари деб атаймиз. Алгоритмик тил - алгоритмларни бир хил ва аниқ ёзиш учун ишлатиладиган белгилашлар ва қоидалар тизимидир. Алгоритмик тил оддий тилга яқин бўлиб у математик белгиларни ўз ичига олади. Тузилган алгоритмларни тўғридан-тўғри машинага бериб бўлмайди, шу сабабли ёзилган алгоритмни бирор бир алгоритмик тилга ўтказиш зарур. Ҳар қандай алгоритмик тил ўз қўлланилиш соҳасига эга. Масалан, муҳандислик ҳисоб ишларини бажаришда Паскал, Бейсик ва Фортран. Иқтисод масалаларини ечишда Паскал ва Кобол. Мантикий дастурлаш учун Пролог ва бошқалар. Ўқув жараёнлари учун Бейсик, Паскал ва бошқалар.

Паскал, Фортран ва Кобол тиллари универсал тиллардан ҳисобланади. Ассемблер тили машина тилига анча яқин тил бўлиб ўрта даражадаги тилдир. Алгоритмик тил инсон тилларига қанча яқин бўлса, у тилга юқори даражали тил дейилади. Машина тили эса энг пастки даражали тилдир.

5.Масалаларни ечиш босқичлари

Компьютерда масалани ечиш босқичлари қуйидагилар:

1.Масалани қўйиш ва унинг математик моделини ишлаб чиқиш. Ушбу босқичда бошланғич маълумотлар таркиби аниқланади. Масалани қўйиш одатда ечиладиган масаланинг асосий хусусиятларини оғзаки тавсифлаш билан тугалланади ва сўнгра, масала математик модели ёзилади.

2.Ечиш усулини танлаш. Масала математик ифодаланиб бўлгандан сўнг уни ечиш усули танланади. Бунда изланаётган натижаларнинг бошланғич маълумотларга боғлиқлигини аниқлайди. Ҳозирги замон ҳисоблаш математикаси фан ва техникага оид турли масалаларни ечиш учун кўплаб сонли усулларга эга.

3.Масалани ечиш алгоритмини ишлаб чиқиш. Бу босқичда компьютерда масалани ечиш учун бажариладиган амаллар (буйруқлар) кетма-кетлиги тавсифланади. Биз буни алгоритмлаш деб атаймиз.

4.Дастурлаш. Бунда алгоритм бирор алгоритмик тилга кўчирилади.

5.Дастурни киритиш ва хатоларини тузатиш.

6.Масалани бевосита компьютерда ечиш ва натижаларни таҳлил қилиш. Бунда дастурда керакли бошланғич қийматлар берилиб компьютерда натижа

олинади ва таҳлил қилинади. Бу босқичда агар керак бўлса алгоритм ва дастур модернизация қилиниши ҳам мумкин. Янги олинган маълумотлар асосида керакли хулосалар ишлаб чиқилади. Бу маълумотлар қўйилган масалани турлича таҳлил қилишга, мураккаб жараёнларни тушунишга, оламшумул янгиликларнинг очилишига, янги назарияларнинг, техника мўъжизаларининг яратилишига хизмат қилади. Умуман олганда «Модел-алгоритм-дастур» учлиги моделлаштиришнинг интеллектуал мағзини ташкил этади, бундай учликсиз компютерда мураккаб масалаларни ечишда муваффақиятга эришиб бўлмайди.

6.Дастурлаш воситалари

Дастурлаш воситаларининг асосий вазифаси дастур тузувчиларга компютерда дастур матнларини киритиш, таҳрирлаш ва ишга тушуриш каби бир қанча ишларни бажаришда қулайлик яратади.

Турбо Паскал дастурлаш воситаси Паскал тилида компютерда дастурлар яратиш учун махсус дастур бўлиб хизмат қилади. Turbo Pascal дастурлаш воситаси ҳам ўз матн муҳаририга эга бўлиб, у кўп томондан Turbo Basic, Turbo Prolog, Turbo C ва бошқа дастурлаш воситаларининг матн муҳаррирларига ўхшаб кетади. Turbo Pascal дастурлаш воситасининг дастурий таъминоти қуйидагилардан ташкил топган:

Turbo.exe -асосий дастур бўлиб тизимни ишга туширади;

Turbo.tpl -дастурлар библиотекаси;

Turbo.hlp -ёрдам чақириш файли;

Graf.tpu -график функция ва процедуралар библиотекаси;

CGA.bgi, EgaVga.bgi -монитор учун график драйверлари;

Tpc.exe -Pas кенгайтмали файлни Exe кенгайтмали файлига ўтказди.

Turbo Pascal ойнаси горизонтал меню ва дастур матнини киритиш майдонидан иборат. Горизонтал меню қуйидагилардан иборат:

File пункти қуйидаги буйруқлар мавжуд:

New -экранни тозалаш ва оператив хотирада янги файл очиш;

Save -оператив хотирадаги дастурни асосий хотирага ёзади;

Save As -бошқа ном билан дастурдан нусха кўчиради;

Open -керакли файлни очишб экранга чақиради;

Exit -тизимдан чиқиш.

Edit -пункти дастур матнини қайта таҳрирлашда ишлатилади.

Run -пункти хотирадаги дастурни бажаришга беради.

Comple -пункти дастурни компилятсия қилиб хотирага ёзади.

Options -пункти тизимнинг параметрларини ўзгартиради.

Функционал тугмачалар вазифалари:

F1 -ёрдам чақириш;

F2 -файлни дискга сақлаш;

F3 -файлни ишга тушуриш;

F5 -экранни очиш ва ёпиш;

F6 -экранни актив ҳолатга ўтказиш;

F9 -дастурни қадам билан

бажариш;

F10 -бош менюга ўтиш.

Матн муҳаририда тугмалар ёрдамида буйруқларни бериш.

Тугмани босиш	Буйруқнинг бажарилиши	Тугмани босиш	Буйруқнинг бажарилиши
Ctrl →	ўнга битта сўз ўтиш	Ctrl N	қатор қўйиш
Ctrl ←	чапга битта сўз ўтиш	Ctrl K B	блок бошини ажратиш
Ctrl W	юқорига силжиш	Ctrl K K	блок охирини ажратиш
Ctrl Z	пастга силжиш	Ctrl K T	сўзни ажратиш;
Home	қатор бошига ўтиш	Ctrl K Y	ажратилган блокни ўчириш
End	қатор охирига ўтиш	Ctrl K S	Блокни қўчириш
Ctrl Home	биринчи қаторга ўтиш	Ctrl U	буйруқни бекор қилиш
Ctrl End	охирги қаторга ўтиш	Ctrl F9	дастурни ишга тушуриш
Del	ўнг ёзувни ўчириш	Ctrl C	дастурни ишдан тўхтатиш
Ctrl Y	қаторни олиб ташлаш		

7.Маъруза бўйича саволлар

- 1.Алгоритм нима ва у қандай хоссаларга эга?
- 2.Алгоритмнинг қандай турлари мавжуд?
- 3.Алгоритмни тасвирлашнинг қандай усуллари биласиз?
- 4.Қандай алгоритмик тилларни биласиз?
- 5.Масалани компьютерда ечиш қандай босқичлардан иборат?
- 6.Математик модел деганда нимани тушинасиз?
- 7.Қандай дастурлаш воситаларини биласиз?
- 7.Турбо Паскал дастури матн муҳаррирининг қандай буйруқларини биласиз?

МАЪРУЗА № 2

ПАСКАЛ АЛГОРИТМИК ТИЛ ЭЛЕМЕНТЛАРИ

Маъруза режаси:

1. Кириш.
2. Паскал алгоритмик тилининг алфавити.
3. Сонлар, ўзгармаслар ва ўзгарувчилар.
4. Ўзгарувчилар ва ўзгармасларнинг дастурда тавсифланиши.
5. Стандарт арифметик ва бошқа функциялар.
6. Арифметик ва мантикий ифодалар.
7. Паскал тилида дастур структураси.
8. Маълумотларни киритиш ва чиқариш операторлари
9. Маъруза бўйича саволлар.

А д а б и ё т л а р

1. Арипов М., Ҳайдаров А. Информатика асослари, Ўқув қўлланма, Тошкент, Ўқитувчи” 2002й.
2. Шохамидов Ш., Неъматов А. ва бошқалар. “Информатика” фанидан маърузалар матнлари. ТТЕСИ, Тошкент-2003й.

1. Кириш

Паскал тили универсал тиллардан бири бўлиб, бошқа тилларга караганда имкониятлари кенгроқ тилдир. Сўнги йилларда Паскал тили жуда такомиллашиб, тобора оммалашиб бормоқда. Pascal тили Швецариялик олим Н.Вирт томонидан яратилиб, кейинчалик Borland корпорацияси тамонидан ривжлантирилди.

2. Паскал алгоритмик тилининг алфавити

Паскал тили ҳам бошқа дастурлаш тиллари каби ўз алфавитига ва белгиларига эга. Паскал тили 26 та латин ҳарфлариин, 0 дан 9 гача бўлган араб рақамларини ва қуйидаги белгиларни ишлатади: бўшлиқ белгиси; 4 та арифметик амаллар +, -, *, / ; мантикий амалларни бажариш учун <, >, <=, >=, <>, = белгиларини ишлатади. Улардан ташқари вергул, нукта, икки нукта, кичик, катта ва ўрта қавсчи ҳам ишлатади.

3. Сонлар, ўзгармаслар ва ўзгарувчилар

Паскал тилида қуйидаги турдаги сонлар ишлатилади:

- ҳақиқий турдаги сонлар;
- бутун турдаги сонлар;
- белгилар (символлар).

Ҳақиқий турдаги сонлар умумий ҳолда қуйидаги кўринишда бўлади:

$s\ a1a2...an.\ b1b2...bk$

Бу ерда s ишора(+ ёки -) ёки бўш жой; a1a2...an бутун қисм; b1b2...bk каср қисм. Масалан: +3,147 сони +3.147 ёки 3.147

-143,03 сони -143.03

57,0 сони 57.0

0,493 сони 0.493 ёки .493

Ҳақиқий сонларнинг ўзгариш диапазони компьютернинг турига қараб турлича бўлади. $10^{-38} < X < 10^{38}$ X - ихтиёрий сон. Улар экспоненциал (даражали) кўринишда ифодаланиши ҳам мумкин.

Масалан: $0,43 \cdot 10^{-6}$ компьютерда .43E-6

0,0003 компьютерда 3E-4

Бутун сонлар умумий ҳолда қуйидагича ёзилади s a1a2...an.

