

**“Алгоритмик тиллар ва програм-  
малаштириш” фанидан**

**“Алгоритмик тиллар ва програм-  
малаштириш” фанидан**

# **МАЪРУЗА МАТНЛАРИ**

# 1 - М А О Р У З А

## Кириш: Алгоритмик тиллар ва дастурни асослари.

### Режа.

1. Ўйсызча алгоритм тушинчаси тшърисида.
2. Дастурлаштириш тиллари. Компиляция ва интерпритация.
3. Програмлилаштириш тиллари класификацияси.
4. Алгоритм тузиш ва дастурлаштиришда кузатиладиган хатоликлар тшърисида.

**Таянч сшъзлар:** *Алгоритм, блоксхема, дастурлаштириш, алгоритмик тиллар, компиляция, интерпритация, дастурлаштириш тиллари класификацияси, хатоликлар, Алгоритм формаларини турлари, процедура.*

Маолумки I-курсни I-чи ва II-чи семестрлари давомида “Информатика” фани шъитилиб, унда махсус маорузада алгоритм ва дастур тушунчалари ойсызча атрофлиги шърганилган. Энди эса биз бу тушунчаларни тшъла анализ йиламиз.

Алгоритм дейишганда - йшъйилган масалани ечимига олиб борувчи аниы ва тшъла операциялар кетма кетлиги тушинилади. Ихтиёрий масалани хох у оддий шъсоб иши, ёки мураккаб жараёни бошъйаришга йаратилган бшълишига йарамасдан уни компрютерда ечиш учун йуйдаги кетма-кетликда иш кшърилади:

1. Ўшъйилган масалани анализ йилиниб йандай катталиклар киритиб, йандай натижалар олинишини аниыланди.
2. Масалани ечимига олиб борувчи операциялар кетма-кетлигини танланади ёки ишлаб чиыилади, йони масалага алгоритм тузилади.
3. Тузилган алгоритмга бирор алгоритмик тил билан дастур ёзилади ва компрютерга киритилади.
4. Компрютерда дастурни тшъри ёки нотшъри ишлашини аниылаш учун синов катталиклар киритилиб, дастур отладка (созланади) йилинади.
5. Дастур тшъри ишлаётганлигига ишонч шъсил йилингандан кейин берилганлар киритилиб натижа олинади.

Булардан кшъриниб турибдики, алгоритм дастурлаштиришда алоъида босыични ташкил этади.

Алгоритмларни бир неча формаларда тузилиши мумкин, Бу формалар йуйдагилар:

- Ёзишмалар кшъринишида;
- Формула кшъринишида;
- Жадвал шаклида;
- График кшъринишида;
- Блок схема шаклида.

Бозирда энг кенг ышлланиладиган формаси блок схема шаклидир. Бу формада алгоритмдаги хар бир тур операция щзини шартли белгисига эга (Бу “Информатика” курсида кщриб чиыилган).

Дастур бу машина буйруулар тшплари бщлиб, улар ёрдамида компютер тавсия этилган алгоритмни бажаради. Бошья сщз билан айтилганда дастур бу алгоритмни машинага берилиш формасидир.

Компютерлар дастлаб ишлаб чиыарилган давирда дастурлар машина кодлари ёрдамида ёзилган. Масалан: 14 0142 2146 3541.

Бу ерда 14-сонни операция коди тисобланиб, икки сонни ыщшишни билдиради. Кейинги келтирилган сонлар хотира ячейкалари бщлиб, улар икки сонни ыаердан олишини ва натижа ыайси ячейкага ёзилишини билдиради. Демак келтирилган сонлар кетма кетлиги машина кодида ёзилган алгоритмни ифодалайди. Алгоритм ыуйидагига: 0142 адресли ячейкадан биринчи сонни, 2146 адресли ячейкадан иккинчи сонни олиб ыщшиб 3541 чи адресли ячейкага натижани ёзиш.

Мутахассислар бу турда дастур анча ыййин ва машаыятли эканлигини, ёзишда хатоликлар йщл ыщйиш эътимолига катталигини билиниб бу машина-кодларини шартли белгилар асосида белгилаш системасини яратдилар.

Бу системани алгоритмик тил деб аталди. У ёрдамида шартли равишда машина буйруулари белгиланади. Ёзилган дастурни бир махсус дастур ёрдамида машина кодларига щтказиш зарур бщлади.

Дастурлаштириш тилидаги буйрууларни тилни операторлар деб аталади. Машина кодига щтказувчи махсус дастур асосан икки турда бщлади:

Дастур - комплятор (комплятор - ташкил этувчи, йиувчи) - алгоритмик тилда ёзилган дастурни машина кодига айлантрииб, юкланувчи сифатида магнит дискга саылаб ыщяди.

Шундан кейин дастур текистига боълиы бщлган холда бажаришга берилиши мумкин. Эски адабиётларда дастур компляторларни транслятор (таржимонлар) деб аталган.

Дастур интерпретатор-дастур тексти билан биргаликда ишлайди. У дастур матинини хар бир ёзувни операторини хал ыилиб (интерпретация ыилиб) дархол бажаришга беради. Машина тилига щтказилган файл тосил ыилинмайди. Шунинг учун интерпрота билан ишлайдиган дастурлар жуда секин ишлайди. Унда дастурдаги ыайтарувчи операциялар хар сафар ыайта интерпретация ыилинади.

Дастурлаштириш тилларини аниы классификацияси йщы. Уларни тахминан ыуйидагиларга бщлиш мумкин:

1. Асемблер тили (ыуйи дастурлаштириш тили).
2. Юыори даражадаги тиллар.
3. Маолумотлар баозаси билан ишловчи тиллар.

Асемблер тили хар бир буйруыи (оператори) битта машина буйруыига мос келади. Шунинг учун уни гоъида машина тили тям деб юритадилар. У ёрдамида самарадорлиги юыори бщлган дастурларни тузиш мумкин, лекин уларни тузиш катта меънат ва квалификация талаб ыилади.

Юъори даражадаги тилларга мисол ыилиб BASIC, Фортран, Фокалр, Паскалр, Си ва бошыаларни киритиш мумкин. Улар шзларини имкониятлари ва мшлжалланган сохаларига ыараб фарыланадилар. Масалан фортран инженер-техниклар учун мшлжалланган бшлса Cobol - комерек ыисоб ишларига мшжалланган. Паскалр ва Си универсал дастурлаштириш тилларидир.

Алгоритмик тилларда дастур ёзишда асосан икки хил катталиклар кузатилади: а) Синтаксис хатоликлар-дастур ёзишда орфографик ва схематик хатоликларга йшл ышйилиши мумкин. Бу хатоликларни компрютерга киритиш давомида ёки бажарилишга берилганда алгоритмик тил интерпретатори ёки компилятор кшрсатади уларни тузатиш осон. б) Логик хатоликлар - дастур компрютерга киритилиб бажаришга берилганда натижа чиыади. Лекин олинган натижа ыаыиыатга тшъри келмайди. Бундай хатоликларни аниылаш ва тузатиш анча меънат ва малака талаб ыилади.

**Назорат саволлар.**

1. Ёисыача алгоритм тушинчаси тўғрисида таъриф бериш.
2. Дастурлаштириш деганда нимани тушинаси?
3. Дастурлаштириш тиллари деганда нимани тушинаси?
4. Компиляцияга таъриф бериш.
5. Интерпритация нима
6. Программлаштириш тиллари классификацияси деганда нимани тушунаси.
7. Алгоритм тузишни кўрсатиш.
8. Алгоритм тузиш дастурлаштиришда кузатиладиган хатоликлар тўғрисида маълумот бериш.
9. Қандай алгоритм формалари бор?
10. Процедура деганда нимани тушинаси?

## 2 - М А О Р У З А

### Компьютерларда информацияларни ифодалаш.

#### Режа.

1. Санови системалари.
2. Бир санови системасидан бошья санови системасига штказиш.
3. Иккилик сон битлар ва байтлар.
4. 16 лик сон.

**Таянч иборалар:** Санови система, бир санови системасидан бошья санови системасига штказиш, санови системаси турлари, позицион санови системаси, позицион бщлмаган санови системаси, иккилик санови системаси, саккизлик санови системаси, щнлик санови системаси, щнолтилик санови системаси, битлар, байтлар.

Бирор бир миьдорни баъолаш учун санови системалари киритилган. Улар иккига бщлинади позицион, позицион бщлмаган.

Агар раьамни сондан тутган щрни аъамиятга эга бщлса бундай системани позицион, акс холда позицион бщлмаган санови системалари дейилади.

Позицион санови системаларига мисол ьилиб иккилик, саккизлик, щнолтилик санови системалари келтириш мумкин. Р асосга эга бщлган ихтиёрий N сонини умумий кщринишда ьуйидагича ёзиш мумкин:

$$a_R \cdot P^R + a_{R-1} \cdot P^{R-1} + \dots + a_1 \cdot P + a_0 \quad (1)$$

Сонни щнлик санови системасидан бошья бир санови системасига штказиш учун сон кетма кет равишда щтиладиган санови системаси ас- осига бщлинади натижа ва ьолдиьлар кетма-кет равишда тескари ёзилиб щыилади.

Масалан:  $247_{10} = ?_8$

$$\begin{array}{r|l} 247 & 8 \\ \hline 240 & 30 \\ \hline 7 & 24 \\ \hline & 6 \end{array}$$

демак  $247_{10} = 367_8$

Бир санови системасидан щнлик санови системасига щтиш учун (1) формуладан фойдаланилади ва ьисобланади.

Масалан:  $367_8 = ?_{10}$

$$3 \cdot 8^2 + 6 \cdot 8^1 + 7 \cdot 8^0 = 3 \cdot 64 + 6 \cdot 8 + 7 \cdot 1 = 192 + 48 + 7 = 247, \quad \text{демак } 367_8 = 247_{10}$$

Компьютерда информациялар иккилик сонларда ифодаланади. Чунки компьютернинг ташкил этган физик ьурилмалар асосан турьун ьолатида икки ьолат бщлиши мумкин. Масалан тоь бор ёки йщы, калит очий ёки

берк. Биринчи ʋолатларини “1” иккинчи ʋолатларини “0” деб олинади. “0” ва “1” ларни комбинациясидан ихтиёрий сонни ифодалаш мумкин.

Сонларни “0” ва “1” раѡамлари ёрдамида ифодаланиши иккилик саноѡ системасида ифодаланиши дейилади. Раѡамларни иккилик раѡамлар ёки иккилик разрядлар дейилади.

Щнлик саноѡ системасидаги раѡамларни иккилик сонлар билан ифодаланиши:

0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001

Битта иккилик информация “0” ёки “1” 1 битни ташкил этади “бит” сщзи иккилик системасидаги раѡам деган маонони беради.

Байт 8 битдан иборат бщлиб, битта сон ёки битта белгини (символни) ифодалайди. Битта байт билан 256 символни ифодалаш мумкин, чунки 8 битда 256 та сонни комбинация ѡилиш мумкин. 256 та коддан иборат жадвал мавжуд. Бу жадвални ASCII- кодлари дейилади.

Щн олти разряддан иборат информацияни машина сщзи, 32 разрядлигини иккиланган сщз дейилади. Байтни энг кичик информация ѡажмий бирлиги дейилади. Шунинг учун фойдаланишга щнга каррали бщлган бирликлар Кбайт, Мбайт, Гбайт киритилган.

1 Кбайт=1024 байт, 1Мбайт=1024 Кбайт, 1 Гбайт=1024 Мбайт

Адабиётларда 16 лик саноѡ системаси кенг ѡщлланилмоѡда. Чунки у иккилик сонларни ѡисѡа кщринишда ифодалаш имкониятини беради. Хар бир тщрта иккилик сон разряди битта 10лик сон раѡами билан белгиланади.

Масалан 1000 10 10 11110 иккилик сонни чап томондан бошлаб тщртадан ѡилиб разрядга бщлинса.

115E га тенг 16 лик сон ѡилиб чиѡади.

