

O'ZBEKISTON RESPUBLIKASI
OLIY VA O'RTA MAXSUS TA'LIM VAZIRLIGI

TERMIZ DAVLAT UNIVERSITETI

FIZIKA- MATEMETIKA FAKULTETI

AMALIY MATEMATIKA VA INFORMATIKA KAFEDRASI

Matematika bo'limi talabalari uchun
informatikadan ma'ruzalar

(I-semestr uchun)

Tuzuvchi :

O`q .Mengliyev Sh. X.

Termiz 2008

MAVZU 1: KOMPYUTER SINFLARI, ISHLASH PRINSIPLARI VA TASHKIL ETUVCILARI

Reja

- 1. Informatika fani, uning mazmuni va vazifalari. Informatika fani predmeti haqida*
- 2. Kompyuter haqida umumiy ma'lumot*
- 3. Kompyuter sinflari*
- 4. EHM ning tuzilishi va ishlash tamoyili*
- 5. Xulosa*

Foydalanilgan adabiyotlar

Tayanch so`z va iboralar: axborot, super kompyuter, katta kompyuter, shaxsiy kompyuter, Noutbook kompyuterlari, minikompyuterlar arifmetik– mantiqiy qurilma, boshqaruvchi qurilma

Informatika fani, uning mazmuni va vazifalari

Informatsion jamiyatni barpo etish borasida respublikamizda keng kulamda ishlar olib borilmokda. Buning uchun ukuvchilarning informatsion madaniyatini oshirishga katta axamiyat berilayapti. «Informatika» fani esa talabalarning bilim va malakalarini kengaytirishga asosiy ukuv kullanma xisoblanadi. Mazkur fanning asosiy vazifasi jamiyatning a'zolari, xususan ukuvchi va talabalarni zamonaviy axborot va algoritmik madaniyat soxibiga aylantirishdan iborat. Bunda ta'lim jaraeni zamonaviy kompyuterlarda mavjud bulgan dasturiy vositalar erdamida olib borilishi lozim.

Informatika - informatsiya (xabar, axborot, ma'lumotlar) ni jamlash va kayta ishlashning usullarini urganadi.

Informatika fan sifatida axborotlash jaraenlari konuniyatlarini urganadi. Informatsion jaraen keng tushuncha bulib, ma'lumotlarni jamlash, uzatish, saklash, tuplash, kidirish va iste'molchiga berishgacha bulgan jaraenlarni uzida jamlaydi.

Informatika atamasi XX asrning 60-yillarida ishlatila boshlangan bulsada, uning aloxida fan sifatida ajralishi 40-50 yillarga tugri keladi. Bu davrda kupchilik tadkikotlar axborot kidirish muammolari bilan uzviy boglik edi. Ayni shu davrda axborotlarni kidirish tizimi va usullari yaratildi.

Informatikaning yuzaga kelishi va rivojlanishida xisoblash texnikasi va boshka texnik vositalarning urni bekies, chunki axborotlar bevosita xisoblash texnikasi kumagida kayta ishlandi va bu fan uzining xususiy, yangi, nostandart uslub va usullariga egadir.

Informatsiya suzi lotincha «information» suzidan olingan bulib, axborotlarni tushuntirish va taxlil kilish degan ma'noni bildiradi.

Informatsiya suzi xabar kurinishida buladi. Xabar - bu informatsiyaning suzlashuv, matn, tasvir, jadval, sonli ma'lumotlar va x.k. kuriishidagi turidir. Xabarlar mazmuni bilan kizikuvchilar ma'lumotni iste'mol kiluvchilar deb yuritiladi.

Ma'lumot turlari biologik, sotsial va elementar bulishi mumkin. Kishilik jamiyatidagi axborotlar - sotsial, usimliklar va xayvonot dunesidagi axborotlar biologik, tabiatdagi boshka axbortlar - elementar ma'lumotlar deyiladi.

Ma'lumotning uchta asosiy xossasi mavjud: atributiv, pragmatik va dinamik. Ma'lumotning atributiv xossasi shundayki, uningsiz informatsiya mavjud emas, pragmatik xossasi – ma'lumotni amaliyot uchun qullanilish darajasini belgilaydi, dinamik xossasiuni vakt buyicha uzgarish jaraenini belgilaydi.