Масалан: +345 сони +345 ёки 345

-106 сони -106

Бутун сонлар ўзгариш диапозони -32768 дан +32767 гача. Агар бутун сон қиймати бу диапозондан чиқса, у ҳақиқий сон шаклида ифодаланади ёки компьютер турига қараб, у ўнлилик санок тизимида ифодаланиши ҳам мумкин. Белгили ўзгармаслар диапазони 0 дан 255 тагача бўлади. Мисол. `Pascal`, `405.5`

Паскал тилида идентификатор сўзи ишлатилиб улар ўзгармасларни, ўзгарув-чиларни, белги(метка), процедура ва функциялар номларини белгилашда ишлатилган номлардир. Идентификатор латин алфавити ҳарфларидан бошланиб қолганлари белги ёки рақам кетма-кетлигидан ташкил топган бўлиши мумкин. Масалан: xx, xx1, alfa&.

4.Ўзгарувчилар ва ўзгармасларнинг дастурда тавсифланиши

Паскал тилида дастур ишлаш мобайнида қиймати ўзгармайдиган идентификаторлар **ўзгармаслар** дейилади. Улар дастурнинг бош қисмида **Const** сўзи билан эълон қилиниб унга аниқ қиймат тенглаштирилади.

Мисол. Const aa1=2.27; Pi=3.14; Radius=14;

Дастурда қийматлари ўзгариши мумкин бўлган идентификаторга **ўзгарувчилар** дейилади. Улар дастур бош қисмида **Var** сўзи билан эълон қилинади. Ўзгарувчилар номи келтирилиб, уларнинг турлари берилади. Ўзгарувчилар турлари асосан бутун, ҳақиқий, белгили(символли), матнли ва мантикий бўлади. Улар мос равишда бутун - **Integer**, ҳақиқий - **Real**, белгили- **Char**, матнли(қаторли) - **String** ва мантикий- **Boolean** деб ёзилади. Масалан: Var a, d1, alfa : Integer; c121, df : Real;

Etx, xx : Char; st: String; fl : Boolean;

Мантикий ўзгарувчилар фақат иккита қиймат қабул қилади: "True" (чин) ва "False"(ёлғон).

Маълумотлар турларини умумий ҳолда иккига ажратиш мумкин:

- стандарт турлар. Бу турлар Паскал тили томонидан аниқланган бўлади;
- дастурчи томонидан киритиладиган (аниқланадиган) турлар.

Стандарт турлар таркибига қуйидагилар киради: бутун, ҳақиқий, белгили (символ), қатор (строк), мантикий, кўрсаткичли ва variant.

Дастурчи турларни дастурнинг **Var** бўлимида ўзгарувчиларни тавсифлашда аниқлайди ёки махсус турларни аниқлаш учун бўлим бўлган турларни тавсифлаш **Type** бўлимида аниқлайди. Бу бўлим умумий ҳолда қуйидагича бўлади.

Type

<тур номи>=<турнинг тавсифи>;

Мисол: Type

TColor=(Red,Blue,Black);

Var Color1,Color2,Color3: TColor;

Type бўлимида дастурчи томонидан янги Tcolor номли тур киритилмоқда ва у Red, Blue, Black мумкин бўлган қийматларни қабул қилиши мумкин.

Var бўлимида дастурчи томонидан тури аниқланган учта Color1, Color2, Color3 ўзгарувчилар тавсифланмоқда.

Бу ўзгарувчиларни тўғридан тўғри қуйидагича ҳам тавсифлаш мумкин.

Var Color1,Color2,Color3: (Red,Blue,Black);

Стандарт турларни Type бўлимида тавсифлаш шарт эмас, уларни тўғридан тўғри Var бўлимида тавсифлаш мумкин.

5.Стандарт ва ностандарт математик функциялар

Стандарт математик функциялар.

Функция номи	Тилда ёзилиши	Функция номи	Тилда ёзилиши
Sinx	SIN(x)	$\sqrt{x}$	SQRT(x)
Cosx	COS(x)	Arctgx	ARCTAN(x)
Ln x	Ln(X)	x	ABS(x)
e ^x	EXP(x)	x ²	SQR(x)
		a ^b	EXP(b*LN(a))

Ностандарт математик функциялар.

$$1. \sec x = \frac{1}{\cos x}; \quad 2. \operatorname{cosec} x = \frac{1}{\sin x}; \quad 3. \operatorname{Tgx} = \frac{\sin x}{\cos x}; \quad 4. \operatorname{Arcctgx} = \operatorname{Arctg} \frac{1}{x};$$

$$5. \operatorname{Arc} \sin x = \operatorname{Arctg} \frac{x}{\sqrt{1-x^2}}; \quad 6. \operatorname{Arc} \cos x = \operatorname{Arctg} \frac{\sqrt{1-x^2}}{x}; \quad 7. \operatorname{Arc} \sec x = \operatorname{Arctg} \frac{1}{\sqrt{1-x^2}};$$

$$8. \operatorname{Arc} \operatorname{cosec} x = \operatorname{Arctg} \sqrt{1-x^2}; \quad 9. \operatorname{Log}_a b = \frac{\operatorname{Ln} b}{\operatorname{Ln} a}; \quad 10. \operatorname{Padian} = \frac{\operatorname{Gradius} \cdot \pi}{180}$$

Ўзгартириш функциялари.

Функция	Қиймати
Chr(n)	Коди n га тенг символ
IntToStr(k)	Бутун k ни тасвирловчи сатр
FloatToStr (n)	Ҳақиқий n тасвирлови сатр
StrToInt (s)	Сатрни бутун сонга ўтказиш
StrToFloat (s)	Сатрни ҳақиқий сонга ўтказиш

Round (n)	Ҳақиқий сонни яхлитлаш
Trunc (n)	Ҳақиқий сон каср қисмини олиб ташлаш
Frac(n)	Касрли соннинг каср қисми
Int (n)	Каср соннинг бутун қисми

6. Арифметик ва мантикий ифодалар

Ифодаларда ҳисоблашлар тартиби қавслар ичидаги ифодалар бажарилгандан кейин қуйидаги амаллар бажарилади:

1. NOT амали;
2. *, /, DIV, MOD, AND;
3. +, -, OR;
4. таққослаш белгилари: <, >, <=, >=, <>, =, IN.

Ифодадаги амал натижаси қандай турда бўлиши амалларда қатнашаётган ўзгарувчиларнинг турларига боғлиқ. Агар иккита ўзгарувчининг тури Integer ёки Real бўлса, амал натижаси ҳам Integer ёки Real бўлади. Агар бири Integer иккинчиси Real бўлса натижа Real бўлади. NOT, OR, AND ва таққослаш амалларининг натижалари эса Boolean турида бўлади.

Pascal тилида шарт - бу мантикий турдаги ифода бўлиб, у фақат «чин» (True) ёки «ёлғон» (False) қийматни қабул қилади.

Қуйидаги мантикий амаллар ишлатилади: NOT-«инкор»; AND-«мантикий ва»; OR-«мантикий ёки».

Бу мантикий амалларнинг бажарилиш натижалари қуйидагича:

Op1	Op2	Op1 AND Op2	Op1 OR Op2	NOT Op1
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

Масалан: (5<6) AND (6<50) -мантикий ифода рост (True),
 (20>0) OR (20<0.5) -мантикий ифода рост (True),
 NOT(100>3) -мантикий ифода ёлғон (False).

Мантикий ифодаларни бирор бир мантикий ўзгарувчига юбориш ҳам мумкин.

Масалан: F:=(A<B) AND (A<C);

Бу ерда, агар иккала шарт бажарилгандагина F мантикий ўзгарувчи “чин” (True) қийматни қабул қилади. Акс ҳолда “ёлғон” (False) қийматни қабул қилади.

7. Паскал тилида дастур структураси

Паскал тилида дастур матни бош ва асосий бўлимдан ташкил топади. Бош бўлим дастур номи ва ўзгарувчилар, ўзгармаслар, массивлар, белгилар(меткалар), процедуралар ва функцияларни тавсифлашдан иборат бўлади. Асосий бўлим дастур танаси дейилиб, унда дастурда бажариладиган

хамма операторлар кетма-кетлиги берилади ва у Begin (бошламоқ) сўзи билан бошланиб End (тугаш) сўзи билан тугайди. Умумий ҳолда дастур структураси қуйидаги кўринишга эга:

```
Program <дастур номи>;  
Uses <Фойдаланадиган библиотекалар (модуллар) рўйхати>;  
Label <Ишлатиладиган белгилар(меткалар) рўйхати>;  
Const <Ишлатиладиган ўзгармасларни аниқлаш>;  
Type <Янги турларни аниқлаш>;  
Var <Ўзгарувчиларни эълон қилиш>;  
      <Процедура ва функцияларни аниқлаш>  
Begin  
      <Асосий қисм >  
End.
```

8.Маълумотларни киритиш ва чиқариш операторлари

Бирор бир масалани ечишнинг чизикли бўлган алгоритмига дастур тузишда алгоритмдаги келтирилган кетма-кетликлар асосида операторлар ёзилади. Бундай дастурларни тузушда асосан ўзгарувчилар қийматини киритиш, натижаларни чиқариш ва шу билан бирга ўзлаштириш операторлари ишлатилади.

Компьютер фойдаланувчи томонидан қўйилган масалани аниқ ва тушунарли кўрсатмалар берилгандагина бажара олади. Бу кўрсатмалар маълум бир маънони англатувчи сўзлардан иборат бўлиб, компьютерга қандай операцияни бажариш лозимлигини билдиради ва бу кўрсатмаларга **операторлар** дейилади. Операторлар дастур ишлаганда кетма-кет равишда бажарилади. Паскал тилида бир сатрда бир неча операторларни ёзиш умкин. Дастурдаги ўзгарувчилар қийматларини дастур ичида ўзлаштириш оператори ёрдамида ҳам бериш мумкин. Лекин дастурда ўзгарувчи қийматини ташқаридан киритиш қулайлик туғдиради ва умумийликни таъминлайди.

Read оператори ўзгарувчилар қийматларини экрандан компьютер хотирасига киритиш учун ишлатилади. У қуйидаги кўринишларга эга.

```
Read(c1,c2,...,cn); Readln(c1,c2,...,cn); Readln;
```

бу ерда c1,c2,...,cn - ўзгарувчилар номи; ln - қўшимчаси қийматни киритиб кейинги қаторга ўтишни билдиради.