10 лик	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2 лик	0	01	10	11	100	101	110	111	1001	1010	1011							
16 лик	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11

### **Назорат саволлар.**

1. Саноы системага таориф беринг
2. Бир саноы системасидан бошыа саноы системасига щтказиш учун нима ыилади?
3. Ыандай саноы системаси турлари мавжуд?
4. Позцион саноы системасига таориф беринг.
5. Позцион бщлмаган саноы системасига таориф беринг.
6. Иккилик саноы системаси нима?
7. Саккизлик саноы системаси нима?
8. Щнлик саноы системаси нима?
9. Щнолтилик саноы системаси нима?
10. Битлар ва байтлар.





### 3 - М А О Р У З А

#### Персонал компютерлар асосий ыурилмалари.

##### Режа.

1. Система блоки.
2. Микропроцессор.
3. Оператив хотира.
4. Ташыи ыурилмалари.

**Таянч иборалар:** Система блоки, микропроцессор, оператив хотира, ташыи ыурилмалари, доимийхотира, сичыон, модем, монитор, клавиатура, сканер.

Система блоки компютерларни асосий ыурилмаларни шз ичига олади. Улар ыуйидагилар:

Микропроцессор - компютерда асосий ыурилмаси бшлиб, у компютер ыурилмалари шртасидаги алобани таоминлаб уларни иш фаолиятини бошыаради.

Компютер таркибига кирувчи ички хотира икки хил бшлади: доимий ва оператив хотира. Биринчи типа хотирадиги информация завод томонидан ёзилган бшлиб иш жараёнида шзгармайди. Оператив хотира шзида дастурларни, берилганларни, оралиы натижаларни саылаб ундаги информация иш жараёни шзгариб туради.

Хотира ыурилмасини кичик-кичик уячаларидан иборат деб тасаввур ыилинади. Хар бир уяча 1 та байт информацияни саылай олади. Уларни хар бир шз тартиб номерига эга бшлиб бу номерни ячейка адреси деб аталади.

Компютерга ышйилган микропроцессорлар тури хажимдаги оператив хотирани кштара олиши мумкин.

Масалан: Intel-8088 ва Intel-8086 микропроцессорлари 1 Мбайтгача хотирани, Intel-80386 микропроцессор эса 16 Мбайтгача кштара олади.

Хотира шартли равишда блоklarга олинади. Битта блок 64 К байтни ташкил этиб сегмент деб аталади.

Микропроцессор магнит дисклардаги информацияларга бевосита мурожат эта олмайди. У улар билан иш кшриш учун уларни оператив хотирага юклаб олади.

Персоналр компютерларни видеосистемаси икки ыисмдан иборат. Улар дисплей ва адаптердир. Адаптер махсус ыурилма бшлиб, система блокига шрнатилиб у видео сигналларни ишлаб чиывчи логик схемалардан тузилган. Уни шз хотираси ьажми бшлиб дисплейга (мониторга) узатиладиган информациядан шз хотирасидан узатади. Видео хотирадаги информация шзгариши билан экрандаги информациялар шзгаради. Марказий процессор шша хотирага ьам оператив хотира каби тез мурожат ыила олади. Шунинг учун ыисыа ваьт ичида унча мураккаб бшлмаган тасвирни ьосил ыилиш мумкин.

Монитор электрон нур трубасидан иборат бўлиб ёювчи схемаси мавжуд. Электрон нур 1/50 секундда экранда тўтла ҳаракат бўла олади. Ҳар ыандай тасвир экранда пикселлар тўтплами (нуыталар тўтплами) билан ўосил ыилинади. Турли маркадаги мониторлар экрандаги турли сондаги нуыталарга эга шунинг учун уларни битта характеристикаси экранда нечта нуыта ўосил ыила олишидир.

Бундан ташыари монитори рангли, рангсиз, экран диаметри билан ўам характерланади.

Масалан EGA мониторлари имконияти 640x350 нуыта, VGA мониторлари 640x480 нуытагача.

Видеоадаптер таркибига махсус монитор контролери микропроцессори кириб, у монитorni бошыариш учун хизмат ыилади. У марказий процессорни баози элементар ишлардан озод ыилади.

Клавиатура ёрдамида компютерга буйруылар, информациялар киритилади. Уни киритиш ыурилмаси деб аталади. У ёрдамида матнли информациялар киритилишида ташыари компютер бошыарилади.

Клавиатуралар 101 ва 105 клавишлари мавжуд. 105 лигига баози махсус клавишлар ыўйишган. Ундаги клавишларни шартли равишда ыуйдаги гуруҳларга бўшлинади:

1. Ҳарф, раыам ва белгиларни киритиш клавишлари;
2. Курсорни бошыариш клавишлари;
3. Бир клавишни бир неча белги теришини таоминловчи ёрдамчи клавишлар;
4. Махсус клавишлар Del, PrintScreen ва бошыалар;
5. Функционалар клавишлар;

Функционал клавишларни вазифасини компютерга юкланган дастурлар белгилайди. Шунинг учун уларни дастур бошыаруви клавишлар дейилади.

Магнит дискларга тўтловчи ыурилмалар асосан икки хил бўлиб ыаттиы магнит дисклар ва эгилувчи магнит дискларга тўтловчиларга бўшлинади. Биринчи тур ыурилмаларни адабиётларда винчестер деб аталади. Эгулувчи магнит дисклар 5,25 дюйм ва 3,5 дюймли бўлади.

Одатда 5,25 дюймли дисклар 360 Кбайт ёки 1,2 Мбайт, 3,5 дюймли 720 Кбайт ва 1,44 Мбайтга форматланиши мумкин.

Ўозирги пайтда асосан уч хил типдаги принтерлар ишлатилмоыда.

1. Матрицали - яони ҳар бир символ маолум сондаги игналар билан лентага урилиб ўосил ыилинади. Уларни ишлаш принципи содда ва ишончлигига ортиы, лекин уларни ёзиш сифати ва тезлиги шнга юыори эмас.
2. Сепувчи (сийхли) принтерлар, ёзиш сифати юыори бўшлиши билан бирга рангли тасвир чиыара олади.
3. Лазерли принтерлар, ёзиш сифати ва тезлиги юыори.

Сичыонга 1964 йил Э.Дуглас томонидан яратилган бўлиб компютерга информацияни тезроы киритиб осонроы бошыариш учун хизмат ыилади. Ўозирда ыўлланилаётган баози система ва амалий

дастурлардан сичыонларсиз фойдаланиш ыййин. У щзини харакати ва клавишлари ёрдамида компрютерга сигнал юборади.

### **Назорат саволлари.**

1. Система блоки нима?
2. Микропроцессор деганда нимани тушинасиз?
3. Оператив хотира нима?
4. Ташыи ыурилмалари деганда нимани тушинасиз?
5. Доимийхотира деганда нимани тушинасиз?
6. Сичыонча нима вазифани бажаради?
7. Модем ыандай вазифани бажаради?
8. Монитор турларини айтинг.
9. Клавиатура ыандай вазифани бажаради?
10. Сканерни вазифаси?

## 4 - М А О Р У З А

### Машина кодлари ва тўхталишлар.

#### Режа.

1. Машина кодлари.
2. Тўхталишлар.
3. Дастурлаштириш тиллари ва трансляторлар.

**Таянч иборалар:** *Машина кодлари, тўхталишлар, дастурлаштириш тиллари, трансляторлар, микропроцессор, оператив хотира, порт, транслятор, компилятор, ассемблер транслятори*

Юзорида кўриб ўтдик, микропроцессорлар компютер ёзурилмалар шртасидаги алоҳида таоминлаб уларни иш фаолиятини бошқариб, компютерда масала ечилишини ташкил этувчи мураккаб ёзурилмадир.

Лекин олиб ёралганда у оддий, аниқ сондаги буйруулар тўпламини бажара олувчи машинадан иборат.

Масалан ёзувдаги каби буйруулар:

- Оператив хотирани бир жойидан бир сонни ол, иккинчи жойидан иккинчи сонни ол ва уларни ёшқиб хотирани учинчи жойига жойла;
- Оператив хотирани бир жойдаги информациядан иккинчи жойга нусха ол;
- Оператив хотирани бирор жойдаги информацияни портга узат (порт компютер билан ташви ёзурилма шртасидаги алоҳида таоминловчи ёзурилма);
- Ёки портдаги информацияни хотирани аниқ адресли жойига ёз: ва шулар қабилар.

Микропроцессор фаёат унга узатилаётган буйруулар кетма-кетлигини келтирилган тартибда бажаради холос. Буйруулар бажарилиши тартиби оператив хотирага ёзилган ёшқиб, ёйси буйруудан кейин ёандай операция бажарилиш кераклигини аниқ ёзилган.

Микропроцессор бир буйрууни бажараётганда уни адресини махсус регистрда сақлаб туради. Бақариб ёшқилиши билан регистрдаги адрес автоматик равишда биттага ортиб микропроцессор навбатдаги буйрууни бақаришга тайёр ёшқлади.

Иш давомида микропроцессорни буйрууларни бақариш кетма-кетлигини шзгарувчи буйруулар ҳам мавжуд. Уларни шартли ёки шартсиз

штиш буйруылари дейилади. Микропроцессорларни буйруыларни бажаришда маолум сондаги олдинги ёки орыадаги буйрууни бажаришга штишини таоминлайдилар.

Шундай ыилиб микропроцессор оператив хотирага ёзилган буйруулар (инструкциялар) кетма-кетлигини бажаради. Бу кетма- кетликини машина кодлари дейилади.

Машина кодлари аниы маонога эга бщлиб буйруулар кетма-кетлигидан иборат бщлсада уларни бир кщришда инсонлар тушиниши жуда ыйин. Масалан мисол сифатида 16 лик системасида ёзилган буйруулар ыуйидагича бщлиши мумкин:

```
BC 232 EE 83 FO 87311 B 8 CD 01 E8
OC 130 E 283 FE 82342 BA CD 01 E 8
B 1 D 30 EE 93 FO 87 A 13 B 8 CDB 1 E 8
```

Кщриниб турибдики машина кодларини билиш дастурчи учун унчалик зарур эмас. У дастурлаштириш тилларини билиши етарлидир. Дастурлаштириш тиллари ёрдамида осон дастур тузади. Ёзилган дастур компрютерда бажарилиши учун уни машина кодига штказиш зарур. Бу ишни эса махсус интерпретация ыилувчи ёки трансляция ыилувчи дастурлар бажаради.

Микропроцессор буйрууларни бажариши жараёнида кутилмаган ходисаларга эотибор бериши керак. Чунки уларни олдиндан айтиб бщлмайди. Лекин клавиатурани бирор клавиши босилганда ёки сичюн ёрдамида бирор сигнал келгандан микропроцессорни харакат механизми олдиндан белгилаб ыщйилиши керак.

Уни бажариш тезлиги шу даражада катта, битта клавишни босгунга ыадар бир неча минг операцияни бажариб ыщя олади.

Микропроцессор бирор сигнал келиши билан ходисани кутиши, агар хеч нарса бщлмаса щз ишини давом етказиш учун ыанча ваыт кутиши белгиланиши зарур. Бу келтирилган ьолат прервания (ваытинча тщхтатиш) деб аталиб, у микропроцессорга ташыи воыеага эотибор беришини таоминлайди.

Бундай тщхталишлар клавиатура ёки винчестер ёки бошыа ыурилмадан бщлиши мумкин.

Тщхталишларни ыайта ишлашда микропроцессор олдиндан белгилаб ыщйилган тщхталишлар жадвалига мурожат ыилиб у ерда кщрсатилган инструкцияларни бажарилади.

Транслятор ёки компилятор - бу махсус дастур бщлиб, алгоритмик тилда ёзилган дастурни (инструкцияни) машина кодига штказади.

Бу жараённи компиляция ёки транслятсия деб аталади.