Informatika fanining rivojlanish ma'lumotlarni jamlab, qayta ishlash imkoniyatiga ega bulgan kompyuterlarning dunega kelishi bilan bevosita boglik. Chunki ma'lumotlarni kaytap ishlashning axamiyatga molik kismi shunday avtomatik kurilmalarga yuklanyapti-ki, ular inson ishtirokisiz uzok muddat davomida ma'lumotlarni bir necha million marta tezrok kayta ishlash imkoniyatiga egadir.

Kompyuter (inglizcha computer - «xisoblovchi» ma'nosini bildiradi)ning yaratilishi, xalk xujaligining turli soxalarida ishlab chikarish texnologiyalarini tubdan uzgartirish imkoniyatini beradi. Bu esa uz navbatida zamonaviy

xisoblash texnikasidan unumli foy-dalanishga va «Informatika» fanini mukammal urganishga chorlaydi.

Ma'lumot yoki axborot tarixan moddiy va ma'naviy boyliklar qatoridagi qadriyatlardan bo'lib kelgan. Tinch hayot davrida xom ashyoni qayta ishlash, inshootlarni puxta qilib yaratish, tabiat injiqliklariga bardosh bera olishga doir tajriba xulosalari yozma yoki og'izdan-og'izga ko'chuvchi ma'lumot, oila, qabila millat-elatlarning mavqeini belgilovchi manba va boylik sifatida qadrlangan. Urush yoki tahlikali kunlarda esa dushman qurolli kuchlari, rejalari, mudofa imkoniyatlari haqida ma'lumot hayot-mamot masalasi bo'lgan. Shu bois ma'lumotga nisbatan har doim uni saqlash, tezkorlikda uzatish va to'g'ri tahlil qilish kabi masalalar dolzarb bo'lib, kelgan. Masalan, ma'lumotni qulay va ishonchli saqlash maqsadida qog'oz ixtiro qilingan, tezkorlikda va ta'sirchan uzatish telegraf, telefon, radio, televidenie ish olib brogan.

To'g'ri va tezkor tarzda katta xajmdagi ma'lumotni qayta ishlash maqsadida esa kompyuter ixtipo qilingan, deyish mumkin. Ishlab chiqarish kuchlari imkoniyatlari hamda fan-texnika yuqori cho'qqilarga ko'tarilgan zamonda xam ma'lumot yoki axborot o'ta muxim ahamiyatga ega tovar sifatida namoyon bo'ladi.

Endi yangi ma'lumot yoki bilimlarni yaratuvchi bir qator mutaxassisliklar mavjudki, muayan shaxs, tashkilot, tarmoq hatto davlatlar taqdiri va salohiyati ulardan o'z vaqtida olingan sifatli ma'lumotlarga bog'liq desak mubolag'a bo'lmaydi. Bu mutaxassislarni kuch qudrati, bir tomondan, o'z sohalaridagi yuqori malakasi bolan belgilansa, ikkinchi tomondan, hisoblash mashinalari (kompyuterlar) zamonaviy axborot texnologiyalarini o'zlashtirganliklarida

namoyon bo'ladi. Haqiqatan ham, kompyuter, aniqrog'i u va unga ulanadigan nihoyatda kengaytiradigan yordamchi qurilmalar majmuasi quyidagi sxemaga ko'ra ma'lumotni qayta ishlaydi, o'zidan o'tkazadi:

Axborot —→ kompyuter —→ axborot

Ko'p hollarda kompyuterga kiritiladigan bilimlar yoki ma'lumotlar bazasi sifatida namoyon bo'ladi, unda hosil qilingan axborot esa o'z iste'molchisiga ega bo'lgan yuqori baxolarga ega tovar sifatiga qadrlanadi. Kompyuterlar

imkoniyatlarining takomillashuvi hamda ishlab chiqarish va hayotning turli sohalarida intensive tarzda kirib borishi mos fan sohasini jumladan, informatika fani predmetining bir necha bor tubdan yangilashiga olib kelgan. Hozirgi kunda kompyuter va unga mos dasturiy taminot bazasi imkoniyatlarini bu fanni o'qitishda har bir soha mutaxassislariga muayyan bilim va ko'nikmalar majmuasini berish imkonini beradi va taqozo qiladi. Bu yo'sindagi majburiylik kompyuter unga ulanadigan qurilmalar majmuasi hamda mavjud amaliy dasturlar zaxirasi katta bo'lgani uchun bir kishi tomonidan to'laqonli o'zlashtirilishi nihoyatda mushkul masala ekanligidan kelib chiqadi. Shu bois biz fani dasturini mutaxassisliklarga ko'ra uch turga ajratishni taklif etamiz: fizika –matematika, mexanika va muxandislik kasblari, tabiiy fanlar, jumladan tibbiyot va iqtisodiyot fanlari hamda gumanitar fanlar uchun alohida.