Мисоллар: Read(Sm1,Sm2); Readln(x1,x2,x3); Readln;

Бу ерда биринчи оператор Sm1 ва Sm2 ўзгарувчилар қийматини экрандан киритади. Иккинчи оператор эса x1,x2,x3 ўзгарувчилар қийматини экрандан киритади ва киритишни кейинги қаторга ўтказди. Охирги оператор эса киритишни кутади ва қатор ўтказди.

Write оператори оддий маълумотларни ва ўзгарувчилар қийматларини компьютер экранига чиқариш учун ишлатилади. У қуйидаги кўринишларга эга.

```
Write(c1,c2,...,cn); Writeln(c1,c2,...,cn); Writeln;
```

бу ерда $c_1, c_2, \dots, c_n$ - оддий матнлар ёки ўзгарувчилар номи; ln - қўшимчаси чиқаришни кейинги қаторга ўтишни билдиради.

Мисоллар: `Write(Summa);` `Write('Натижа йук');`
`Write('Тенглама ечими x1=', x1, 'x2=', x2);`

Оддий маълумотларни чиқаришда улар матн деб қаралади ва у қўштирноқ ичида ёзилади. Чиқариш оператори ёрдамида ўзгарувчилар қийматини формат кўринишда ҳам бериш мумкин: `Write(c:m:n);` бу ерда c -ўзгарувчи; m -шу ўзгарувчи қиймати узунлиги; n -қийматнинг каср қисми ва унда $n-1 < m$ бўлиши керак.

Мисол. `Write(x:8:4);`

Агар $x=155.01021$ бўлса, қуйидаги ёзув чиқади 115.0102.

`Write('Махсулот сони:', kol:5);`

Агар $kol=15$ бўлса, қуйидаги ёзув экранга чиқади. Махсулот сони:

15

Дастур матнини тушунтириш мақсадида кўпинча дастурда изоҳлар келтирилади. Дастурда изоҳлар исталган жойда берилиши мумкин. Изоҳ катта қавс ичида ёзилади.

Масалан: { Бу матн дастурга изоҳ беради } { Бу жойда ечим аниқланмоқда }

Дастурда маълум ҳисоблашларнинг натижаларини бирор бир ўзгарувчида сақлаш учун ўзлаштириш (юбориш) оператори ишлатилиб, у «:=» белгиси ёрдамида қиймат юборилиши керак бўлган ўзгарувчидан кейин қўйилади.

Масалан: $i:=0$; i -қиймати нолга тенглашади, яъни i ўзгарувчига нол юборилади деб тушунилади. Бунда машина i ўзгарувчи учун ажратилган хотирасига нол ёзиб сақлайди. Мисол: $B:=5$; $C:=4$; $A:=(B+C)/2$;

Бу ерда, агар A бутун идентификатор бўлса, унинг қиймати 4 га, акс ҳолда эса 4.5 қийматга эга бўлади.

Ўзлаштириш оператори ёрдамида ихтиёрий ифодаларни ҳисоблаш ва натижаларни хотирага жойлаштириш мумкин.

Чизиқли структурали алгоритмларни дастур шаклида ёзиш учун олдин ишлатиладиган ўзгарувчилар руйхати келтирилиб, кейин алгоритмдаги бажарилишлар кетма-кет равишда амалга оширилиши керак.

Мисол: Текисликдаги икки нуқта орасидаги масофани топиш дастури.

Program XY;

Var x1,y1,x2,y2,d: Real;

Begin

Write('Нуқта координаталарини киритинг:'); Readln(x1,y1,x2,y2);

d:=Sqrt(Sqr(x1-x2)+Sqr(y1-y2));

Writeln('Масофа=',d);

Readln;

End.

МАЪРУЗА №3

ШАРТЛИ ВА ШАРТСИЗ ЎТИШ ОПЕРАТОРЛАРИ

Р е ж а:

1. Кириш.
2. Шартли ўтиш операторлари.
3. Шартсиз ўтиш ва танлаш операторлари.
4. Назорат саволлари.

А д а б и ё т л а р

1. Файсман А. Профессиональное программирование на Турбо Паскале. 1992й.
2. Арипов М., Ҳайдаров А. Информатика асослари, Ўқув қўлланма, Тошкент, Ўқитувчи” 2002й.
3. Шохамидов Ш., Неъматов А. ва бошқалар. “Информатика” фанидан маърузалар матнлари. ТТЕСИ, Тошкент-2003й.

1. Кириш

Шундай масалалар борки уларнинг алгоритми тўғридан тўғри кетма-кет равишда бажарилмайди. Бунда шартнинг берилишига қараб у ёки бошқа ҳисоблашлар бажарилади. Бундай ҳисоблаш жараёнларни дастурлаш Паскал тилида шартли ва шартсиз ўтиш операторлари ёрдамида амалга оширилади. Тармоқланувчи ҳисоблаш жараёни таркибида яна тармоқланишлар бўлиши мумкин. Бундай жараёнларга мураккаб тармоқланувчи ҳисоблаш жараёнлари дейилади.

2. Шартли ўтиш оператори

Pascal тилида шартли ўтиш операторининг икки хил кўриниши мавжуд: тўлиқ ва қисқа.

Тўлиқ кўриниш:

```
If <шарт> then Begin
    <шарт рост бўлганда бажариладиган операторлар>
End
Else
    Begin
    <шарт ёлғон бўлганда бажариладиган операторлар>
End;
```

Қисқа кўриниш:

```
If <шарт> then Begin
```

<шарт рост бўлганда бажариладиган операторлар>

End;

Бу ерда IF -агар; then -у ҳолда; else -акс ҳолда маъносини билдирувчи хизматчи (калит) сўзлар.

Биринчи кўринишдаги шартли операторда, агар шарт бажарилса биринчи Begin ва end ичидаги операторлар кетма-кет бажарилади, акс ҳолда иккинчи Begin ва end ичидаги операторлар кетма-кет бажарилади.

Иккинчи кўринишдаги шартли оператор қуйидагича ишлайди. Агар берилган шарт бажарилса Begin ва end ичидаги операторлар кетма-кет бажарилади, акс ҳолда улар бажарилмайди.

Агар бажарилувчи операторлар сони битта бўлса Begin ва End сўзларини ёзиш шарт эмас.

Мисоллар:

1) If A>0 Then Begin C:=1; B:=C+1; End

Else Begin C:=0; B:=4; End;

2) If D=A Then D:=A Else A:=D;

Ҳар бир шартли ўтиш оператори ичида бошқа ички шартли операторлар жойлашиши ҳам мумкин, масалан If b1 then a1 else If b2 then a2 Else a3;

Мисоллар.

A:=0.5; B:=-1.7; IF A<B THEN A:=B ELSE B:=A;

Жавоб: 0.5<-1.7 ёлғон бўлганлиги сабабли B:=A оператор бажарилади, ва бунда A=0,5 ва B=0,5 экенлиги келиб чиқади.

A:=0.1; B:=0.1; C:=0.5; D:=0;

IF (A<B) OR (A>C) THEN D:=B+C ELSE

IF B=A THEN BEGIN D:=C; C:=A; END;

Жавоб: (0.1<0.1)ёки(0.1>0.5) бу мантикий ифода ёлғон бўлганлиги сабабли B=A шарт текширилади. Бу шарт чин бўлганлиги сабаб D=0,5 га, C=0,1 қийматларга тенг экенлиги келиб чиқади.

3.Шартсиз ўтиш ва танлаш операторлари

Дастурда шундай ҳолатлар бўладиган операторларнинг бажарилиш шартига қараб дастурнинг у ёки бу қисмига тўғридан-тўғри ўтишга тўғри келади. Бундай ҳолатларда шартсиз ўтиш операторидан фойдаланиш мумкин.

Шартсиз ўтиш операторининг кўриниши қуйидагича:

Goto n;

Бу ерда n -белги(метка) бўлиб идентификатор ёки бутун сон бўлиши мумкин. Goto - ўтиш маъносини билдиради.

n- белги дастурнинг бош қисмида Label сўзи ёрдамида эълон қилинган бўлиши шарт. n бошқарилиш узатиладиган жойга n: шаклида қўйилади.

Мисол:

.....

Goto L2;


```
.....  
L2: C:=x*y;
```

.....
Кўп ҳолларда барор бир параметрнинг қийматига қараб керакли операторларни бажаришга тўғри келади. Бундай ҳолларда танлаш операторини ишлатган қулай. Танлаш оператори кўриниши қуйидагича бўлади:

```
Case s of  
  1: A1;  
  2: A2;  
  .....  
  n: An;  
  Else Begin  
    <B1,B2,..Bn>  
  End;  
End;
```

Бу ерда Case -хизматчи сўз бўлиб танлаш маъносини беради; of -«дан» маъносини беради; s-оператор селектори; 1,2,..n-оператор белгилари; A1,A2,...An ва B1,B2,...Bn-операторлар.

Case оператори тармоқланиш жараёнида берилган бир неча операторидан бирини танлаш йўли билан амалга оширади. Операторлар кетма-кетлигини танлаш оператор селекторининг қийматига қараб аниқланади. Оператор селектори ҳақиқий бўлмаган ўзгарувчи ёки ифода бўлиши мумкин. Агар оператор селектори қиймати оператор белгилари ўзгармас қийматига тенг бўлмаса B1,B2,...Bn-операторлари кетма-кет бажарилади. Шартли ўтиш операторининг қуйидаги кўриниши

```
If B Then A1 Else A2;  
танлаш операторининг қуйидаги операторига эквивалентдир.  
Case B of  
  True: A1;  
  False: A2;  
End;
```

Мисол: $ax^2+bx+c=0$ квадрат тенгламанинг илдизларини топиш дастури тузилсин.

```
Program kvт;  
Var a,b,c,x1,x2,d:Real;  
label L1;  
Begin  
  Writeln('Kvadrat tenglama koeff.kiriting:');  
  Write('a='); Readln(a); Write('b='); Readln(b); Write('c='); Readln(c);  
  If a=0 Then begin x1:=c/b; Write('Javob bitta x=',x1); Goto L1; end;  
  d:=b*b-4*a*c;  
  If d<0 Then Write ('Haqiqiy echim yo`q')  
    Else begin x1:=(-b+Sqrt(d))/2/a; x2:=(-b-Sqrt(d))/2/a;
```

```
Writeln('x1=',x1); Write('x2=',x2); Readln;  
End;
```

```
L1:  
End.
```

МАЪРУЗА №4 **ЦИКЛ ОПЕРАТОРЛАРИ**

Р е ж а:

- 1.Кириш.
- 2.FOR цикл оператори.
- 3.WHILE цикл оператори.
- 4.REPEAT цикл оператори.
- 5.Мураккаб цикллар.
- 6.Назорат саволлари.