Энг ыуйи дастурлаштириш тили ассемблер бщлиб у машина кодига яын. Бу тилда процессор буйруылари махсус ыисыа белгилашлар олган. Масалан ADD - ыщшиш, MUL - кщпайтириш, JMP - штиш (ирыш).

Ассемблер транслятори хар бир буйрууни битта машина кодига айлантиради. Шунинг учун ассемблерда дастур ёзган дастурчи машинани

ѡамма имкониятларидан тѡтла фойдаланиш имкониятига эга бѡлади. Лекин у ѡамма ѡурилмаларни тѡтласига эотиборга олиши, бу эса жуда катта малака ва меънат талаб ѡилиши аниѡ.

Шунинг учун катта дастурларни ѡѡюри даражадаги дастурлаштириш тилида ёзиш маѡсадга мувофиѡ. Алгоритмик тилларда ёзилга битта буйруѡ машина кодида бир неча буйруѡлардан иборат бѡлиши мумкин. Алгоритмик тилларда дастур ёзишда хар бир ѡурилмани хар томонлама ѡисобга олиш шарт эмас. Программист асосий эотиборини дастурни ѡурилмаларда бажарилиш деталарига ѡаратмасдан масалани мохияти ва ечиш йѡлга ѡаратади.

### **Назорат саволлар.**

1. Машина кодлари деганда нимани тушинасиз?
2. Тѡхталишлар нима?
3. Дастурлаштириш тиллари деганда нимани тушинасиз?
4. Трансляторлар деганда нимани тушинасиз?
5. Микропроцессор ѡандай вазифани бажаради?
6. Оператив хотира нима?
7. Порт нима?
8. Транцилятор деганда нимани тушинасиз?
9. Компилятор деганда нимани тушинасиз?
10. Ассемблер транслятори нима?

## 5 - М А О Р У З А

### Операцион система ва файллар билан ишлаш.

#### Режа.

1. MS-DOS операцион система.
2. Информациялар саылаш.
3. MS-DOSнинг кобиы дастурлари.

**Таянч иборалар:** *MS-DOS операцион система, информациялар саылаш, MS-DOSнинг ыобиы дастурлари, Norton Commander системаси, файл, жорий диск,каталог,йшл,NC панеллар,магнит диск.*

Компьютерлар икки ыисмга бщлинади: аппарат ыисми ва дастур таоминоти. Дастур таоминотини асосини операцион система ташкил этади. Компьютер электр тармоьига уланиши билан автоматик тарзда операцион система оператив хотирага юкланиб ичига тушади. Унинг асосий фуқнцияси компьютер асосий ыурилмаларини иш фаолиятини бирламчи тестдан щтказиб ыурилмалар щртасидаги алоьыаларни щрнатиб информация ыабул ыилиш, ыайта ишлаш ва чиьаришга тайёрлайди.

Ъозирда асосан IBM PC компьютерларда MSDOS операцион системаси ыщлланилмоьда. MSDOS операцион системаси ишга тайёр экранлиги

C:\>

белгилаш чиьиши билан билдиради. Экранга чиьыан белгини MSDOS операцион системасининг ишга таклифи дейилади. Бу ерда C диск жорий эканлигини жорий каталог илдиз каталог эканлигини билиш мумкин.

“Информатика” курсида бу операцион система билан тщла танишганлигимиз учун уни фаьат асослари тщърисида тщхталамиз. Уни ташкил этувчи дастур модуллар BIOS, IO.sys, MSDOS.sys ва Command.com лардир. Уларни хар бир аниы функцияларини бажаради, кенг ыщлланиладиган буйруьлари:

copy, ren, rd, md, del, time, date, type ва бошыалар.



Компьютер таркибига кирувчи ташыи хотира ысобланган магнит дисклар юбюрида кщриб щтканимиздек икки хил ыаттиы ва эгулувчан дисклар. Ыаттиы диск винчестер ыажми одатда етарли даражада катта. Шунинг учун MSDOS дастурлари асосида уни бир неча логик дискларга бщлиб олинади ва хар бири алоъида дискларда сифатли фойдаланилади. Уларни номери ыуйидагича агар битта эгилувчан улщвчи ыурилма бщлса уни номи А: бщлиб, ыаттиы дисклар номери С:, D:, E: ва ыоказо бщлади.

Иш давомида актив бщлган дискни жорий диск деб аталади.

Файл дейилганда оператив хотирада ёки магнит дискда маолум жойни эгалловчи маолум номга эга бщлган информация тщплами тушунилади. Файлни ыандай номланиши ва уни кенгайтмасини файл типини кщрсатишини информатика курсида тщла щрганилган. Масалан:

- .exe, .com - бажарилувчи файллар.
- .bat - буйруыли файл.
- .c - Си тилида ёзилган дастурли файл.
- .asm - ассемблер дастурси.
- .txt - матнли файл.
- .sys - системали файл.

Каталоглар файллар гурухини бир ном билан саылаб ыщйиш учун хизмат ыилади. Каталог щз ичига каталогни (под каталогни) ыам олиши мумкин. Одатда NC система магнит дисларидаги информацияларни экранга чиыарилганда каталоглар номлари катта ыарф, файллар номлари кичик ыарф билан чиыади. Каталог ыажми кщрсатиладиган устунда [Dir ...] деган информация чиыиб, уни каталог (директория) эканлигини билдиради.

Файлларни устида операциялар бажарилади, шу файлни магнит дискларни ыайси каталогида жойлашганлигини операция буйруыи таркибида кщрсатилиши керак.

Бу кщрсатиш “\” символ орыали олиб борилиб уни йщл деб аталади. Масалан: DOS\sys\format.exe.

MSDOS буйруыларидан фойдаланишни енгиллаштириш ыулай холлатга келтириш учун кобиы дастурлар ёзилган. Улардан бири Norton Commander системасидир.

Бу системани бир неча версиялари мавжуд бщлиб, улар бир бирини имкониятларини кенгайтириб тщлдириб борган. NC ишга тушиши билан экранда уни асосий экрани ыосил бщлиб, унда NC панеллари чиыади. Чап ва щнг панелда танланган каталог ёки магнит дискдаги информациялар (файл ва каталог руйхати) кщринади. Курсор турган панелдаги каталог жорий каталог ыисобланади. Панеллар ыуйи ыисмида буйруылар сатри кщриниб кщради. Бу буйруылар сатри Ms DOS операцион система буйруыларидан фойдаланиб операциялар бажариш имкониятини беради.

NC системасидан фойдаланиб ёки буйруылар сатрида Ms DOS операцион системаси буйруылари ёрдамида файллар устида амаллар бажариш, уларни бажарилиш натижалари NC панеллари орыасида жойлашган бщлади. Уни кщриш учун NC панелларини Ctrl+O клавишаларини босиш билан олиб ыщйилади.



### **Назорат саволлар.**

1. MS-DOS операцион системаси хаёида нималарни биласиз?
2. Информациялар саылаш деганда нимани тушинасиз?
3. MS-DOS нинг ыобиы дастурлари нима?
4. Norton Commander системаси вазифаси?
5. Файлнинг вазифаси?
6. Жорий диск деганда нимани тушинасиз?
7. Каталогнинг вазифаси?
8. Йшл деганда нимани тушинасиз?
9. NC панеллар хаёида нималарни биласиз?
10. Магнит диск деганда нимани тушинасиз?

## 6 - М А О Р У З А

### Турбо-Паскалр ва унинг модуллари.

#### Режа.

1. Паскалр алгоритмик тили тшърисида.
2. Турбо-Паскални шрнатиш ва уни модуллари.
3. Интеграллашган мухит.

**Таянч иборалар:** *Паскалр алгоритмик тили тшърисида, Турбо-Паскалр ни шрнатиш, Турбо-Паскалр модуллари, интеграллашган мухит, блоклар билан ишлаш, редакторлаш асосий клавишлари, менюлар билан ишлаш, file менюси, edit менюси, search менюси, options менюси, compile менюси.*

Паскалр алгоритмик тили 1969 йил Н.Вирт томонидан яратилган. У юёйори даражадаги алгоритмик тил тисобланиб ёууйидагиларга эга:

1. Дастурлаштириш концепциясини ва структурасини системали ва аниё ифодалайди;
2. Дастур тузишни системали олиб бориш имконини беради.
3. Дастур тузиш учун бой термин, структура схемаларига эга.
4. Ёосил бшлган хатоликларни анализ ёилиш системасига эга бшлиб, у дастурлаштиришни маолум даражада шргатишни ташкил эта олади.

1981 йили Borland фирмаси томонидан яратилган Турбо-Паскалр алгоритмик тили яратилиб у ёам шз навбатида бир неча версиясига эга.

4.0 версиясидаги бошлаб дастур ёзишни киритишнинг бажарилишини ташкил этувчи интеграллашган мухит ёосил ёилиш имконияти киритилган.

5.5 версиясида Турбо-Паскалр ёрдамида байтли дастурлаштириш мумкинлиги.

6.0 версияда ассемблер тилида ёзилган дастурларни Паскалр тилида ёзилган дастур таркибига ёшшилиш имконияти ёшшилди.

Мутахассисларни фикрча Паскалр ва Си алгоритмик тиллари шртасида мусобаёа бшлиб туради. Улар бир биридан баози ёолатларда устунликларга эга. Масалан:

- Паскалр тилини СИ алгоритмик тилига ёараганда шрганиш осонроё;
- Паскалр тилида дастурни ички логикасини осон тушунилади;
- Си дастурлаштириш тили компрютер барча имкониятларидан фойдаланиш имкониятига эга;
- Си дастурлаштириш тили тез ва ёисёа дастур тузиш имкониятини беради;
- Си тилида ёзилган дастурни отладка ёилиш анча ёийин.

Турбо-Паскалр системаси одатда бир неча дискетларда берилади. У дискетлардан бири INSTALL/COMPILER бшлиб унда instell.exe файли ёам мавжуд. Instell.exe файли мавжуд бшлган файлни дисководга ёшйиб файлни ишга туширилади. Дастур фойдаланувчидан баози шсровларни олиб боради.

Масалан: системани винчестерга ыщясизми? Ва ыандай каталогларга жойлаштирасиз ва х.о.

Саволларга жавоб берилгандан кейин дастурни таклифига биноан кшрсатилган номли дискларни дисководга ыщйиб турилади. Шу билан Турбо-Паскалр системаси ишга тайёр бшлади системани асосий файллар \TP каталогида жойлашади. Улар ыуйидагилар:

- Turbo.exe - системани ишга тушириш файли;
- Turbo.hlp - ёрдам файли;
- Turbo.tp - Turbo.exe файлида ишлатиладиган конфигурациясини белгиловчи файли;
- Turbo.tpl - Turbo-Pascal ни резидент модуллари;
- Tptour.exe - интеграллашган мухитда ишлашни таништирувчи дастур;

c:\TP\BGI каталогида системасини график режимда зарур бшлган файллар жойлашган.

- Graph.tpu - график режимда ишлашни таоминловчи дастур
- Bgi - видео система билан ишлашни таоминловчи файллар ортирмаси
- Chr - шрифтларни саылаб турувчи файллар.

Интеграллашган мухит дейилганда дастур ёзишни ташкил этиш, дастурни зарур ьолатларда ишга тушириб кшриш, компановка даврида хатоликларни топиш, дастурни ихтиёрий ыисмини редакторлаш имконияти берувчи мухит тушинилади.

Turbo.exe файли бажаришга берилиши билан экранга интеграллашган мухит иш режимларини кшрсатувчи меню (юьори ыисмида), экранни ыуйи ыисмида энг зарурий клавишларни вазифалари чиьади.

F10 функционал клавишини босиш билан системани асосий менюсига кирилади. Бир бшлимдан бошыа бшлимга щтиш учун курсорни бошыариш клавишларидан фойдаланилади. Асосий менюни бирор бшлим ички менюсига кириш учун шу бшлимга курсорни олиб келиниб Enter клавиши босилади.

Масалан File номли асосий меню бшлимида ыуйдаги ички буйруылар мавжуд.