Kompyuter haqida umumiy ma'lumot

Kompyuter –ingilizcha so'z bo'lib, u hisoblovchi demakdir. U hozirda faqat hisoblovchi bo'lmasdan, mantlar, tovush, video va boshqa ma'lumotlar ustida ham amallar bajaradi. Shunga qaramasdan, hozirda uning eski nomi kompyuter saqlangan, uning asosiy vazifasi turli ma'lumotlarni qayta ishlashdan iborat.

Hozirgi vaqtda kompyuterlarning turli hillari mavjud bo'lib, bular raqamli, analogli (uzluksiz), raqamli analogli, mahsuslashtirilgan, shularning ichida raqamli kompyuterlar

foydalanishi bajaradigan amallarning universalligi, hisoblash amallarining aniqligi va boshqa amallarning yqoriligi uchun bu kompyuterlardan ko'proq foydalaniladi. Xozirgi vaqtda jaxonda kompyuterlarning besh turi ko'proq ish olib boradi.

Kompyuterlarni sinflash

Kompyuterlarni xotirasining hajmi , bir sekunda bajaradigan amallar tezligi, ma'lumotlarning razryad to'rida tasvirlanishiga qarab besh guruhga bo'lish mumkin.

- ✓ Super kompyuterlar
- ✓ Katta kompyuterlar
- ✓ Mini kompyuterlar
- ✓ Shaxsiy kompyuterlar
- ✓ Bloknot kompyuterlar

Xar bir kompyuter o'zining ishlashi, gabariti bilan bir –biridan farq qiladi.

Super kompyuterlar- juda katta tezlikni talab qiladigan va katta hajmdagi masalalarni echish uchun mo'ljallangan bo'ladi. Bunday masalalar ko'p bo'lib bular, ob-havoning global prognoziga oid masalalar , uch o'lchovli fazoda turli oqimlarning kechishini o'rganish, global axborot sistemalariga oid masalalarni o'rganadi. Bu toifadagi kompyuterlar bir sekunda 10 trilliardlab amallarni bajarishga mo'ljallangan.

Reyting	Super kompyuter rusumi	Ishlab chiqa- ruvchi mam-lakat	Firma nomi	Prosesor soni	Tezligi (Gflops)
1	Intel ASCI Red	AQSH	Intel (AQSH)	9472	1338
2	SGI ASCI Blue	AQSH	SGI	6144	634

			(AQSH)		
3	SGI T3E1200	AQSH	SGI (AQSH)	1084	430
4	Hitachi SR 8000	Yaponiya	Hitachi (Yaponiya)	128	368
5	SGI T3E900	AQSH	SGI (AQSH)	1324	264

Katta kompyuterlar- Fan va texnikaning turli sohalariga oid masalalarni echishga mo'ljallangan. Ularning amal bajarish tezligi va xotira hajmi superkompyuterlarnikiga qaraganda bir- ikki pog'ona past. Bularga misol sifatida AQSHning CRAY, IBM 390, 4300, IBM ES/9000, Fransiyaning Borrous 6000, Yaponiyaning M1800 rusumli kompyuterlari misol bo'la oladi.

Minikompyuterlar- kichik kompyuterlar hajmi va bajaradigan amallar tezligi jihatidan katta kompyuterlardan kamida bir pag'ona pastir. Shuni aytish joyizki, ularning o'lchamlari ixchamlashib, hatto shaxsiy kompyuterlardek kichik joyni egallaydiganlari yaratilmoqda. Bunday kompyuterlar turkumiga ilk bor yaratiladigan PDP-11 (Program Driver Processor –dasturiy boshqaruv prosessori) turkumini, ilgari harbiy maqsadlar uchun ishlatilgan.

Shaxsiy kompyuterlar- hozirgi vaqtda korxonalar, tashkilotlar, oily o'quv yurtlarida keng tarqalgan bo'lib, ularning aksariyati **IBM** rusumiga mos kompiyuterlar deganda, ularning turli kompaniyalar ishlab chiqarishiga qaramay, ham texnik, ham dasturiy ta'minoti mosligi, ya'ni bir-biriga to'g'iri kelishi nazarida tutiladi. Bunday kompiyuterlar hajmi jihatidan kichik (bir stol ustiga joylashadi), amal bajarish tezligi, masalan PENTIUM-3 MMX prosessori o'rnatilgan kompiyuterlarida xozirgi kunda 750-1000 megagersni, xotira hajmi esa, 64-128 megabaytni tashkil qiladi. Bu ko'rsatkichlar tez o'zgarib, xar ikki yilda