А д а б и ё т л а р

- 1.Файсман А. Профессиональное программирование на Турбо Паскале. 1992й.
- 2.Арипов М.,Хайдаров А. Информатика асослари, Ўқув қўлланма,Тошкент, Ўқитувчи” 2002й.

1.Кириш

Кўплаб шундай масалалар борки параметрларнинг ўзгаришига қараб маълум ҳисоблашлар бир неча марта такрорланиб бажарилиши мумкин. Масалан, бирор бир функцияни номълум х нинг бир неча қийматида унинг мос қийматларини ҳисоблаш керак дейлик. Бундай ҳисоблашларни компютерда дастур тузиб бажариш учун циклик дастурлар тузиш керак бўлади. Бу каби дастурларни шартли операторлар ёрдамида тузса ҳам бўлади. Лекин Паскал тилида циклик структурали дастур тузиш учун бир неча махсус операторлар мавжуд. Улар For, While ва Repeat операторларидир.

2.FOR цикл оператори

For оператори такрорланишлар сони аниқ бўлган циклик жараёнлар ташкил этишда ишлатилади. Унинг умумий кўриниши қуйидагича:

For i:=m1 to m2 Do S;

Бу ерда i-цикл параметри; m1,m2 -i параметрининг бошланқич ва охириги қиймати бўлиб, улар ўзгармас сон ёки ифода бўлиши мумкин; S-цикл танаси бўлиб, бир неча операторлардан ташкил топиши мумкин.

Агар цикл танаси бир неча оператордан иборат бўлса улар **Begin** ва **End** ичига олинади.

Мисол. 1,2,...10 сонлар йиғиндисини ҳисоблаш дастурсини тузинг.
Program S10;

```

Const kn=10;
Var i: Integer; S: Real;
Begin
  S:=0;
  For i:=1 to kn do S:=S+i;
  Write ('S=',S); Readln;
End.

```

Агар то сўзни **DoWnto** сўзига алмаштирилса цикл параметри тескари бўйича ўзгаради, яъни -1 қадам билан. У ҳолда цикл кўриниши қуйидагича бўлади: **For i:=m1 DoWnto m2 Do S;**

Мисол. 10 дан 1 гача сонларни экранга чиқариш дастурсини тузинг.

```

Program SP;
Var i: Integer;
Begin
  For i:=10 DoWnto 1 do Write (i);
End.

```

3.WHILE цикл оператори

While цикл оператори такрорланишлар сони олдиндан аниқ бўлмаган ҳолларда такрорланишни бирор бир шарт асосида бажаради. Берилган шарт олдин текширилади ва кейин шартнинг бажарилишига қараб керакли операторлар бажарилади. Бу операторнинг умумий кўриниши қуйидагича:

While B Do S;

Бу ерда B -мантиқий ифода; S -цикл танаси бўлиб, бир ёки бир неча операторлар кетма-кетлигидан иборат бўлиши мумкин. Мантиқий ифода 'True' ёки 'False' қиймат қабул қилади.

Агар мантиқий ифода 'True' қиймат қабул қилса S операторлари бажарилади, акс ҳолда бажарилмайди, яъни цикл ишлашдан тўхтади.

Мисол. 1,2,...,10 сонлар йиқиндисини ҳисоблаш дастурсини тузинг.

```

Program S10;
Const n=10;
Var i: Integer; S: Real;
Begin
  S:=0; i:=0;
  While i<n do Begin i:=i+1; S:=S+i; End;
  Write ('S=',S);
End.

```

4.REPEAT цикл оператори

Repeat цикл оператори ҳам такрорланишлар сони олдиндан аниқ бўлмаган ҳолларда такрорланишни бирор бир шарт асосида бажаради. Олдин цикл танасидаги операторлар кетма-кетлиги бажарилади. Берилган шарт кейин текширилади. Агар берилган шарт рост (True) бўлса, бошқарув

циклдан кейинги операторни бажаришга ўтади, акс ҳолда цикл такрорланади. Бу операторнинг умумий кўриниши қуйидагича:

Repeat

S

Until B

Бу ерда В -мантикий ифода, 'True' ёки 'False' қиймат қабул қилади; S - цикл танаси бўлиб, бир ёки бир неча операторлар кетма-кетлигидан иборат бўлиши мумкин. Агар мантикий ифода 'False' қиймат қабул қилса циклда такрорланиш давом этади, акс ҳолда тўхтади.

Мисол. 1,2,...,10 сонлар йиғиндисини ҳисоблаш дастурсини тузинг.

Program S10;

Const n=10; Var i: Integer; S: Real;

Begin

S:=0; i:=0;

Repeat i:=i+1; S:=S+i; Until i>=n;

Write ('S=',S);

End.

Одатда WHILE оператори REPEAT операторига нисбатан кўп ишлатилади. Бунга сабаб кўпчилик масалаларда цикл тугалланиш шarti цикл бошланмасдан текшириш мақсадга мувофиқдар.

5.Мураккаб цикллар

Кўпчилик масалаларни ечишда тузилган дастурда ичма-ич жойлашган цикллар ташкил этишга тўғри келади. Бундай циклларга мураккаб цикллар дейилади. Мураккаб циклларда қуйидаги талаблар бажарилиши зарур.

- ички цикл ташқи цикл ичида тўлиқ ётиши керак;
- цикллар бир-бири билан кесишмаслиги керак;
- цикл ичига ташқаридан тўғридан-тўғри кириш мумкин эмас;
- цикл параметрлари ҳар хил идентификаторлар билан белгиланиши керак;

Мисол. $S = \sum_{i=1}^{10} \prod_{j=1}^5 \frac{i+j}{\sqrt{i \cdot j}}$ ифодани ҳисоблаш дастурсини тузинг.

Бу формулада агар йиғиндини очсак у қуйидаги кўринишга келади.

$$S = \sum_{i=1}^{10} \prod_{j=1}^5 \frac{i+j}{\sqrt{i \cdot j}} = \prod_{j=1}^5 \frac{1+j}{\sqrt{1 \cdot j}} + \prod_{j=1}^5 \frac{2+j}{\sqrt{2 \cdot j}} + \dots + \prod_{j=1}^5 \frac{10+j}{\sqrt{10 \cdot j}}$$

Program SP;

Var i,j: Integer; S,P: Real;

Begin

S:=0;

For i:=1 to 10 do

Begin

P:=1; For j:=1 to 5 do P:=P*(i+j)/Sqrt(i*j);

S:=S+P;

End; Write ('S=',S);
End.

МАЪРУЗА № 5

МАССИВЛАР ВА УНИНГ ЭЛЕМЕНТЛАРИНИ ТАРТИБЛАШ

УСУЛЛАРИ

Р е ж а:

- 1.Кириш.
- 2.Массивлар ва улар устида иш юритиш.
- 3.Массив элементларини тартиблаш усуллари.
- 4.Маъруза буйича саволлар.

А д а б и ё т л а р

- 1.Культин Н.Б. Программирование в Turbo Pascal 7.0 и Delphi. “Санкт-Петербург”, 1997г.
- 2.Файсман А. Профессиональное программирование на Турбо Паскале. 1992й.
- 3.Арипов М.,Хайдаров А. Информатика асослари, Ўқув қўлланма,Тошкент, Ўқитувчи” 2002й.

1.Кириш

Кўп ҳолларда жадвал ёки матрицалар кўринишидаги маълумотлар билан иш юритиш керак бўлади. Жадвалда маълумотлар жуда кўп бўлгани сабаб, уларнинг ҳар бир ячейкасидаги сонни мос равишда битта ўзгарувчига қиймат қилиб берилса улар устида иш бажариш анча нокулайликларга олиб келади. Шу сабаб дастурлашда бундай муаммолар массивларни ишлатиш ёрдамида ҳал қилинади.

2.Массивлар ва улар устида иш юритиш.

Массив - бу бир ном билан белгиланган қийматлар тўплами ёки жадвалдир. Массивнинг ҳар бир элементи массив номидан сўнг ўрта қавс ичига олинган рақам ва арифметик ифода ёзиш билан белгиланади. Қавс ичидаги рақам массив индексини белгилайди. Векторни бир ўлчовли массив, матрицани икки ўлчовли массив деб қараш мумкин.

Бир ўлчовли массивда унинг ҳар бир элементи ўзининг жойлашган ўрин номери билан аниқланади ва номери қавс ичида индекс билан ёзилади. Икки ўлчовли массив элементи ўзи жойлашган сатр ва устун номерлари ёрдамида аниқланади. Шу сабаб икки ўлчамли массив элементи иккита индекс орқали ёзилади. Масалан: $A[i,j]$ бу ерда i -сатр номери j -устун номерини билдиради.

Ҳар бир массив ўз ўлчамига эга бўлиб ва у дастурда эълон қилинган бўлиши керак. Массивни эълон қилиш дастурнинг бош қисмида берилиб, унинг ёзилиши умумий ҳолда қуйидагича бўлади:

<Массив номи>:Array[ўлчам] of <элемент тури>;

Масалан:

A,B:Array[1..100] of real;

C,A1,D:Array[1..10,1..15] of real;

Бу ерда А ва В массивлари 100тадан элементга эга. C,A1,D1 массивлари эса $10 \times 15 = 150$ тадан элементга эга.

Массивларни эълон қилишдан мақсад массив элементлари учун компьютер хотирасидан жой ажратишдир.

Массив элементлари қийматларини киритиш учун цикл операторларидан фойдаланилади. Мисол: For i:=1 to 10 do Read(A[i]);

Бу мисолда А массивнинг 10 та элементи қийматини экрандан кетма-кет киритиш керак бўлади. Худди шундай массив қийматларини экранга чиқариш ҳам мумкин. Мисол: For i:=1 to 10 do Write(A[i]);

Дастурда массив элементларини ишлатганда уларнинг индекси эълон қилинган чегарадан чиқиб кетмаслиги керак.

3.Массив элементларини тартиблаш усуллари.

Массивни тартиблаштиришнинг бир неча усуллари (алгоритмлари) мавжуд. Улардан қуйидаги усулларни қараб чиқамиз:

-танлаш усули;

-алмаштириш усули.

Танлаш усули ёрдамида массивни ўсиш бўйича тартиблаштириш алгоритми қуйидагича:

1.Массивнинг биринчи элементида бошлаб қараб чиқилиб энг кичик элемент топилади.

2.Биринчи элемент билан энг кичик элемент жойлари алмаштирилади.

3.Иккинчи элементида бошлаб қараб чиқилиб энг кичик элемент топилади.

4.Иккинчи элемент билан энг кичик элемент жойлари алмаштирилади.