File-Open (F3 клавиши) - файлни редакторлаш учун хотирага юклаш. Бу буйруы ишга туширилиши билан экранга мавжуд файллар руйхати чиьиб, ихтиёрий файлни танлаш мумкин;

File-Save (F2 клавиши) - редакторланган файлини щз номи билан ыайта дискка ёзиш. Агар файл отга эга бшлиш система Nomane xx.pas номини бериб янги ном ёзиш имкониятини беради;

File-Save as - редактирлаётган файлни бошыа ном билан ёзиш;

File-New - интеграллашган мухит янги файл ташкил ыилишни бошлайди;

File-Exit (Alt+X) - интеграллашган системада ишлашни тугатиш;

Run-Run - дастурни ишга тушириш;

Compile- Compile - дастурни компановка ыилиш;

Compile.DestintIOOn - компиляция натижасини дискга ёзиш керакли ёки фаьат хотирада ыолишини белгилаш.

Агар компиляция фаыат хотирада бщлса "MEMORY" сщзи, агар дискда бщлса "Disk" сщзи чиыади. Бир режимдан бошыа режимга щтиш учун "ENTER" клавишини босилади.

OptIOOn-Directories - интеграллашган мухитда ишлатиладиган каталогни аниылаш.

Интеграллашаган мухитда дастурни редакторлаш учун кенг тарыалган умумий клавишлардан фойдаланиш методикасига кенг эотибор берилган.

Редакторлаш асосий клавишлари:

Enter - янги ыатор ыщшиш;

Del - курсор турган символни щчириш;

Backspace - курсор олдида турган символни щчириш;

Ins - ыщйиш режимини щрнатиш ёки олиш;

Вверх, вниз стрелки, PgUp, PgDn - курсорни бошыариш;

Ctrl+PgUp - текст бошига щтиш;

Ctrl+PgDn - текст охирига щтиш;

Блоклар билан ишлаш:

Ctrl - "K""B" - блок бошини белгилаш;

Ctrl - "K""K" - блок охирини белгилаш;

Ctrl - "K""Y" - белгиланган (ажратилган) блокни щчириш;

Ctrl - "K""C" - блокдан нусха олиш;

Ctrl - "K""V" - блокни бошыа жойга кщчириш;

Ctrl - "K""P" - блокни ыобозга чиыариш;

**Назорат саволлар.**

1. Паскалр алгоритмик тили тўғрисида нималарни биласиз?
2. Турбо-Паскалр ни шўнатиш ыандай бўлади?
3. Турбо-Паскалр модуллари хаида нималарни биласиз?
4. Интеграллашган мухит хаида нималарни биласиз?
5. Блоклар билан ишлаш ыандай бўлади?
6. Редакторлаш асосий клавишлари тўғрисида нималарни биласиз?
7. Менюлар билан ишлаш ыандай бўлади?
8. File менюси хаида нималарни биласиз?
9. Edit менюси хаида нималарни биласиз?
10. Search ,options ва compile менюлари тўғрисида нималарни биласиз?

## 7 - М А О Р У З А

### Катталиклар ва уларнинг турлари.

#### Режа.

1. Паскалр тилида шзгарувчилар ва уларни белгиланиш.
2. Катталиклар типлари.
3. Символлар ва ыаторлар.
4. Касир ва бутун сонлар.

**Таянч иборалар:** *Паскалр тилида шзгарувчилар, шзгарувчиларни белгиланиши, катталиклар типлари, символлар , ыаторлар, касир сонлар, бутун сонлар, логик типли шзгарувчи, кшрсаткичлар,манзиллар.*

Дастур бажарилиши давомида берилганлар ва ыисоблаш натижалари, оралиы натижалар шзгарувчилар ёрдамида саыланади.

Шзгарувчи бу хотирани маолум ыисми бщлиб баози информацияни саылаб туриш учун хизмат ыилади ва аниы номга эга бщлади.

Паскалр алгоритмик тилида шзгарувчидан фойдаланишдан олдин уни типи (тури) ва номи кшрсатилиши керак.

Шзгарувчи тури (типи) шзгарувчи ыабул ыилиши мумкин бщлган ыйиматларни олдиндан белгилаш билан бирга хотирага ыанча жой зарур бщлишини белгилайди. Шзгарувчи номи, хотирани кшрсатилган (ажратилган) жойдаги информацияга дастурда мурожат ыилишни таоминлаш учун ишлатилади.

Турбо-Паскалр шзгарувчиларни тилини мавжуд белгилаган сщзлар ёрдамида аниыланади. Масалан:

Integer - бутун сонли шзгарувчили белгилайди.

String - символли (текстли) шзгарувчи.

Boolean - логик типли шзгарувчи. Дастурда иштирок этадиган шзгарувчиларни дастурни бош ыисмида Var сщзи аниылаштирилади. Масалан:

Var

I, J: Integer {I, J - иккита бутун сонли шзгарувчи}

S:String {S - символли шзгарувчи}

Бу ёзув билан биз дастурда иштирок этадиган I, J шзгарувчилар фаыат бутун сонларни шзлаштиришини ва хотира уларни хар бири учун 2 байтдан жой ажратиши кераклигини кшрсатамиз. S - шзгарувчи эса текстли (символли) шзгарувчи эканлигини билдириши билан у учун 256 байт жой ажратилиши билдирилади.

Шзгарувчиларни номи битта дастур таркибида бир хил бщлмаслиги ва улар имконият даражасида шзлаштирадиган катталиклар номларига мос бщлиши маысадга мувофиы. агар дастурда иккита катталик бир хил номи



билан номланган бўлса дастурни компиляция ыилиш даврида хатолик борлиги кўрсатилади (Error4: Duplicate identifier).

Ўзгарувчи номи дастурда латин алфавитини бош ёки кичик ҳарфлари билан ёзилиш мумкин. Турбо-Паскалр улар орасидаги фарқни эътиборга олмайди.

Ўзгарувчи номлари латин алфавити ҳарфлари ёки "\_\_\_" белги билан бошланиб ҳарфлар, рақамлар ёки "\_\_\_" белгисидан иборат бўлиб чегаравий сони 63 та позиция. Масалан:

```
Var
  My Name, Imy,: String
  Kol_tivar: Integer
```

Одатда дастур таркибида иштирок этувчи ўзгарувчилар бутун сонли, каср сонли, символли, логик ва кўрсаткич типларига бўлинади.

Бутун сонли типига мисол ыилиб 14, 16, -111, 777 ва бошчалар келтириш мумкин. Уларни ифодалаш учун махсус сўзлар Integer, ShortInt, Byte, Longint ва Word дан фойдаланиш.

Каср сонларга 2.12, 3.14, - 2.97 ва х.о. каср сонларни ифодалаш учун Real, Single, Double, Extended ва Comp. сўзларидан фойдаланилади.

Символларни, символли ыаторларни ифодалаш учун String сўзидан фойдаланилади.

Маълумки, логик катталиклар икки хил ыйматни True (ҳақиқат) ва False (ёльон) ни сўзлаштириш мумкин. Бундан ўзгарувчиларни ифодалаш учун Boolean сўзидан фойдаланилади.

Кўрсаткич компютер хотирасидаги берилганлар жойлашган хотира адресини кўрсатади. Кўрсаткичлар Pointer сўзи билан ажратилади.

Бутун сонли тип ўзгарувчилар юъорида кўрсатилгандек бир неча сўзлар билан белгиланади. Улар бир-биридан кўрсатилган ўзгарувчи учун хотирада ыанча миъдордаги жой ажратиш зарурлигини белгилайди.

Shorting - 128 ... 127	- 8 бит
Integer - 32768 ... 32767	- 16 бит
Longint - 2147483648 ... 2147483647	- 32 бит
Byte - 0 ... 255	- 8 бит
Word - 0 ... 65535	- 16 бит

Кўриниб турибдики дастурда иштирок этаётган ўзгарувчинини типини тўғри кўрсатиш хотирадан эффектив фойдаланишда муҳим амалиётга эга бўлади.

Бирор алгоритмик тилда дастур тузишни шрганишда ҳар бир янги нарсга мисол билан ёндошиш маъсадга мувофиъ. Шунинг учун элементар дастурларни тузиш асосларини бериб борамиз.

Мисол учун бутун сонлар билан ишловчи дастурларга ыуйдагиларни келтириш мумкин:

```
Program butun_s (бу ыаторни ёзиш шарт эмас)
```

```

Var (щзгарувчилар типини кщрсатиш сщзи)
  I, J: Integer; (I ва J бутун сонли типли щзгарувчи эканлигини
кщрсатиш)
  Sum, Kup: Longint (Sum, Kup узун бутун сонли типдаги щзгарувчилар)
begin (дастурни бажарилувчи ыисми begin сщзи билан бошланади)
  Writeln ('биринчи сонни киритинг');
  Readln (I);
  Writeln ('Иккинчи сонни киритинг');
  Readln (J);
  Sum: I+J;
  Kup: I*J;
  Writeln ('Улар йиьиндиси =', Sum);
  Writeln ('Улар кщпайтмаси =', Kup);
end (Паскалда дастур шу сщз (оператор) билан тугалланади).

```

Келтирилган дастур бажаришга берилса экранда "Биринчи сонни киритиш" деган информация чиyaди. Компьютер сон киритилишини кутиб туради.

Сон киритилиши билан иккинчи сонга ыймат сщралади ва ьоказо. Бу ерда ыщштирноы ичидаги информацияни ыщйиш кщймаслик натижага таосир этмайди. Лекин уларни ыщйиш дастурдан фойдаланашни анча тушинарли ьолатга келтиради.

Дастурни ыуйидагича модификациялаш мумкин:

```

Program butun_s2
Var
  I, J: Integer
begin
  Writeln ('биринчи ва иккинчи сонларни киритинг')
  Readln (I, J)
  Writeln ('улар йиьиндиси', 'I+J', кщпайтмаси I*J)
End.

```

Турбо-Паскалр алгоритмик тилида 16 лик саноы системасидаги сонлар билан ьам ишлаш имконияти мавжуд. Фаьат сонни олдига \$ белгисини ыщйиш билан сонли 16 лик системасида эканлигини билдирилади.

Каср сонли катталиклар дейилганда 10 лик каср сонлар тушинилади. Агар компьютерда сопроцессор мавжуд бщлса Турбо-Паскалр ыуйида келтирилган типларни барчасида ишлай олади. Агар йщы бщлса фаьат "real" типидагина ишлатилади.

типи	сон оралиьы	разряди	хотирада жой эгалаши
real	2.9e-39-1.7e38	11-12	6 байт
single	1.5e-45-3.4e38	7-8	4 байт
double	5.0e-324-1.7e308	-16	8 байт

extended	3.4e-4932-1.1e4932	1 9-20	10 байт
comp	-92e18-9.2e18	12-20	8 байт

Масалан:

```

Program Real_1;
Var
    I, J: Real;
Begin
    WriteLn ('Биринчи сонни киритинг');
    ReadLn (I);
    WriteLn ('Иккинчи сонни киритинг');
    WriteLn ('Улар йиьиндиси=', I+J);
    WriteLn ('Улар кцпайтмасининг=' I*J);
    If j <> 0 then WriteLn ('Улар бщлинмаси=' I/J);
end.

```

Ўамма раъам, ьарф, белгилар ASCII жадвалда аниы номерга эга. Шу номерни символни коди деб аталади. Символни дастурда кщрсатиш учун у ыщштирноыыа олинади ёки - билан биргаликда уни жадвалдаги коди келтирилади. Масалан "А" символини #65 симболи деб ьам кщрсатиб кетиш мумкин.

Символли щзгарувчилик Паскалр тилида char сщзи билан кщрсатиб ыщйилади. Хар бир символини кщрсатиш учун хотирада бир байт жой ажратилади.

Ыаторли щзгарувчи string сщзи билан кщрсатиб щтилади. Битта ыаторли щзгарувчи 255 позицияча бщлиши мумкин. Демак string билан кщрсатилган ыатор хотирада 255 та байт жой эгаллайди. Ыаторли катталиклар учун +1 та хотира ячейкаси ишлатилади. Битта ыщшимча байтда ыатор узунлиги саыланади.

Демак битта ыаторлик катталикини саылаш учун 256 та байт зарур бщлади.

Хотирадан эффектив фойдаланиш учун агар ыаторни катталик узунлиги аниы бщлса уни кщрсатиш маъсадга мувофиы. Масалан string [50]. Мисол:

```

Var
    ch: char; (ch - симболи)
    Name: string [10]; (Name 10 та символдан иборат ыатор)
    Inform: string; (Inform - 255 та символдан иборат ыатор)

```

Дастурни кщрсатилган ыисмида символ ва ыаторлик катталикларга хотирада жой ажратиб ыщйилади.

Паскалр тилида ыаторларни бир бирига ыщшиб янги ыатор ьосил ыилиш, ыаторларни бироы ыисмни ажратиб олиш, янгиси билан алмаштириш каби операцияларни бажариш мумкин.

Мисол:

```
Program Any String;  
Var  
    ch: Char;  
    Any, Body: String;  
Begin  
    ch:='A';  
    Body:='ABCD';  
    Any:='ch+ch';  
    Any:='Any+Body';  
    WriteLn (Any);  
    Any:=ch+Any+'***'+Body;  
    WriteLn (Any);  
end.
```

### **Назорат саволлар.**

1. Паскалр тилида щзгарувчиларга таориф беринг.
2. Щзгарувчиларни белгиланиши ыандай бщлади?
3. Катталиклар типлари деганда нимани тушинасиз?
4. Символлар билан ишлаш ыандай бщлади?
5. Ыаторлар билан ишлаш ыандай бщлади?
6. Каср сонлар ишлаш ыандай бщлади?
7. Бутун сонлар ишлаш ыандай бщлади?
8. Логик типли щзгарувчига таориф беринг.
9. Кщрсаткичлар деганда нимани тушинасиз?
10. Манзиллар деганда нимани тушинасиз?

## 8 - М А О Р У З А

### Массивлар ва тўпламлар

#### Режа.

1. Массивлар.
2. Ёзувлар.
3. Тўпламлар.
4. Қўрсатувчилар.
5. Файли типлар ва янги ёторни ташкил этиш.

**Таянч иборалар:** *Массивлар, ёзувлар, тўпламлар, қўрсатувчилар, файли типлар, янги ёторни ташкил этиш, типли файллар билан ишлаш, типсиз файллар билан ишлаш, матнли файллар билан ишлаш, файллар билан ишлашнинг асосий процедуралари.*

Массивлар аниё миёдордаги элементларга эга бўлиб ҳар бир элемент массивда аниё номерга эга ва массив маолум отга эга. Паскалр тилида массивни катталиклар “array” сўзи билан белгиланади ва элементларни номерга диапозини қўрсатилади.

Мисол:

Program Arrays

Var

M1: array [5...100] of real;  
I: array [-1...100] of integer;  
VM: array [1..10, 1...20] of string;

begin

M1 [7]: = 4.3417;  
I [0]: = 240;  
VM [5,3]: = 'This in a string';

end.

Паскалр алгоритмик тилида бир неча типдаги шўгарувчилар, ёки бир типдаги бир неча шўгарувчилар битта блокга ёзувлар ёрдамида бирлаштирилади. Ёзув record сўзи билан белгиланади.

Мисол:

Var

Dragon = Record  
First\_head: String [40];  
Recond\_head: String [40];  
Whird\_head: String [40];  
end;

begin

Dragon.First\_head:='Antoni';

```

    Dragon.Second_head:='Mory';
    Dragon.Third_head:='Dr.B.X. Soron';
end.

```

2-чи мисол:

```

Program Record
Var
    Circle=Record;
    x, y: Integer;
    Reditus: Integer;
    Color: Byte;
    end;

    Book=record
    Writer: String: [50];
    year: Integer;
    end;
Begin
    Circle.x:=20;
    Circle.y:=20;
    Circle.color:=1;
    Book.writer:='Tom';
    Book.year:=1613;
    K:=Book.year;
end.

```

Турбо-Паскалр алгоритмик тилида битта ёзув таркибига кирувчи индентификаторлардан фойдаланишда With... do ёрдамида ёзув номини ыайта ыайта ёзмаслик имкониятини беради. Мисол учун ьозирги мисолни ыуйидагича ёзиш мумкин:

```

Program record_3
Var
    circle=Record
    x, y: integer;
    Radius: integer;
    Color: Byte;
    end;
    Book=record
    Writer: string [50];
    year: integer;
end;
begin
    With Circle do;
    begin

```

```

        x:=20;
        y:=20;
        Color:=1;
    end;
    With Book do
    begin
        Writer:='Tom';
        Year:=1613;
        K:=year;
    end;
end.

```

Паскалр тилида тщпламлар билан ишлаш кщзда тутилган. Тщплам дейилганда ихтиёрий тартибдаги обоектлар йиьиндиси (бирлиги) тушунилади. Тщпламни элемент сони 256 та ками 0 та.

Тщпламлар ыуйидагича кщрсатилади:

```

Program Set_Example
Var
    Set1, set2, set3: set of 0...9;
    My_Name chars: set of char;
    Test: Boolean;
begin
    Set1: [2,5];
    Set2: [2, 0, 7, 2];
    Set3: Set1 * Set2;
    Set3: Set1 + Set2;
end.

```

Кщрсаткичлар щзгарувчи ьисобланиб улар берилганларни щзларида саыламасдан балки берилганлар адресларини, яони хотирада жойлашган жойни саылайдилар. Кщрсаткич щзи фаьат 4 байт жой эгаллаб, берилмалар эса щнлаб Кбайт жой эгаллаши мумкин.

Кщрсаткичдан фойдаланишдан олдин хотирани бирор участкасини ажратиб уни белгилаб олинади. (Танланган номли кщрсаткич билан). Бу ишни GetMem ёки New процедуралари ёрдамида амалга оширилади.

Мисол:

```

Var
    P: pointer (P ни кщрсаткич деб олиш);
begin
    Getmem (P, 1000);
end.

```

Кщрсаткични Addr функцияси ёрдамида аниы бир щзгарувчини кщрсатилиши ёки Ptr функцияси ёрдамида аниы хотирани ячейкасига



мурожаат ыилишни ташкил ыилиш мумкин. Бундан ташыари бирор кщрсаткични фаыат ыаторли щзгарувчини кщрсатувчи яони

```
My Pointer: ^ String;
ёки байтни кщрсатувчи
Pointer 1: ^ byte деб белгилаш мумкин.
```

Мисол:

```
Var
  S: String [200];
  P: ^byte;
  SPtr: Pointer;
begin
  S: ='This is a string';
  P: =Ptr ($ 50, $ 55);
  SPtr: = Addr (S);
end
```

Информацияни магнит дискига ёзиш ёки щыиш учун Паскал тилида файл типли щзгарувчан фойдаланилади. Файл типли щзгарувчини дастурни юыори ыисмида file ёки text сщзлари билан аниыланади.

Мисол:

```
My File 1, My File 2: file
Big File: file
Росм: text
```

Text шартли сщзи билан белгиланган файл матнли бщлиб маолум узунлигидаги ыаторларидан иборат бщлади.

Кщпчилик холларда дискка ёзилаётган ёки щыилаётган файлдаги информациялар бир хил типда бщлиши мумкин. Бу холларда щзгарувчини белгилашда уни типини кщрсатиб кетиш лозим. Мисол:

```
f byte: file of byte
my date: file of integer
stars: file of string
names: file of string [20]
letter: text
```

Агар of ва ундан кейинги ёзувлар иштирок этмаса файлни типлаштирилмаган дейилади. Мисол:

```
Var
  my data: file of integer;
  i, j, sum: integer;
begin
  assign (my data, 'd:\to\ my file dat');
  rewrite (my data);
```

```

Writeln ('Юстас-Алексу ...');
Writeln ('Биринчи сонни киритиш');
Readln (I);
Writeln ('Биринчи сонни ёзаман');
Writeln (my data, I);
Writeln (Иккинчи сонни киритинг);
Readln (I);
Writeln ('Иккинчи сонни ёзаман');
Writeln (my data, J);
Sum: =I+J;
Writeln ('Уларнинг йиьиндиси', sum);
Writeln (my data, sum);
Close (my data);
end.

```

Файллар билан ишланаётганда ыуйидаги процедуралардан фойдаланилади:

Assign (f: name: string)

Бу процедура ташыи щзгарувчини номига файл щзгарувчини щзлаштирилади: f - файли щзгарувчи name - файл номи.

Close (f) - очилган файлни беркитиш. f - олдиндан очилган файлни щзгарувчиси.

Read (R, V): - файлдан файл элементини щыиш. V - файл элементи.

Reset (f: file) - мавжуд файлни очиш f - файл щзгарувчи олдиндан assign процедураси ёрдамида файл билан боъланиши керак.

Rewrite (f:file) - янги файл ташкил этиш ва очиш. Бу процедурада ьам файли щзгарувчи файл номи билан assing процедураси билан боъланиши керак. Янги ташкил этилаётган файл номи assing да кщрсатилади.

Агар шу номи билан файл мавжуд бщлса у щчириб янги шу номли файл ьосил бщлади.

Write (f, v) - файли щзгарувчи элементлари ёзилади.

Юьюридаги мисол d: дискни tp каталогидан my file.dat файлини ьосил ыилиш унга сонлар ёзилди. Энди ундан информацияни щыиш дастурини кщрайлик:

```

Var
    mydata: file of integer;
    i, j, sum: integer;
begin

```

```

assign (my data, 'd:\tp\ my file dat');
reset (my data);
Writeln ('Юстас-Алексу ...');
Readln (mydata, I);
Writeln ('myfile.dat файлидан биринчи сон щыилди');
Writeln ('Y=', I);
Readln (my data, J);
Writeln ('myfile.dat файлидан иккинчи сонни щыилди');
Writeln ('Y=', J);
Readln (my data, sum);
Writeln ('myfile.dat файлидан sum щыилди');
Writeln (my data, sum);
Close (my data);
end.

```

Text файл типи матнлар билан ишлашга мўлжалланган. Бу тип ыаторлардан иборат бўлиб ҳар бир ыатор ыатор штиши ёки “Enter” билан ажратилган ыаторлар file of string типида кўрсатилаган каби узунликга эга бўлиши мумкин. Text файл типидан фойдаланишдан Readln ва Writeln процедураларида олдиндан маолум бўлмаган узунликдаги ыаторлар билан ишлаш мумкин.

Мисол:

```

Var
    mytext: text;
    S: string;
begin
    assing (mytext, 'd:\tp\mytext.txt');
    rewrite (mytext);
    Writeln ('Сизнинг исмингиз');
    Readln (S);
    Writeln ('Сизни исмингиз диска ёзиляпти');
    Writeln (mytext, S);
    Close (mytext);
    Writeln ('Файл ёпилди');
end.

```

### **Назорат саволлар.**

1. Массивларнинг вазифасини ифодалаб беринг.
2. Ёзувлар билан ишлашни тушунтиринг.
3. Тщпламлар хаёида нималарни биласиз?
4. Кщрсатувчилар хаёида нималарни биласиз?
5. Файллар билан ишлашни тушунтиринг.
6. Янги ыаторни ыандай ташкил этади.
7. Типли файллар билан ишлашни тушунтиринг.
8. Типсиз файллар билан ишлашни тушунтиринг.
9. Матнли файллар билан ишлашни тушунтиринг.
10. Файллар билан ишлашнинг асосий процедураларини ифодалаб беринг.

## 9 - М А О Р У З А

### Паскалр алгоритмик тилида операциялар.

#### Режа.

1. Операциялар ва приоритетлик.
2. Щзлаштириш операцияси.
3. Арифметик операциялар.
4. Муносабат операциялари.