5.Бу процесс битта охириги элементгача такрорланади.

Бу алгоритм дастури қуйидагича бўлади:

Program Sort;

Const n=5;

Var i, j, min, k, buf: Integer; a: Array[1..n] of Integer;

Begin

Writeln ('Массивни тартиблаштириш');

Write (n:3, ' -та массив элементини киритинг');

For k:=1 to n Do Read(a[k]);

For i:=1 to n-1 Do

Begin { кичик элементни топиш }

```

min:=i;
For j:=i+1 to n Do
  Begin
    If a[j]<a[min] then min:=j;
    buf:=a[i]; a[i]:=a[min]; a[min]:=buf;
    For k:=1 to n Do Write (a[k], ' ');
    Writeln;
  End;
End;
Writeln('Массив тартиблаштирилди. ');
End.

```

Дастур натижаси:

```

Массивни тартиблаштириш
5 та массив элементини киритинг
12 -3 56 47 10
Тартиблатириш
-3 12 56 47 10
-3 10 56 47 12
-3 10 12 47 56
-3 10 12 47 56
Массив тартиблаштирилди.

```

Алмаштириш усули ёрдамида массив элементларини ўсиб боришда тартиблаштириш алгоритми қуйидагича:

1.Массивнинг биринчи элементидан бошлаб кетма-кет ҳамма қўшни элементлар бир-бири билан солиштирилиб, агар биринчиси иккинчисидан кичик бўлса улар жойи алмаштирилиб борилади.

2.Бу процесс давомида кичик қийматли элементлар массив бошига катта элементлар эса охирига силжитилиб борилади. Шу сабаб бу усул «пузирка» усули ҳам дейилади.

3.Бу процесс массив элементлар сонидан битта кам марта такрорланади.

Масалан:

3 2 4 5 1 бунда 3 билан 2 ва 5 билан 1 алмаштирилади.

2 3 4 1 5 бунда 4 билан 1 алмаштирилади.

2 3 1 4 5 бунда 3 билан 1 алмаштирилади.

2 1 3 4 5 бунда 2 билан 1 алмаштирилади.

1 2 3 4 5

Бу алгоритм дастури қуйидагича бўлади:

```

Program Sort;
Const n=5;

```

```

    Var i,j,min,k,buf: Integer; a: Array[1..n] of Integer;
Begin
    Writeln ('Массивни пузирек(купикча) усулида тартиблаштириш');
    Write (Size:3,'та массив элементини киритинг');
    For k:=1 to n Do Read(a[k]);
    Writeln ('Тартиблатириш');
    For i:=1 to n-1 Do
        Begin
            For k:=1 to n-1 Do
                Begin
                    If a[k]>a[k+1] then
                        Begin
                            buf:=a[k]; a[k]:=a[k+1]; a[k+1]:=buf;
                        End;
                End;
            For k:=1 to n Do Write (a[k], ' '); Writeln;
        End;
    Writeln('Массив тартиблаштирилди. ');
End.

```

Дастур натижаси:

Массивни пузирек усулида тартиблаштириш

5 та массив элементини киритинг

3 2 4 1 5

Тартиблаштириш

2 3 4 1 5

2 3 1 4 5

2 1 3 4 5

1 2 3 4 5

Массив тартиблаштирилди.

Массивда энг кичик ёки энг катта элементни излаш алгоритми маълумки биринчи элемент энг кичик (катта) деб олиниб кейин бошқа элементлар билан кетма-кет солиштирилиб чиқилади. Солиштирилиш охириги элементгача бажарилади. Қуйида бу алгоритм дастури келтирилган:

Program MinMax;

```

Var i,min: Integer; a: Array[1..10] of Integer;

```

```

Begin

```

```

    Writeln ('Массивдан энг кичик элементни излаш');

```

```

    Write (' 10-та массив элементини киритинг');

```

```

    For i:=1 to 10 Do Read(a[i]);

```

```

        min:=1;

```

```

    For i:=2 to 10 Do

```



```

If a[i]<a[min] Then min:=i;
Writeln('Изланаётган энг кичик элемент:',a[min]);
Writeln('Элемент номери',min);
End.

```

МАЪРУЗА №6 ҚИСМ ДАСТУРЛАР

Р е ж а:

- 1.Кириш.
- 2.Процедуралар.
- 3.Функциялар.
- 4.Айрим ностандарт процедура ва функциялар.
- 5.Назорат саволлари.

А д а б и ё т л а р

- 1.Файсман А. Профессиональное программирование на Турбо Паскале. 1992й.
- 2.Арипов М.,Хайдаров А. Информатика асослари, Ўқув қўлланма,Тошкент, Ўқитувчи” 2002й.

1.Кириш

Дастурлаш жараёнида шундай ҳолатлар бўладики, бир хил операторлар кетма-кетлигини дастурнинг бир неча жойларида такроран ёзишга тўғри келади. Бундай такрорланишни йўқотиш мақсадида дастурлашнинг кўпгина тилларида қисм дастур тушунчаси киритилган. Улар мустақил дастур бўлаги сифатида дастурнинг бош қисмида бир маротаба ёзилади.

Паскаль тилида қисм дастур процедура ёки функция кўринишида берилади. Асосий дастур билан процедура орасида ўзгарувчилар қиймат алмашуви формал ва фактик параметрлар ёрдамида амалга оширилади. Процедура ичида яна бир неча процедура ёки функция ишлатилиши мумкин. Дастурда эълон қилинган ўзгарувчилар, шу дастурдаги процедура ва функцияларга нисбатан глобал дейилади. Процедура ва функцилар ичида эълон қилинган ўзгарувчилар локал дейилади. Уларнинг таъсир доираси шу қисм дастур ичида бўлади.

2.Процедуралар

Процедураларни эълон қилиш қуйидагича бўлади.

```

Procedure <проц.номи> (<формал параметрлар>);
    <ўзгарувчи ва ўзгармасларни тавсифлаш қисми>
Begin
    <асосий қисм>

```

End;

Формал параметрларни шу процедура бош қисмида ёки сарлавҳада эълон қилиш мумкин. Масалан. **Procedure AB (x,y: Real);**

Ҳар қандай процедурани кичик бир дастур деб қараш мумкин. Процедура ҳам дастурга ўхшаб бош ва асосий қисмлардан тошқил топади. Бош қисмда процедура номи ва унинг параметрлари эълон қилинади. Асосий қисм операторлар кетма-кетлигидан ташқил топган бўлиб, улар Begin - End ичига олинади. Процедура номи фойдаланувчи тамонидан берилади.

Мисол. Procedure Dr(Var x,h1,h2,z1,z2 : Real);

Var h,z: Real;

Begin

h:=h1/z1+h2/z2; z:=z1/z2; x:=(h+z)/2;

End;

Бу процедурада h1,z1,h2,z2 параметрлар қиймати процедурага мурожат қилинганда аниқланган бўлиши керак. Натижани эса x- параметр узатади. h ва z ўзгарувчилар ички ўзгарувчилардир. Бу процедурага дастурдан қуйидагича мурожат қилинади Dr(x,h1,h2,z1,z2). Процедурага мурожат қилинганда мос параметрлар қиймати бир бирига узатилади. Бериладиган формал ва фактик параметрлар сони тенг ва улар турлари бир хил бўлиши шарт. Лекин параметрлар номлари ҳар хил бўлиши мумкин.

3.Функциялар

Функциялардан фойдаланиш ва уларни ташқил қилиш худди процедура каби бўлиб, у қуйидагича бўлади:

Function <ф-я номи>(<формал параметрлар>):<ф-я тури>;

<ўзгарувчи ва ўзгармасларни тавсифлаш қисми>

Begin

<асосий қисм>

End;

Функциянинг процедурадан фарқи, унга мурожат қилинганда натижа фақат битта бўлиб, у шу функция номига узатилади.

Мисол 1. Қуйидаги ҳисоблашни функцияни ишлатган ҳолда дастурини тузинг.

$$C_n^m = \frac{n!}{m!(n-m)!}.$$

Program Kol;

Var ncm:Real; n,m,i: Integer;

Function Fact (k: Integer): Integer;

Var P,i: Integer;

Begin

P:=1; For i:=1 to k do P:=P*i;

Fact:=P;

End;

```

Begin
  Read(n,m); l:=n-m; ncm:=Fact(n)/Fact(m)/Fact(i);
  Write('ncm=',ncm);
End.

```

Мисол 2. Қуйидаги ҳисоблашни процедурани ишлатган ҳолда дастурсини тузинг.

$$z = \frac{tha - th^2(a - b)}{\sqrt{th(a^2 - b^2)}}, \quad thx = \frac{e^{2x} - 1}{e^{2x} + 1}.$$

```

Program Fun1;
Var a,b,z,c,d,t1,t2,t3: Real;
Procedure Th(Var x,r: Real);
  Var c: Real;
  Begin    c:=exp(2.0*x);  r:=(c-1)/(c+1);    End;
Begin
  Read(a,b);
  th(a,t1); c:=a-b;  th(c,t2);
  d:=Sqr(a)-Sqr(b); th(d,t3);
  z:=(t1+Sqr(t2))/Sqrt(t3);
  Write('z=',z:10:3);
End.

```

4.Айрим ностандарт процедура ва функциялар

Паскаль тилида бир қанча махсус процедура ва функциялар мавжуд бўлиб, улар қуйидаги гуруҳларга бўлинади:

- қаторни қайта ишлаш;
- файллар билан ишлаш;
- динамик ўзгарувчилар учун хотирани бошқариш;
- арифметик функциялар;
- экран билан ишлаш.

Уларнинг айримларини кўриб чиқамиз:

Halt- дастурни бажаришдан тўхтатиш;

Odd(i)- I-тоқ бўлса “True” акс ҳолда “False” қиймат олади;

Edit- бажарилаётган блокдан чиқиш;

Random- 0 дан 1 гача бўлган сонни тасоддифан олиш;

GotoXY(x,y)- курсорни кўрсатилган жойга қўйиш;

ClrScr- экранни тозалаб, курсорни экран бошига қўйиш;

Trunc- аргументнинг бутун қисми:

Str(I;Var S:String)- рақамни символга ўтказиш (I-ифода ёки ўзгарувчи);

Val(S:String; Var P;ko:Integer)- символни рақамга ўтказиш (P-ўзгарувчи);

Length (S:String)- қатор узунлигини аниқлаш.

Pos(st1,st) - қатордаги қатор қисми ҳолатини аниқлаш.

Мисол: st:='Тошкент'; st1:='кент'; p:=pos(st1,st); Жавоб: p=4. Агар изланаётган қатор қисм қаторда йўқ бўлса қиймат нулга тенг бўлади.