**Таянч иборалар:** *Операциялар, приоритетлик, щзлаштириш операцияси, арифметик операциялар, битлар устида операциялар, логик операциялар, адрес операциялари, тцплаш операциялари, ыаторли операциялар, типлар устида операциялар.*

Паскалр тилида берилганлари ыайта ишлаш учун ыуйидаги операциялар тури мавжуд:

- щзлаштириш операцияси (:=)
- арифметик операциялар (+, -, \*, /, div, mod)
- битлар устида операциялар (shl - чапга суриш, shr - шнга суриш ва бошыалар)
- муносабат операциялари (>, <, >=, <=)
- логик операциялар
- адрес операциялари
- тцплаш операциялари
- ыаторли операциялар

Маолумки битта ифодада бир неча операциялар мавжуд бщлиши мумкин. Масалан:

$X + 2 \text{ div } 8 - 6 * 13$

Ифодани бажарилишида операцияни приоритетлигига эотибор берилади.

Келтирилган мисолда div (бутун бщлиш) биринчи, иккинчи кщпайтириш, учинчи ыщшиш ва ниъоят айириш операцияси бажарилади.

Паскалр тилида операциялар приоритетликни ыуйидагича ифодалаш мумкин.

операциялар	приоритет	категория
@, not	биринчи	адресли
*, /, div, mod, and, shl, shr	иккинчи	кщпайтириш
+, -, or, xor	учинчи	ыщшиш
=, <, >, <=, >=, in	тщртинчи	муносабат

Щзлаштириш оператори (операцияси) := шаклида бщлиб у, щзгарувчига тенгликдан кейинги ифода ыйиматини щзлаштиради. Ифода щрнида щзгарувчи сон ьам бщлиши мумкин.

Арифметик операцияларни бажарилиш тартиби, бошья алгоритмик тиллари каби . Фаъят бу ерда бутун бўшлиш ва модулга олиш операцияларни ишлаш принципига эотибор бериш керак.

Масалан:  $5 \div 2 = 2$   
 $17 \bmod 3 = 2$

Муносабат операциялари бажарилиши билан иккита ўлатни биринчи True (хабиыат) ёки False (ёльон) беради. Яони икки ыйматни солиштириш юыоридагини бирини танлайди.

Мисол:

```
Program Proba
Var
  A, Secret: integer;
  Test: Boolean;
begin
  Secret: = 3;
  Writeln (' 0 дан 9 гача бўлган ихтиёрий сон киритинг');
  Readln (A);
  Test: =A= Secret;
  Writeln ('Сизни жавобингиз' Test);
end.
```

Бу келтирилган дастур киритилаётган сонни '3' га солиштиради. Агар 3 сонни киритилса True, акс холларда False чиыади.

### Назорат саволлар.

1. Операцияларга таориф беринг?
2. Приоритетликка таориф беринг?
3. Щзлаштириш операцияси деганда нимани тушинасиз?
4. Арифметик операцияларни ишлаш жараёни ыандай?
5. Битлар устида операцияларни тушунтиринг?
6. Логик операцияларни тушунтиринг?
7. Адрес операцияларини тушунтиринг?
8. Тщплаш операцияларини тушунтиринг?
9. Ёаторли операциялар деганда нимани тушунасиз?
- 10.** Типлар устида операциялар деганда нимани тушинасиз?

## 10 - М А О Р У З А

### Мураккаб операциялар.

#### Режа.

1. Логик операциялар.
2. Адреслар устида амаллар.
3. Ёаторлар устида операциялар.

**Таянч иборалар:** *Логик операциялар, адреслар устида амаллар, ёаторлар устида операциялар, шзгарувчини адресини ёисоблаш операцияси, кшрсатиш операцияси, ёайта номлаш операцияси, ёаторли катталиклар устида операциялар, арифметик операциялар устида операциялар, битлар устида операциялар, тицплаш операциялари устида операциялар.*

Маолумки ёуйидаги логик операциялар мавжуд:

And (ва) - агар икки шарт бажарилсагина True (хаёиёат) ёийматини беради.

or (ёки) - хеч бщлмаса икки шартни бири бажарилса True (хаёиёат) ёийматини беради.

Not (ёшчы) - хаёиёат ёийматни ёльонга, ёки тескарисига алмаштириш операцияси.

Мисол:

Var

Years; Money: Integer;

Test: Boolean;

begin

Writeln ('Введите 2 числа');

Readln (Years, Miney);

Test: = (Years = 100) and (Money = 100\*100) OR;

(Years = 100\*100) and (Money = 100);

Writeln ('Test=' Test);

end.

Дастур бажаришга берилса киритилган сонни бири 100, иккинчиси 10000, иккинчи сон эса 100 га тенг бщлгандагина тенг (хаёиёат) сщзи чиёади. Бошыа хар ёандай сонлар киритилса False (ёльон) натижа беради.

Адреслар устида ишлашни икки операцияси мавжуд.

1. (@) - шзгарувчини адресини ёисоблаш операцияси.

2. (^) - кшрсатиш ёки ёайта номлаш операцияси.

Мисол:

Program Arrow\_And\_Goal

type

string Ptr=^string (ёаторга кшрсатувчи типи аниёланди)



```

Var
    Arrow: string Ptr (Arrow - ыаторга кщрсатувчи деб аниыланди)
    Goal: string; (Goal - ыаторли катталик)
begin
    Goal: 'I am a Goal' (Goal - щзгарувчи, 'I am a Goal' - ыаторини
щзгартириш)
    Arrow: @ Goal (Arrow Goal - щзгарувчини адреси бщйича
аниылангани кщрсатади)
    Arrow^: 'The Arrow Was there' (Arrow - кщрсатаётган адреси
щзгарувчи энди 'The Arrow Was there' сщзига айланди. Янги Goal энди 'The
Arrow Was there' тенг)
end.

```

Мисол:

```

Program Adres
type
    Qatorcos=^String
Var
    Adres: Qatorcos
    Gap: string
begin
    Gap:= 'Мен кетдим'
    Adres:= @ Gap
    Adres ^:= 'Сен кутдинг'
    Writeln (Gap)
end.

```

Дастур бажаришга берилса натижа “Сен кутдинг” сщзини печат ыилади. Дастур бош ыисмида Gap щзгарувчисига “Мен кетдим” щзлаштирилган. Кейин уни адреси топилиб шу адресга бошыа ыаторли катталик ёзилади.

Ыаторли катталиклар устида операциялар дейилганда фаыат ыаторларни ыщшиш (ёпиштириш) тушинилади. Мисол:

```

Var
    String_, My Str: String;
begin
    My Str: = '-1234-';
    String_:= My Str+My Str;
end.

```

Мисол 2:

```

Program String1;
Var
    N, Nam1, Nam2: String [10];
begin

```

```
Nam1: = 'Карим';  
Nam2: = 'Жон';  
N: =Nam1+Nam2;  
Writeln (N);
```

end.

Дастур бажарилишга берилса экранга “Каримжон” сўзи ўосил бўлади.

### Назорат саволлар.

1. Логик операцияларга таориф беринг.
2. Адреслар устида амалларни ишлаш жараёни ыандай бщлади?
3. Ыаторлар устида операцияларни ишлаш жараёни ыандай бщлади? 4. Щзгарувчини адресини ыисоблаш операцияси деганда нимани тушинасиз?
5. Кщрсатиш операциясини ишлаш жараёни ыандай бщлади?
6. Ыайта номлаш операциясини тушинтиринг.
7. Ыаторли катталиклар устида операциялар деганда нимани тушинасиз?
8. Арифметик операциялар устида операциялар деганда нимани тушинасиз?
9. Битлар устида операцияларни тушинтиринг.
10. Тщплаш операциялари устида операциялар деганда нимани тушинасиз?

## 11 - М А О Р У З А

### Цикл операторлари.

#### Режа.

1. Repeat оператори.
2. While оператори.
3. For оператори.

**Таянч иборалар:** *Цикл операторлари, Repeat оператори, While оператори, For оператори, until ифода, қўрсатиш операцияси, ыайта номлаш операцияси, ыаторли катталиклар устида операциялар, арифметик операциялар устида операциялар, битлар устида операциялар, тицплаш операциялари устида операциялар.*

Дастур таркибидаги бирор блокни бир неча маротаба ёки бирор шарт бажарилганга ыадар бажариш (ыисоблаш) талаб ыилинса цикл операторларидан фойдаланилади.

Repeat оператори Until сщзи билан биргаликда ишлатилади. Операторни ишлатиш формаси ыуйидагича

```
Repeat
оператор
оператор
until ифода
```

Бу ерда “ифода” шрнида логик ифода бщлиб, цикл логик ифода ‘хаыиыат’ ыийматини олингунча ыадар давом этади.

Мисол:

```
Program Sic1 1;
Var
    passw, answ: string;
begin
    passw: ='Boss';
    Repeat ;
    Writeln ('Фойдаланувчи паролни киритинг') ;
    Readln (answ);
    if passw <> then
        Writeln ('Сиз фойдаланувчи эмас')
    else Writeln ('Салом фойдаланувчи');
    Until answ=Passw;
end.
```

Келтирилган дастур ‘Boss’ сщзи парол сифатида киритилмагунча бажаришдан тцхтамайди.

While цикл операторида шартни бажаришишини текширилиш, яони циклни сонини цикл бош ыисмида аниыланади. Цикл формаси ыуйидагича;

While ифода do операторлар;

Бу ерда ҳам ифода логик ифодадир. While операторидан фойдаланиб юзоридаги дастурни буйидагича ёзиш мумкин;

```

Var
passw, answ: string;
begin
    passw:= 'Boss'
    answ:=''
    While answ <> Passw do
begin
    Writeln ('Паролни киритинг')
    Readln (answ)
    if pass <> answ then Writeln ('Сиз фойдаланувчи эмассиз')
    else writeln ('Салом фойдаланувчи')
end;
end.

```

For оператори циклни ташкил этган операторларни неча маротаба ыайта ишлаш сони аниы бщлган холларда ыщлланиши маысадга мувофиы.

Циклни ёзиш формаси:

for индекс: = 1 ифода to 2 ифода do операторлар.

Бу ерда индекс - бутун типи щзгарувчи, 1-ифода ва 2-ифодалар бутун типли ифодалардир. Мисол:

```

Program sick3;
Var
    i: integer;
begin
    for i:= 1 to 100 do;
        Write (*');
end.

```

Циклда щзгарувчи ыйматини фаыат орттирилмасдан камайтирилиши ҳам мумкин:

```

Program sircl5
Var
    i: integer
begin
    for i:= 100 do to 1 do
        down to
        Write (*')
end

```

**Назорат саволлар.**

1. Цикл операторларига таориф беринг.
2. Repeat операторини тушунтиринг.
3. While операторини тушунтиринг.
4. For операторини тушунтиринг.
5. Until ифодани тушунтиринг.
6. Ёайта номлаш операциясини тушинтиринг.
7. Ёаторли катталиклар устида операциялар деганда нимани тушинасиз?
8. Арифметик операциялар устида операциялар деганда нимани тушинасиз?
9. Битлар устида операцияларни тушинтиринг.
10. Тщплаш операциялари устида операциялар деганда нимани тушинасиз?

## 12 - М А О Р У З А

### Паскалр тилида дастур структураси.

#### Режа:

1. Дастурда белги (метка)
2. Доимийлар.
3. Дастур структураси.

**Таянч иборалар:** *Дастурда белги (метка), доимийлар, дастур структураси, Цикл операторлари, Repeat оператори, While оператори, For оператори, until ифода, кичрсатиш операцияси, ыайта номлаш операцияси, ыаторли катталиклар устида операциялар.*

Агар дастурда Goto оператор иштирок этган бўлса уни штиш жойи белги ёрдамида кичрсатилиши керак. Операторни ёзилиш формаси:

Label L1, L2

Мисол:

```
Var
  x: integer
begin
  x:=-1
  L1: x: =x+1
  L2: if x<100 then Goto L1
  Writeln (x)
end.
```

Паскалр тилида дастур ёзишда кшп маротаба ишлатиладиган баози доимийларни белгилаб олиш маъсадга мувофиы. Мисол:

```
Const
  Width = 50
  Height = 10
Var
  I, J: Integer
begin
  Writeln
  for J: 1 to Height do
  begin
    For I: 1 to Width do Write ('*')
    Writeln
  End;
End.
```

Паскал тилидаги дастурни умумий структураси ыуйидагича бщлади.

Program <дастур номи>

label <белгилар>

Const <доимийлар>

Type <берилгалар типини аниылаш>

Var <щзгарувчиларни кщрсатиш>

<процедуралар ва функциялар>

begin <асосий дастур танаси>

end.

Паскал тилида кщрсатишга бош бщлим: label, type, Const, var процедура ва функциялар бщлиши шарт эмас. Агар улар бщлса Паскалр тилида улар кщрсатилган тартибда жойлашиши керак ва хар бир бщлим бир марта бщлиши шарт.

Турбо-Паскалда юыоридаги аниы кетма-кетлик саыланмаслиги мумкин. Фаыат дастур номи биринчи дастур танаси эса дастурда охири бщлиши керак. Улар дастурда ихтиёрий кетма-кетликда сонда бщлиши мумкин. Лекин бир бщлимда иштирок этаётган щзгарувчилар унда олдинги бщлимда аниыланган бщлиши шарт.



### **Назорат саволлар.**

1. Дастурда белги (метка)ни тушунтиринг.
2. Доимийларни тушунтиринг.
3. Дастур структурасини тушунтиринг.
4. Repeat операторини тушунтиринг.
5. While операторини тушунтиринг.
6. Цикл операторларига таориф беринг.
7. For операторини тушунтиринг.
8. Until ифодани тушунтиринг.
9. Ёайта номлаш операциясини тушинтиринг.
10. Ёаторли катталиклар устида операциялар деганда нимани тушинасиз?

## 13 - М А О Р У З А

### Процедура ва функциялар.

#### Режа.

1. Процедуралар аъамияти.
2. Процедуралар.
3. Процедура параметрлари.

**Таярч иборалар:** *Процедуралар аъамияти, процедуралар, процедура параметрлари, “локал” иъзгарувчи, поддастур, ыйта номлаш операцияси, ыаторли катталиклар устида операциялар, процедурани структураси, глобал иъзгарувчи, статус иъзгарувчи.*

Маолумки кшпчилик алгоритмик тилларда ыйтарилиб ыисобланувчи операцияларни алоъида блоklarга олиниб бу блоklarга зарур маротаба мурожаат ыилиш ташкил этилади. Бу эса дастурларни эффективлигини ортириб, уларни кшринишини ыисыа ва компакт бшклишига ёрдам беради. Бейсик тилида бундай бшлакларни поддастур деб аталади.

Паскалр алгоритмик тилида ым бу операцияларни ташкил ыилиш эотиборга олинган бшлиб уларни процедура деб аталади.

Мисол сифатида ыуйидагини олайлик. Айтайлик биздан экранга 100 та юлдузчаларни чоп этиш талаб ыилинсин. Дастурни ыуйидагича ёзамиз:

```
Program Stars_100;
Prosedure Stars;
Var
  i: integer;
begin
  for i=1 to 100 do;
    Writeln('*')
  end
begin
  Stars;
  Writeln('100 та юлдузча')
  Stars;
  Writeln('200 та юлдузча')
end
```

Мисол. Тшъри тшрт бурчак шаклга эга бшлмаган фигурани юзасини ыисоблаш талаб ыилинсин. Ихтиёрий фигурани иккита ёки бир нечта учбурчак шаклга бшлиб олиш мумкин ва уларни хар бирига ыуйидаги Герон формуласини ышллаб юзаларини топиб, юзалар йииндиси олинади:

$$S = \sqrt{P \cdot (p - a)(p - b)(p - c)}$$

Бу ерда  $a$ ,  $b$ ,  $c$  - учбурчак томонлари,  $p=(a+b+c)/2$ . Демак бу масалани ечишда ҳам процедура сифатида учбурчак юзасини ҳисоблаш дастурини олишимиз керак:

```

Program Phosh
Var AB,BC, CD, DA, AC, SI, S, a, b,: real;
Procedure str1
  begin p: =(a+b+c)/2;
    S: Sqrt (p*(p-a)*(p-b)*(p-c));
  end
begin
  read (AB,BC, CD, DA, AC);
  a: =AB; b: =BC; c: =AC; str1; S1: =S;
  a: =DA; b: =AC; c: =CD; str1; S1: =S1+S;
  Write (S1);
end.

```

Бу келтирилган дастурда ҳам Str1 номли процедурага икки маротаба мурожаат ёилинмоёда. Хар бир чаъирилишдан олдин процедура таркибида иштирок этаётган  $a$ ,  $b$ ,  $c$  шъгарувчиларга ыйматлар шълаштирилмоёда ва процедурага мурожат этилгандан кейин  $S$  орыали натижа олинмоёда.

Мисолдан кшриниб турибдики дастур билан процедура шртасида алоёа  $a$ ,  $b$ ,  $c$ ,  $S$  шъгарувчилар орыали амалга оширилмоёда. Процедура ичида ички ёрдамчи  $P$  шъгарувчи ҳам иштирок этмоёда.

Паскал тилида ички шъгарувчиларни процедурани шъида кшрсатиш мумкин.

```

Масалан:
Program str2;
Var p:real;
begin
  P: =(a+b+c)/2
  S: = Sqrt (p*(p-a)*(p-b)*(p-c))
end.

```

Процедура ичида аниёланган шъгарувчини “локал” шъгарувчи деб аталади. Агар  $p$  шъгарувчи локал бшлса,  $P$  ташыи шъгарувчи билан хеч ёандай боёлиёликга эга бшлмайди, чунки процедурага мурожат ёилиниши билан  $P$  ёшлланиб чиёишда яна  $P$  эсидан чиёаради. Худди шуни сингари локал метка ишлатиш ҳам мумкин.

Агар дастур таркибида Str1 ва Str2 процедуралари иштирок этган бшлса асосий дастур таркибидаги  $p$  ни аниёланиши олиб ташлаш мумкин, ёолдириш ҳам мумкин.

Юёйоридаги келтирилган дастурда процедурага мурожат ёилишдан олдин процедура иштирок этувчи шъгарувчиларга ыймат шълаштирилди. Бу операцияни шрнига процедурадан параметри (аргументлиги) дан фойдаланиш мумкин. Масалан:

Str3 (AB, BC, AC, S1)

Бундан мурожат учун процедура кўриниши ыуйидагича бўлиши керак.

```

Procedure str3 (a, b, c, s: real);
Var P: real;
begin
  P: =(a+b+c)/2;
  S: = Sqrt (p*(p-a) *(p-b) *(p-c));
end.

```

Процедура номидан кейин ыавус ичида ёзилган параметрлар a, b, c, d, s лар формал параметрлар процедурага мурожат ыилиниши формал параметрлар фактик параметрлар билан алмашади. Уларни мослашуви кўрсатилган тартибда кучаяди.

Юыоридагилар асосида иккинчи дастурни ыуйидагича ёзиш мумкин:

```

Program F2 (input, output);
Var AB,BC, CD, DA, AC, SI, S, a, b: real;
Procedure Str3 (var a, b, c, s: real);
Var p: real
begin P: =(a+b+c)/2;
  S: = Sqrt (p*(p-a) *(p-b) *(p-c));
end;
begin
  read (AB, BC, CD, DA, AC);
  Str3 (AB, BC, AC, S1);
  Str3 (CD, DA, AC, S2);
  Write (S1+S2);
end.

```

Демак умумий холда процедурани структурасини ыуйидагича ифодалаш мумкин.

```

Procedure
    процедура номи (параметрлар)
label
    меткалар
Const
    констаталарни келтириш
type
    берилганлар типини аниылаш
var
    щзгарувчиларни келтириш

“ички процедура ва функциялар”
begin
    процедура жисми

```

end.

Процедура номидаги ыавус ичидаги баози параметрлар var сщзи бщлмаслиги ьам мумкин. Масалан:

```
Procedure Str4 (a, b, c: real; var S: real);
```

```
Procedure prim (k: integer; p: integer);
```

Параметрлар олдида Var бщлмаганларини “формал параметр ьийматлари”, var бщлган параметрларни “формал параметр-щзгарувчи” деб аталади.

Биринчи холдаги формал параметр-ьийматда процедурани чиьаришда параметр щрнида бщлмасдан уни щрнида ихтиьрий ьиймат, ёки ифода ьам бщлиши мумкин. Фаьат ьиймат ва ифода натижаси кщрсатилган типиде бщлиши лозим. Формал параметр-щзгарувчи холда процедурага мурожат ьилинганда щзгарувчи бщлиши шарт.

Буларни эотиборга олиб мисолни ьуйидагича щзгартириш мумкин:

```
Procedure Str4 (a, b, c: real; var S: real);
```

```
Var p: real;
```

```
begin
```

```
  P: =(a+b+c)/2;
```

```
Program prob;
```

```
Var x, y: real
```

```
S = Sqrt (p*(p-a)*(p-b)*(p-c));
```

```
end;
```

```
begin
```

```
  read x;
```

```
  Str4 (3, 14, x sqrt(10-sqr(x)), y);
```

```
  Writeln (y);
```

```
end.
```

### **Назорат саволлар.**

1. Процедуралар аъамиятини тушунтиринг.
2. Процедураларга таориф беринг.
3. Процедура параметрларини тушунтиринг.
4. “Локал” щзгарувчига таориф беринг.
5. Поддастурга таориф беринг.
6. Процедурани структурасини тушунтиринг.
7. “Глобал” щзгарувчи деганда нимани тушунасиш?
8. “Статус” щзгарувчини тушунтиринг.
9. Ёайта номлаш операциясини тушинтиринг.
10. Ёаторли катталиклар устида операциялар деганда нимани тушинасиш?

## 14 - М А О Р У З А

### Паскалр алгоритмик тилида функциялар.

#### Режа.

1. Функциялар ва уларни ташкил этиш.
2. Функциялар параметрлари.

**Таянч иборалар:** *Функциялар, уларни ташкил этиш, функциялар параметрлари, процедуралар аъамияти, процедуралар, процедура параметрлари, “локал” иъзгарувчи, поддастур, ыйта номлаш операцияси, функция структураси.*

Паскал тилида процедуралардан ташыари функциялар ташкил этиш имконияти бор. Функцияга мурожат ыйлинса кшрсатилган типдаги ыйймат олинади.

Функцияларни процедурадан иккита асосий фарыи мавжуд. Биринчи фарыи шундаки функция “funcfOn” сшзидан бошланиши ва уни типини белгилаш билан тугаши керак. Масалан:

```
funcfOn f (a: real; var b: t): real;
funcfOn g (var a, b: integer): integer;
funcfOn h (a: integer): char;
```

Иккинчи фарыи функцияни аниылланишида функция номидан кейин шзлаштириш белгиси := ышлланиши керак.