Copy(st,m,n) - қатордан фрагмент кесиб олади.

Мисол: st:='Тошкент'; p:=Copy(st,4,4); Жавоб: p='кент'.

Delete(st,m,n) - қатордан фрагмент кесиб олиб ташлайди.

Мисол: st:='Тошкент'; p:=Delete(st,4,4); Жавоб: p='Тош'.

МАЪРУЗА № 7

БЕЛГИЛАР ВА ҚАТОРЛАР

Маъруза режаси:

- 1.Кириш.
- 2.Белгилар ва уларни қайта ишлаш.
- 3.Белгили массивлар.
- 4.Қаторлар ва уларни қайта ишлаш функциялари.
- 5.Маъруза бўйича саволлар.

А д а б и ё т л а р

- 1.Культин Н.Б. Программирование в Turbo Pascal 7.0 и Delphi. “Санкт-Петербург”, 1997г.
- 2.Файсман А. Профессиональное программирование на Турбо Паскале. 1992й.

1.Кириш

Шундай масалалар борки уларни матнлар устида иш бажибгина ҳал этиш мумкин. Масалан, файлдаги матнлар ичидан бирор бир исм ёки фамилияни излаш керак ёки бир хил маълумотни бошқа бир маълумотга алмаштириш керак ёки улар устида бирор бир операцияларни бажариш керак бўлсин дейлик. Дастурлашда бундай масалаларни ечишда белгилар ва қаторлардан фойдаланишга тўғри келади. Дастурлашда уларни ишлатишда белги ва қаторли типдаги махсус ўзгарувчилар ёки ўзгармаслардан фойдаланилади.

2.Белгилар ва уларни қайта ишлаш

Махсус белгиларни қайта ишлаш ва сақлаш учун CHAR типдаги ўзгарувчилар ишлатилади. CHAR типдаги ўзгарувчилар қиймати исталган белги бўлиши мумкин. Белгили типдаги ўзгарувчилар дастурнинг бош қисмида ўзгарувчиларни эълон қилиш бўлимида қуйидагича кўрсатилиши керак:

Var
 ном: CHAR;

бу ерда *ном* – белгили типдаги ўзгарувчи номи.

Мисоллар:

Var

Otv: Char;

Ch, Cr: Char;

Бошқа ўзгарувчилар каби белгили типдаги ўзгарувчилар ҳам киритиш (READ) ёки юбориш (қиймат бериш) операторлари ёрдамида қийматга эга бўлиши мумкин. Агар CHAR типдаги ўзгарувчилар қиймат бериш операциясининг натижасида қиймат қабул қилиши керак бўлса, у ҳолда юбориш операциясининг ўнг томонида CHAR типдаги ифода бўлиши ёки белгили (символли) ўзгарувчи ёки белгили ўзгармас бўлиши керак. Символли ўзгармас очилувчи ва ёпилувчи кавичка ичида ёзилади.

Масалан, қуйидаги дастур натижасида ўзгарувчи c1 ўзгармас қиймат, c2 ўзгарувчи c1 қийматини қабул қилади, кейин эса otvet ўзгарувчи қиймати клавиатурадан киритилади. Бунда otvet ўзгарувчи қиймати сифатида битта белги берилиши керак бўлади. Масалан “Да” белгиси киритилса otvet ўзгарувчиси “Д” қийматни қабул қилади.

Var

c1,c2,otvet: Char;

Begin

c1:=`\*`;

c2:=c1;

Write (`Сиз дастур тузишни ўрганишни хоҳлайсизми?`);

Readln(otvet);

Writeln (`Сизнинг жавобингиз:`, otvet);

End.

CHAR типдаги ўзгарувчини бошқа бир CHAR типдаги ўзгарувчи билан ёки белгили ўзгармас билан таққослаш мумкин. Таққослаш шунга асосланганки, ҳар бир белгига мос сон қўйилган.

`0`<`1`<...<`9`<...<`A`<`B`<...<`Z`<`a`<`b`<...<`z`

Худди шундай крил алфавитида ҳам ҳар бир белгига мос сон берилган:

`A`<`Б`<...<`Ю`<`Я`<`а`<`б`<...<`я`

Белгили ўзгарувчи ва ўзгармасларни шартли операторларда ҳам ишлатиш мумкин. Масалан, қуйидаги дастурда otvet ўзгарувчиси дастур сўровига жавоб беришни текширишда ишлатилмоқда.

Var

otvet: Char;

Begin

....

Writeln (`Hisoblsh davom ettirilsinmi? (д/н)`);

Readln(otvet);

If (otvet=`д`) or (otvet=`Д`) Then begin

...

```
End;
End.
```

Ҳар бир белги сон билан кодлаштирилган. Агар дастурда клавиатурада йўқ белгини экрандан киритиш керак бўлса, у ҳолда CHR функциясидан фойдаланилади. Масалан, Ch:=CHR(218); операторининг бажарилиши натижасида Ch белгили ўзгарувчи '┘' белгисини ўзлаштиради.

Қуйидаги дастур 0 дан 128 гача бўлган белгилар кодировкаси жадвалини экранга чиқаради. Бу кодлардан 7 сидан бошлаб 13 гачаси хизматчи, масалан 7 коди билан киритилгагн белги товуш сигналинини беради, 13 коди эса янги қаторга ўтиш буйруғини чақиради.

```
Var
    ch: Char;           {белги}
    Dec: Integer;       {белгининг ўнликдаги коди}
    i, j: Integer;
Begin
    Dec:=0;
    For i:=0 to 16 do
        Begin
            Dec:=i;
            For j:=1 to 8 do {саккизта қатор }
                Begin
                    If (Dec<7) Or (Dec>=14) Then
                        Write (Dec:4,'-',' ',Chr(Dec):1,Chr(179))
                    Else {7 дан 14 гача белгидан чиқмайди}
                        Write (Dec:4,'- ',Chr(179));
                    Dec:=Dec+16;
                End;
            Writeln; {янги қаторга ўтиш}
        End;
    End.
```

Дастур ишлаши натижаси қуйидагича:

```
0 - ! | 16 -   | 32 - | 48 - 0 | 64 -   | 80 - P | 96 - ' | 112 - p |
1 - ^ | 17- □ | 33- ! | 49- 1 | 65- A | 81- Q | 97- a | 113- q |
2 - ε | 18- □ | 34- " | 50- 2 | 66- B | 82- R | 98- b | 114- r |
3 - э | 19- □ | 35- # | 51- 3 | 67- C | 83- S | 99- c | 115- s |
4 - ∂ | 20- □ | 36- $ | 52- 4 | 68- D | 84- T | 100- d | 116- t |
5 - ♣ | 21- □ | 37- % | 53- 5 | 69- E | 85- U | 101- e | 117- u |
6 - ♠ | 22- □ | 38- & | 54- 6 | 70- F | 86- V | 102- f | 118- v |
7 -   | 23- □ | 39- ' | 55- 7 | 71- G | 87- W | 103- g | 119- w |
8 -   | 24- □ | 40- ( | 56- 8 | 72- H | 88- X | 104- h | 120- x |
```

9 -	25- □	41-)	57- 9	73- I	89- Y	105- i	121- y	
10-	26- □	42- *	58- :	74- J	90- Z	106- j	122- z	
11-	27- □	43- +	59- ;	75- K	91- [	107- k	123- {	
12-	28- □	44- ,	60- <	76- L	92- \	108- l	124-	
13-	29- □	45- -	61- =	77- M	93-]	109- m	125- }	
14- ∃	30- -	46- .	62- >	78- N	94- ^	110- n	126- ~	
15- *	31-	47- /	63- ?	79- O	95- _	111- o	127- □	

Қуйидаги дастур матнларни чиқаришни расмийлатириш, яъни безашда CHR функциясини ишлатади.

Var

t1, tr, b1, br, g, v: Char;

i: Integer;

Begin

t1:=Chr(218); { юқори чап бурчак белгиси }

tr:=Chr(191); { юқори ўнг бурчак белгиси }

b1:=Chr(192); { пастки чап бурчак белгиси }

br:=Chr(217); { пастки ўнг бурчак белгиси }

g:=Chr(196); { горизонтал тўғри чизиқ }

v:=Chr(179); { вертикал чизиқ }

Write (t1);

For i:=1 to 32 do Write(g);

Writeln(tr);

Writeln (v, ' Рамка чизиш учун CHR ', v);

Writeln (v, ' функциясини ишлатишга мисол', v);

Write (b1);

For i:=1 to 32 do Write(g);

Writeln (br);

End.

Дастур ишлаши натижаси қуйидагича:

Рамка чизиш учун CHR функциясини ишлатишга мисол

Товуш сигнаlines 7 коди билан экран орқали киритишни, масалан дастурда маълумотни киритишда агар хатолик юз берганда ёки натўғри маълумот киритилганда ишлатиш мумкин. Қуйидаги дастур паролни текширади, агар парол натўғри бўлса компьютер динамикаси товуш сигнаlines беради.

Label

Bye;

Var

```

Password: Integer;
Begin
  Write ('Пароль'); Readln (Password);
  If Password<>'377' then
    Begin Writeln (Chr(7), 'Пароль натўғри!');
      Goto Bye;
    End;
  Bye:
End.

```

4.Белгили массивлар

Кетма-кетликдаги белгиларни қайта ишлаш ва сақлаш учун белгили массивларни ишлатиш мумкин. Масалан, дастурда клавиатурадан фамилияларни киритиш керак бўлсин. Одатда бундай масалани ечишда белгили массивга кетма-кетликдаги белгиларни киритиш билан амалга оширилади. Бунда киритилган белгилар қатори фамилиядаги ҳарфларнинг сонига тенг олинади. Агар Buf белгили массив SIZE ўлчамга эга бўлса, у ҳолда энг содда усулда массив элементларини FOR оператори ёрдамида киритиш мумкин:

```
For i:=1 to SIZE do Read (Buf[i]);
```

Дастурда белгили массивни киритишда анча эффектив бўлган EOLN (End Of Line) курилган функциясида фойдаланиш мумкин. Бу функция агар массивга белги киритилган бўлса TRUE қийматни қайтаради. Агар бирорта ҳам белги киритилмаган бўлса функция клавиатурадан киритишни кутади. Қуйидаги дастурда белгили массив элементларини киритишда EOLN функцияси ишлатилган.

```

const
  SIZE=30;
Var
  Buf: Array[1..Size] of Char;
  n: Integer; {киритилган қаторнинг реал узунлиги}
Begin
  Write ('->');
  n:=0;
  Repeat
    If Not EOLN then Begin n:=n+1; Read (Buf[n]); End;
  Until EOLN OR (n=Size);
  Writeln ('Киритилган қатор узунлиги ', n, ' белгилар. ');
End.