Масалан:

```
f: = 3.14
g: = a+2*b
```

Функциядан фойдаланиб юыоридаги мисолни ыуйидаги дастурлаштириш мумкин:

```
Program F3 (input, output);
Var AB, BC, CD, DA, AC: real;
    funcfOn trig (a, b, c: real): real;
Var P: real;
begin
    P: =(a+b+c)/2;
    trig: = Sqrt (p*(p-a) *(p-b) *(p-c));
end;
begin
    read (AB, BC, CD, DA, AC);
    Write (trig (AB, BC, AC) + trig (CD, DA, AC))
end.
```

Мисолдан ҳам кўриниб турибдики функцияни белгиланиши дастурда катталикларни белгилаб олингандан кейин туради. Функция ичида шунини локал шзгарувчилари ҳам бериши мумкин (процедура каби). Лекин шунини унутмаслик керакки функцияга мурожаат этилганда функция бийматти олинади. Функция номи шзида биймат туради. Умумий кўринишда функция структурасини тутидагича ифодалаш мумкин:

FuncOn функция номи (параметрлари): тип

label

меткалар

Const

доимийларни келтириш

type

берилганлар типини аниылаш

var

шзгарувчиларни келтириш

“процедура ва функцияларни келтириш”

begin

функция танаси

end

Мисол:

```
Program funcOn;
```

```
  Var x, rez: real;
```

```
  funcOn Func: real;
```

```
    Var R: real;
```

```
    begin
```

```
      R: = sin (x) * sin (x);
```

```
      func: =R;
```

```
    end;
```

```
begin
```

```
  Writeln ('Раъамни ёз');
```

```
  Readln (x);
```

```
  Rez:=Func;
```

```
  Writeln ('Синус квадрати тенг' Rez);
```

```
end.
```



**Назорат саволлар.**

1. Функцияга таориф беринг.
2. Уларни ташкил этиш ыандай бщлади?
3. Процедуралар аъамиятини тушунтиринг.
4. Процедураларга таориф беринг.
5. Процедура параметрларини тушинтиринг.
6. “Локал” щзгарувчига таориф беринг.
7. Поддастурга таориф беринг.
8. Процедурани структурасини тушунтиринг.
9. Функциялар параметрлари деганда нимани тушинасиз?
- 10.**Функция структурасини тушунтиринг.

## 15 - М А О Р У З А

### Турбо-Паскалда модуллари.

#### Режа.

1. Модуллар.
2. Модулларни ташкил этиш.
3. Мисоллар.

**Таянч иборалар:** *Модуллар, модулларни ташкил этиш, мисоллар, System ни вазифаси, Printer ни вазифаси, Crt ни вазифаси, Graph ни вазифаси, Delay процедурасинини вазифаси, sound процедурасини вазифаси, Nosound процедурасинини вазифаси, Keypressd процедурасини вазифаси.*

Турбо-Паскалда кичик зарурий дастурларни тузиб олдиндан компиляция ёилиниб тайёр модул сифатида фойдаланиш йўлга ёйилган. Турбо-Паскал алгоритмик тилини асосий ютувларидан бири ҳам шудир. Улар таркибига персоналр компютерларни барча имкониятларидан фойдаланиш имкониятини берувчи процедура ва функциялардир. Модулларга мисол ёилиниб ёуйидагиларни кўрсатиш мумкин.

System - у Паскал тилида ёйланувчи стандарт процедура ва функцияларни шўз ичига олади. У барча дастурлар учун ёйланганиши мумкин.

DOS - Ms-DOS операцион системаси мухитидан ишловчи процедура ва функцияларни шўз ичига олади.

Crt - экран, клавиатура ва компютер динамикаси билан ишлайдиган процедуралар йиьиндиси.

Graph - компютерларни график имкониятларидан кенгроь фойдаланиш имкониятини берувчи процедуралар.

Printer - принтер билан ишлашни осонлаштирувчи процедура.

Graph3 - Турбо-Паскалр 3.0 версиясида ишловчи график процедуралари.

System модули автоматик барча холларда муружат ёилиш имконияти бор.

Бошья модулларни ва фойдаланувчи ташкил этган модуллардан фойдаланиш учун дастур таркибида дастур номи сатри.

Program дастур номи

дан кейин ёуйидаги сатр ёйшилади

uses модул номи.

Агар дастурда бир неча модуллардан фойдаланиш зарур бўлса сатрда барча модулларни кўрсатиш мумкин.

uses модул 1 номи, модул 2, ... модул N.

Паскал тилида модуллардан дастур таркибида фойдаланишдан маъсад, дастур эффективлигини ортиради.

Масалан овоз чиьарувчи дастур тузиш учун CTR модулидаги процедура ва функциялардан фойдаланади.

Sound (i) - процедураси динамикасини улайди ва i частота билан овоз беришни таоминлайди.

Delay (i) - процедураси дастур бажарилишини i милли секундга ортиради.

Nosound - процедураси динамикни щчиради.

Keypressed - функцияси бир клавиш босилса True акс холда False ыййматни беради.

Дастур тузишда System модули таркибидаги Random процедурасидан фойдаланамиз.

Random (i) о ва 1 сонлари орасида тасодикий сонларни беради. Дастур ыуйидаги кщринишга эга бщлады:

```
Program spring;
uses CRT;
begin
Repeat
    sound (1400+Raund (600));
    delay (Raund (10));
    nosound;
    delay (Raund (1300));
until keyppressed;
nosound;
end.
```

```
Unit Study;
interface
    functIOn min(x,y:integer):integer;
    functIOn max(x,y:integer):integer;
```

```
implementatIOn
```

```
    functIOn min(x,y:integer):integer;
        if x<=y then min:=x else min:=y;
    end;

    functIOn max(x,y:integer):integer;
        if x>=y then min:=x else min:=y;
    end;
end.
```

Ёзишмани модулга айлантириш учун уни компиляцияланиши керак. Компиляцияланиш натижасида STUDY. TPU модули ьосил бщлиши керак. Чунки дастур номини Unit Study деб номланган. TPU - файл номини файл номи ёзилган сатрдан олади.

Агар дастур бошланьич тексти номи Myfile.Pas бщлса компиляцидан кейин Myfile.Tpu бщлады.

Study модулидаги Min ва Max функцияларни фойдаланувчи дастурни ёзувчида ёзиш мумкин.

Айтайлик Myfile.Int файлидан бериши 100 та сон чиқарилади. Дастур массивидаги сонларда энг кичик ва энг каттасини экранга чиқаради

```
Uses Crt,Study;
var
    m:array[1..100] of integer;
    f: file of integer;
    min_,max_,i:integer;
begin
    assign(f,'myfile.int');
    reset(f);
    for i:=1 to 100 do
        read(f,m[i]);
    min_:=m[1];
    max_:=m[1];
    for i:=1 to 100 do
        begin
            min_:=min(min_,m[i]);
            max_:=max(max_,m[i]);
        end;
    Sound(300);
    Delay(100);
    NoSound;
    WriteLn('min=',min_,'max=',max_);
end.
```

Sound, Delay, No sound - процедуралари CRT модулидан олинган. Дастур бошланишида функция ва процедуралари фойдаланиладиган модулар кичик қаторда келтирилган.



## Назорат саволлар.

1. Модулларни тушунтиринг.
2. Модулларни ташкил этишни тушунтиринг.
3. Модулларга мисоллар келтиринг.
4. System ни вазифасини тушунтиринг.
5. Printer ни вазифасини вазифасини тушунтиринг.
6. Crt модулинини вазифасини тушунтиринг.
7. Graph модулинини вазифасини тушунтиринг.
8. Delay процедурасинини вазифасини тушунтиринг.
9. Sound процедурасини вазифасини тушунтиринг.
10. Nosound процедурасини вазифасини тушунтиринг.

## 16 - М А О Р У З А

### Турбо-Паскалр интерфейсидан фойдаланиш

#### Режа.

1. File менюси.
2. Edit менюси.
3. Run менюси.

**Таянч иборалар:** *File менюси, Edit менюси, Run менюси, Search менюси, Compile менюси, Debug менюси, tools менюси, Options менюси, window менюси, help менюси.*

File менюсида янги файл очиш, дискга ёзиш, каталог очиш, дискдаги дастурларни экранга чиыариш, умуман, файллар устида аммалар бажарилади:

- New - янги файл очиш.
- Save-F2 - дискка ёзиш.
- Open-F3 - Дискдаги дастурларни экранга юклаш.
- Save As - директорияларни шзгартириш.
- Change Dir - директория шзгартириш.
- Print - принтерга чиыариш.
- Printer Setup - фойдиланилаётган принтер ыурилмасини параметрларни шзнатиш

- Os Shell - операцион Супер глупер система ыобиыи бшчиб, ваытинча операцион системанинг барча ташыи ва ички буйруыларни бажариш учун тайинланган.

Барча операциялар бажариб бшлингандан сшнг TP га ыайтиш учун Exit буйруидан фойдаланилади.

- Exit-Alt +X - TP дан DOS га ыайтиш, ягона TP нинг иш фаолиятини якунлаш учун фойдаланилади.

Тахрирлаш менюси текст бшлаги бир ыанча операцияларни бажариш имкониятини беради.

- Undo - киритилган шзгартиришларни бекор ыилиш. Тахрирланаётган маолумотни башланьич.

- Redo - киритилган шзгартиришлар тиклаш. Undo командасида бекор ыилинган шзгартиришларни ыайта тиклаш.

- Cut - текстни ажратилган ыисмини олиб ташлаб, буферга жойлайди.

- Copy - нусха олиш.

- Paste - ахборот ышчиб ышйиш, курсор турган жойдан бошлаб, ахборотни буфердан жорий файлга олади.

- Clear - Маолумотни ыисмини шчириш.

Show Clipboard - буфер ойнасини очеди.

TP тилидаги дастурларни ишга тушириш менюси бщлиб, у ыуйидаги командаларни щз ичига олади.

Ctrl+F9 - дастурни тщлиы ишга тушириш.

To Corsor - F4 - дастурларштиришни тахрирлаш,

силлиылаш жараёнида кенг ишлатилади. Бу

дастур

орыали дастурни курсор

жойлашган ыисмигача жойлаштириш имконини

беради.

Step - F8 - дастур ыадамма-ыадам ишга тушади.

Trage - F7 - бу ьам дастурни силлиылаш учун

фойдаланиб, ыисм дастурларни ьам

бажарилишини, дастурларнитщърилигини

текширади.

Compile - дастурни компиляция ыилиш ва бажарувчи  
файллар яратиш.

Project - агар дастур бир неча файллардан иборат  
бщлса, проектда кщрсатилади.

OptIOns - компиляция параметрларни щрнатиш.

Debug - дастур бажарилаётган пайтда

щзгарувчилар функцияларини ыийматларини

кщриш ва уларни щзгартириш имкониятини

яратиш.

Bzeak/Watch - менюда дастурни тщхтатиш нуыталарини

ыщщиш ва олишни ташлаш мумкин.





**Назорат саволлар.**

1. File менюсини тушунтиринг.
2. Edit менюсини тушунтиринг.
3. Run менюси ни тушунтиринг.
4. Search менюсини тушунтиринг.
5. Compile менюсини тушунтиринг.
6. Debug менюсини тушунтиринг.
7. Tools менюсини тушунтиринг.
8. Options менюсини тушунтиринг.
9. Window менюсини тушунтиринг.
10. Help менюсини тушунтиринг.

## **АДАБИЁТЛАР:**

- 1.** С.А. Абрамов, Е.В. Зима “Начало программирования на языке Паскал”, М., Наука, 1987г.
- 2.** А. Файсман “Профессиональное программирование на Турбо-Паскал, Info-Ihfomex-Koinco”, 1992 г.
- 3.** О.Н. Перминов “Язык программирования Паскал”, М., Радио и связь, 1982 г.
- 4.** К. Боон “Паскал для всех” М., Энергоиздат, 1988 г.

## М У Н Д А Р И Ж А :

1. Кириш. Алгоритмик тиллар ва дастурларни асослари _____	4
2. Компрютерларда информацияларни ифодалаш ____	7
3. Персонал компютерлар асосий ыурилмалари _____	9
4. Машина кодлари ва тцхталишлар _____	12
5. Операцион Супер глупер система ва файллар билан ишлаш _____	15
6. Турбо-Паскалр ва унинг модуллари _____	17
7. Катталиклар ва уларнинг турлари _____	20
8. Массивлар ва тццпламлар _____	25
9. Паскалр алгоритмик тилида операциялар _____	31
10. Мураккаб операциялар _____	33
11. Цикл операциялари _____	35
12. Паскалр тилида дастур структураси _____	37
13. Процедура ва функциялар _____	39
14. Паскалр алгоритмик тилида функциялар _____	43
15. Турбо-Паскалр модуллари _____	45
16. Турбо-Паскалр интерфейсидан фойдаланиш _____	48
17. Адабиётлар _____	50