```

4.Қаторлар ва уларни қайта ишлаш функциялари

Паскал тилида кетма-кетликда берилган белгилар қатор дейилади. Қаторлар дастур бошида STRING сўзи билан эълон қилинади. Ҳар қандай STRING типдаги ўзгарувчи кетма-кетликдаги белгилар узунлиги 255 тадан ошмаган белгилар қийматларни қабул қилади. Қаторли типдаги ўзгарувчилар дастур бошида қуйидагича кўрсатилиши керак:

```
Var
    ном: STRING;
ёки
    ном: STRING [узунлик];
```

бу ерда *ном* – қатор типдаги ўзгарувчи номи.

Мисоллар:

```
Var
    name: STRING [30];
    Buff, RT: STRING;
```

Агар эълон қилинганда ўзгарувчи узунлиги кўрсатилмаса, унда қатор узунлиги 255 белгига тенг деб қабул қилинади. Қаторли ўзгармас очилувчи ва ёпилувчи кавичка ичида ёзилади. Масалан,

```
Parol:='sekret'; ёки Parol:='1997';
```

STRING типдаги ўзгарувчини бошқа бир STRING типдаги ўзгарувчи билан ёки қаторли ўзгармас билан =, <, >, <=, >=, <> операторлари ёрдамида таққослаш мумкин.

Қуйидаги дастур парол сўрайди, агар парол тўғри киритилган бўлса фамилияни киритиш керак бўлади.

```
Var
    Name: String[30];
    Parol: String[6];
Begin
    Write ('Парол (6 белги)?');
    Readln(Parol);
    If Parol='sekret' then begin
        Write ('Сизнинг фамилиянгиз?');
        Readln(Name);
        Writeln(Name, ', Сизга рухсат берилди.');
```

ёқ!');
End.

Таққослаш операциясидан ташқари қаторли ўзгарувчилар ва ўзгармаслар устида қўшиш амалини ҳам бажариш мумкин. Масалан:

```
F_Name:='Ivanov';
L_Name:='Ivan';
Ful_name:=F_Name+L_Name;
```

Бунда Ful_Name ўзгарувчи 'IvanovIvan' қийматни олади.

Turbo Pascal қаторларни қайта ишлашда қўлланиладиган бир неча процедура ва функцияларга эга.

LENGTH функцияси. Бу функция битта қатор типли ўзгарувчи - параметрга эга бўлиб, қатор узунлигини қайтаради. Қайтариладиган функция қиймати бутун сондан иборат, у қатордаги булгилар сонини аниқлайди. Агар қатор бошида ёки охирида бўшлиқлар бўлса уларни ҳисобга олмайди. Масалан, LENGTH('Ivanov') функция қиймати 6 га тенг, LENGTH(' Ivanov Ivan ') функция қиймати 11 га тенг.

DELETE процедураси. Бу процедура қаторнинг маълум бир қисмини йўқотади. Умумий ҳолда бу процедура қуйидагича ёзилади:

Delete (қатор,р,п);

бу ерда *қатор* – қатор типдаги ўзгарувчи; *р* – ўчирилиши керак бўлган бошланғич белгининг номери; *п* – ўчириладиган қатор бўлаги узунлиги. Мисол:

S:='Shahar Sankt-Peterburg';

Delete (S,8,6);

Натижа 'Shahar Peterburg' бўлади.

POS функцияси. Бу функция қаторда қатор бўлагининг жойини аниқлайди. Умумий ҳолда бу функция қуйидагича ёзилади:

POS (қаторбўлаги, қатор);

бу ерда *қаторбўлаги* – қатордан топилиши керак бўлган қатор типдаги ўзгарувчи ёки ўзгармас. Мисол:

р:=POS('Pe', 'Sankt-Peterburg');

Бунда ўзгарувчи *р* қиймати 7 га тенг бўлади. Агар изланаётган қатор бўлаги бўлмаса функция нул қиймат қайтаради.

COPY функцияси. Бу функция қатордан фрагмент, яъни бўлак ажратиб олади. Умумий ҳолда бу функция қуйидагича ёзилади:

COPY (қатор, р, п);

бу ерда *қатор* – қатор типдаги ўзгарувчи ёки ўзгармас; *р* – қатор бўлаги бошланадиган қатордаги биринчи белги номери; *п* – қатор бўлаги узунлиги. Мисол:

st:='Muxandis Ahmedov';Fam:=Copy(st, 10, 7);

Бунда ўзгарувчи *Fam* қиймати 'Ahmedov' бўлади.

VAL процедураси. Бу процедура қатор кўринишида берилган сонларни оддий сонга ўзгартиради (ўтказади). Умумий ҳолда бу процедура қуйидагича ёзилади:

VAL (*қатор, сон, код*);

бу ерда *қатор* – сонни тасвирловчи қатор типдаги ўзгарувчи ёки ўзгармас; *сон* – қатордаги сонни ўзлаштирувчи ўзгарувчи; *код* – процедура қайтарадиган хатолик коди. Агар қаторда берилган сон оддий сонга ўтказилган бўлса хатолик коди нулга тенг бўлади. Мисол:

S:=’891.01’; VAL (S, R, kod);

Натижа R=891.01 бўлади.

МАЪРУЗА №8

ТУРБО ПАСКАЛ ГРАФИК МОДУЛЛАРИ

Р е ж а:

- 1.Кириш.
- 2.Экранни график ҳолатга ўтказиш.
- 3.Graph модулининг процедура ва функциялари.
- 4.Назорат саволлари.

А д а б и ё т л а р

- 1.Культин Н.Б. Программирование в Turbo Pascal 7.0 и Delphi. “Санкт-Петербург”, 1997г.
- 2.Файсман А. Профессиональное программирование на Турбо Паскале. 1992й.

1.Кириш

Компьютер графикаси илмий тажриба ва лойиҳалашнинг автоматлаштирилган тизимларда, робототехника ва бошқа соҳаларда кенг ишлатилади.

Компьютер монитори график ва матнли маълумотларни экранга чиқаради. Монитор махсус видеоадаптер бошқарувида матнли ва графикли ҳолатда ишлайди. Матнли иш ҳолатида компьютер экрани ҳар бир қаторда 80 тадан позиция бўлган 25 та сатрга бўлинади. Графикли иш ҳолатида эса монитордаги ҳар қандай тасвир худди телевизор экранидек, яъни маълум бир ранга бўялган нукталар тўплами каби ҳосил бўлади. Тасвирнинг қанчалик тиниклиги видеоадаптернинг имкониятига, яъни горизонтал ва вертикал бўйича чиқариладиган нукталар миқдорига қараб белгиланади. Масалан, видеоадаптер SVGA экранга қуйидагича - 640x480, 800x600, 1024x768 нукталарни беради.

2.Экранни график ҳолатга ўтказиш

Оддий матнли шолатдан график шолатга ўтиш учун Graph модулининг **InitGraph** просеdurаси ишлатилади.

InitGraph(Gd,Gm,Path);

бу ерда Gd -драйвер номери; Gm -режим номери;

Path -керакли драйверга йўл кўрсатади.

Агар Path="" бўш бўлса драйверни жорий котологдан излайди. Агар Gd=0 бўлса, керакли драйверни ўзи автоматик равишда танлайди, яъни Gd=Detect. Detect нолга тенг параметр. Graph - модулини чақириш дастур бош қисмида берилади, яъни

uses

Graph;

График режимни олдинги ҳолатига қайтариш учун, яъни ёпиш учун **CloseGraph;** просеdurаси ишлатилади.

3.Graph модулининг процедура ва функциялари

Экранда ҳар бир нуқта ўз координатасига эга. Координата боши экраннинг юқори чап бурчаги бўлиб, у (0,0) дан бошланади. Нуқталар координатаси пастга ва ўнга қараб ўсиб боради.

Graph – модулининг процедура ва функцияларини кўриб чиқамиз:

PutPixel(x,y,color); -процедура экранда координатаси (x,y) бўлган нуқтани тасвирлайди. Color параметри шу нуқтага ранг беради.

Мисол: PutPixel(100,120,Red); бу ерда экраннинг (100,120) нуқта қизил рангга тасвирланади. Red=4 нуқтага қизил ранг беради.

GetPixel(x,y); -функцияси нуқта рангини аниқлайди.

Мисол: Col:=GetPixel(50,80); бу ерда Col параметрига нуқта ранги юборилади.

Line(x1,y1,x2,y2); -процедура координаталари (x1,y1) ва (x2,y2) бўлган кесмани чизади.

Circle(x,y,Radius); - маркази (x,y) нуқта бўлган ва радиуси Radius бўлган айлана чизади.

Rectangle(x1,y1,x2,y2); -процедура тўғри тўртбурчак чизади.

(x1,y1) -юқори чап бурчак координатаси;

(x2,y2) -пастки ўнг бурчак координатаси.

SetColor(Color); -процедура расмга ранг беради. Color ранг номери.

Қуйидаги ранглар ишлатилади:

Ранг номери	Ранг	Ранг номери	Ранг
Bleck=0	қора	Blue=1	кўк
Green=2	яшил	Cyan=3	Брюзаранг
Red=4	қизил	Magenta=5	малинаранг
Brown=6	жигарранг	LightGray=7	оч кулранг
DarkGray=8	тўқ кулранг	LightBlue=9	оч ҳаворанг
LightGreen=10	оч яшил	LightCyan=11	оч брюзаранг
LightRed=12	оч қизил	LightGray=13	оч малинаранг
Yellow=14	сарик	White=15	оқ

Мисол 2.

Қуйидаги дастур тўғри тўртбурчак чизади.

```

Program TT;
Uses Graph;
Var gd,gm: Integer;
Begin
    gd:=detect; InitGraph(gd,gm,'');
    Rectangle(30,30,120,120); Readln;
    CloseGraph;
End.

```

Bar(x1,y1,x2,y2); -рангли ёки штрихланган тўғри тўртбурчак чизиш;
Bar3d(x1,y1,x2,y2,depth,top); -рангли ёки штрихланган паралелопепед чизиш;

FillEllipse(x,y,xradius,yradius); -рангли ёки штрихланган эллипс чизиш;

SetFillStyle(Style,Color); -штрих ва ранг бериш. Бу ерда Style штрих, Color ранг танловчи ўзгармас параметр.

Style ўзгармас параметр бўлиб, у ҳар хил штрихлар билан фигураларни тўлдиради. Улар қуйидагилардир:

```

Const
    EmptyFill=0;      { фон ранги }
    SolidFill=1;      { жорий рангни беради }
    LineFill=2;       { қалин горизантал чизик }
    StstashFill=3;    { ингичка қийшиқ чизик }
    StoshFill=4;      { қалин қийшиқ чизик }
    BkStashFill=5;    { қалин қийшиқ чизик }
    LtstashFill=6;    { қийшиқ йўл-йўл чизик }
    HatchFill=7;      { катакчалар билан тўлдириш }
    XhatchFill=8;     { қийшиқ катак билан тўлдириш }
    InterLeaveFill=9;  { қийшиқ штрих чизик }
    WideDotFill=10;   { кам нуқталар билан тўлдириш }
    CloseDotFill=11;  { бўлак-бўлак нуқталар }
    UseFill=12;       { фойдвланувчи штрихи }

```

График режимда матн ёзиш учун қуйидаги процедуралар ишлатилади.

SetTextStyle(Font, Detection, Size); -керакли шрифтни ишга туширади.

Бу ерда

Font - шрифтни танлаш;
 Detection - ёзув йўналишини белгилаш;
 Size - шрифт ўлчамини танлаш.

Шрифт ва матн ёзуви йўналиши қуйидаги ўзгармас билан аниқланади.

```

Const
    { шрифт }
    DefaultFont=0;   { стандарт шрифт }
    TriplexFont=1;   { векторли шрифт }
    { текст йўналиши }
    HarizDir=0;      { чапдан ўнга }

```

VertDir=1; { пацдан юқорига }
OutTextXY(x,y,TextString); -(x,y) координатали нуқтадан TextString
номли матн қаторини киритади.

МАЪРУЗА № 9

МОДУЛЛАР ВА МАЪЛУМОТЛАР ФАЙЛЛАРИ

Р е ж а:

- 1.Кириш.
- 2.Модул тушунчаси.
- 3.Паскал библиотека модуллари.
- 4.Модул ташкил қилиш.
- 5.Ўзгарувчилар қийматларини файлдан ўқиш ва файлга ёзиш.
- 6.Маъруза буйича саволлар.

А д а б и ё т л а р

- 1.Культин Н.Б. Программирование в Turbo Pascal 7.0 и Delphi. “Санкт-Петербург”, 1997г.

1.Кириш

Турбо Паскаль системасида жуда кўп махсус тайёр процедура ва функциялар мавжудки улар ҳар қайси ўз вазифасига эга бўлиб унга библиотека модуллари дейилади. Ҳар бир библиотека функция ва процедуралардан ташкил топган бўлиб маълум бир турдаги масалани ечишга мўлжалланган бўлади.

2.Модул тушунчаси

Модул деб процедура ва функциянинг алоҳида компиляция қилиниб махсус .tpr кенгайтмалар файли шаклидаги ифодаланган дастурсига айтилади. Модулдан ихтиёрий дастур ичида фойдаланиш мумкин. Модулдан фойдаланиш яъни уни активлаштириш учун дастурнинг бош қисмида қуйидагиларни келтириш зарур.

Uses <Модул1 номи, модул2 номи, . . . ,модулn номи>;

Мисол: Program SS;
 Uses Crt,Graph;

Турбо-Паскаль системасида ҳар бир фойдаланувчи ўз модулини яратиши учун яратиладиган модул структурасини қуйидагича ташкил қилиш зарур.

Unit <модул номи>;

Interface

.....
{Интерфейс қисм- очик (ёзувлар) қисми}

.....
Implementation

.....
{Ёпиқ (ёзувлар) қисми}

.....
Begin

.....
{Модулнинг асосий қисми}

.....
End.

Бу ерда **Unit** - модулнинг сарлавҳаси;

Interface - модулнинг интерфейси, яъни дастур ва бошқа модуллар учун очик (кўринарли) қисмининг бошланишини билдиради. Бу қисмда ўзгармаслар, катталиклар типлари, процедура ва функциялар аникланиб курсатилган бўлади, лекин уларнинг бутун кўриниши кейинги ёпиқ қисмда берилади.

Implementation - модулнинг дастур ва модуллар учун ёпиқ, яъни кўринмайдиган қисмининг бошланишини билдиради. Бу ерда интерфейс қисмда аниқланган процедура ва функциялар яна бир марта кўрсатилиши шарт (уларнинг сарлавҳалари бир хил бўлиши керак).

Инициализация қисми **Begin** ёзувидан кейин бошланади, агар бу қисм мавжуд бўлмаса **Begin** ҳам бўлмайди. Бу қисмда бошқарувни асосий дастурга ўткизишгача қадар бажарилиши керак булган операторлар руйхати жойлашади.

3.Паскал библиотека модуллари

Турбо Паскаль системасида қуйидаги библиотека модуллари мажуд:

System - стандарт процедура ва функцияларни ўз ичига олган бўлиб, бу модул автоматик равишда активлаштирилган бўлади.

Dos - Ms Dos операцион система имкониятларидан фойдаланувчи процедура ва функцияларни ўз ичига олган.

Crt - монитор экрани ва клавиатура билан ишлаш имкониятини яратувчи процедуралар тўпламини ўз ичига олган.

Graph - ҳар хил монитор видеоадаптерларини қуллаган ҳолда компьютер график имкониятларидан фойдаланувчи процедуралар тўпламини ўз ичига олади.

Printer - принтер билан ишловчи кичик модул.

4.Модул ташкил қилиш

Турбо Паскал дастурлаш воситаси ёрдамида ҳар бир дастур тузувчи ўз модулини яратиш имконига эга. Бунинг учун кerkли дастур қисмини

юқорида кўрсатилган структура кўринишда ёзиш ва уни компилятсия қилиш керак.

Мисол тариқасида икки соннинг энг катта ва энг кичигини топиш модулини яратиш дастурини қараймиз. Қуйидаги дастур Min(x,y) ва Max(x,y) функцияларини ўз ичига олган.

```
Unit Study;  
Interface {Интерфейс қисм }  
    Function Min(x,y:Integer):Integer;  
    Function Max(x,y:Integer):Integer;  
Implementation {Ёпик қисм}  
    Function Min(x,y:Integer):Integer;  
Begin  
    If x<=y Then Min:=x Else Min:=y;  
End;  
    Function Max(x,y:Integer):Integer;  
Begin  
    If x>=y Then Max:=x Else Max:=y;  
End;  
{Инисталайтсия қисм мавжуд эмас}  
End.
```

Бу модул компиляция қилиниб Stadu.tpu файл номга эга бўлиши керак. Ундан дастурда фойдаланиш учун дастур бош қисмида Uses Stadu қаторини ёзиш керак бўлади.

5.Ўзгарувчилар қийматларини файлдан ўқиш ва файлга ёзиш

Киритиладиган ва чиқариладиган маълумотлар сони кўп миқдорда бўлса уларни файлда сақлаш дастурда қулайлик туғдиради. Бу маълумотлар оддий матн (текст) файлларида сақланади. Файл ўзгарувчиси дастур бош қисмида эълон қилинади, яъни

```
Type f=text;  
Var fx:f;
```

бу ерда f -файл типи, оддий матн файлни билдиради; fx-файл ўзгарувчиси.

Керакли файлдан маълумотларни ўқишга тайёрлаш учун Assign ва Reset функциялари ишлатилади.

Assign-файл ўзгарувчиси билан ососий файл орасида алоқа ўрнатади.

```
Assign (fx,'c:\a\f1.txt');
```

Reset-файлни топиб уни ишга тайёрлайди. **Reset (fx);**

Бу ерда fx- файл ўзгарувчиси; 'c:\a\f1.txt'-с: дискнинг A каталогидаги f1.txt файлдан ўқишни билдиради.

Файлдаги маълумотларни ўқиш учун Read функцияси ишлатилади.

Read (<файл ўзгарувчиси>, <ўзгарувчилар,массивлар>);

Мисол. Read (fx, x,y,z,A[i],B[I]);

Файлдан ўзгарувчилар ёки массивлар қийматларини ўқиб бўлгандан кейин файл ёпилади. Файлни ёпиш қуйидаги функция ёрдамида бажарилади.

Close (файл ўзгарувчиси); Мисол. Close (fx);

Мисол 1. C: дискдаги AA каталогдаги AB файлдаги маълумотларни ўқиб A ва B массивларга жойлаштиринг. Файл маълумотлари структураси қуйидагича.

15.2 20.5

20.1 25.5

.....

яъни файл структураси икки устундан иборат маълумотлар тўпламидан иборат.

Program FAB;

Type f=text;

Var

A,B: Array[1..100] of Real; m: Integer;

fxz: f;

Begin

Assign(fxz,'c:\AA\AB.txt'); Reset(fxz);

m:=0;

While not eof(fxz) do

Begin m:=m+1; Readln(fxz,A[m],B[m]); End; Close (fxz);

End.

Массив қийматларини бирон матн файлига ёзиб қўйиш учун Assign, Rewrite, Append, Write ва Close функциялари ишлатилади.

Assign- файл ўзгарувчиси билан асосий файл ўртасида аълоқа ўрнатади ва у қуйидагича ёзилади.

Assign(файл ўзгарувчиси, дискдаги файл жойи ва номи);

Append- Файлдан ёзиш учун жой тайёрлайли.

Append(файл ўзгарувчиси);

Write- ўзгарувчи қийматини файл ўзгарувчисига узатади ва файлга ёзади.

Write(файл ўзгарувчиси, ўзгарувчилар);

Close- очилган файлни ёпади. **Close (файл ўзгарувчиси);**

Мисол 2. Юқоридаги C: дискдаги AA каталогдаги AB файлдаги маълумотларни A ва B массивларга ўқиб шу массив мос элементларини қўшиб C массивни ташкил қилинг ва A,B,C массивларини ABC номли файлга ёзинг.

Файл маълумотлари структураси қуйидагича бўлсин.

i	A	B	C
1	15.2	20.5	45.7
2	20.1	25.5	45.6
.....			

Program FABC;

Type f=text;

Var A,B,C: Array[1..100] of Real; i,m: Integer;

fax: f;

```
Begin  
  Assign(fax,'c:\AA\AB.txt'); Reset(fax);  
  m:=0;  
  While not eof(fax) do  
    Begin    m:=m+1;  Readln(fax,A[m],B[m]); End;  
    Close (fax);  
  Assign(fax,'c:\AA\ABC.txt'); Rewrite(fax); Append(fax);  
  For i:=to m do  
    Begin Write(fax,i,A[i],b[i],c[i]); Writeln(fax); End;  
  Close(fax);  
End.
```