kompiyuterlar imkoniyati ikki barabar oshishi, ularning narxi esa shuncha arzonlashish yo'nalishi kuzatilmoqda. Bugungi kunda Pentium IV kompiyuterlari ham jahon bozorida keng tarqalmoqda. IBM PC kompiyuterlarini yuzlab fermalar ishlab chiqarmoqda. Bular IBM, Compaq, Hewlett-Packard, Packard Bell, Toshiba, Apple, Siemens Nixdors, Acer, Olivetti, Gateway, SUN va boshqa fermalardir.

Noutbuk kompyuterlar - Noutbok kompyuterlar hajmi ancha ixcham bo'lib, ammo bajaradigan amallar soni, xotira hajmi shahsiy kompyuterlar qatori xisoblanadi, bu kompyuterlarning qulaylik tamonlaridan biri elektr energiyalarida va akkumliyatorda ishlashligidir.

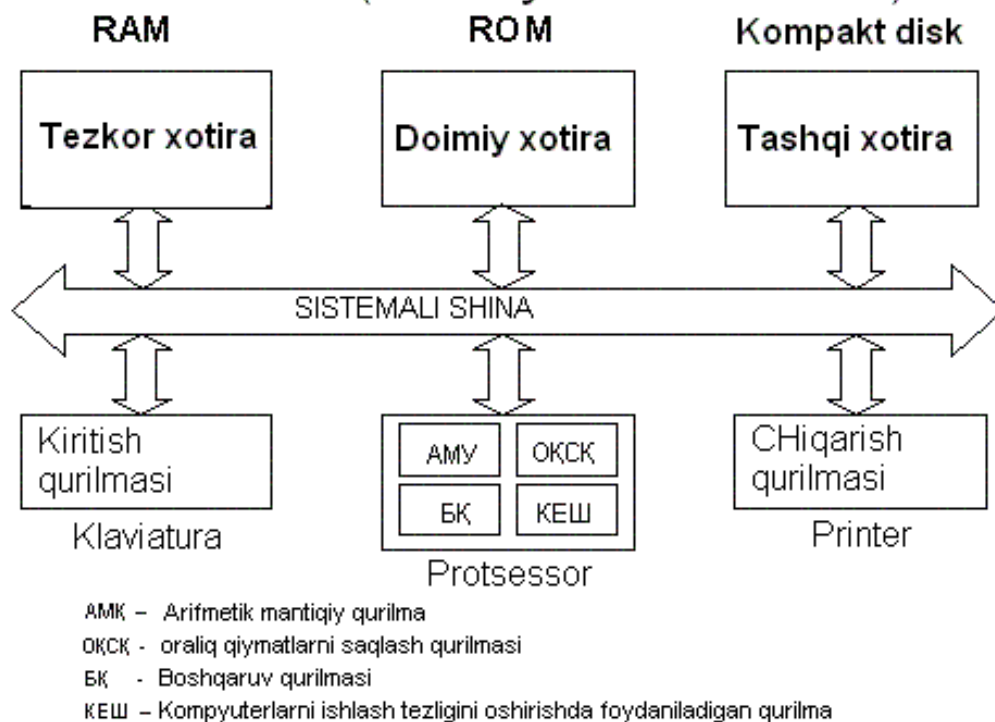
EHM ning tuzilishi va ishlash tamoyili

EHM tuzilishi. Ma'lumotlarni qayta ishlashning elektron avtomati sifatida EHM umumiy boshqaruvga ega qurilma, elementlar murakkab kompleksidir.

Programma va boshlang'ich ma'lumotlarni mashinaga kiritish, zarur amallarni bajarish va natijalarni chiqarish uchun qanday tuzilganidan qat'iy nazar, har qanday EHM o'z tarkibida quyidagi qurilmalarga ega bo'lishi kerak: arifmetik-mantiqiy qurilma (matematik soprotsessor), xotira qurilmasi, boshqaruv qurilmasi, o'qish-kiritish qurilmasi va yozish (chiqarish) qurilmalari bo'lishi kerak.

EHM ning umumiy mantiqiy tuzilishini boshqacha aytganda, EHM arxitekturasi deyiladi. 2-rasmda ifodalangan arxitektura klassik hisoblanadi. U Fon Neyman tomonidan taklif qilingan va u EHM ning barcha xususiyatlarini o'z ichiga oladi.

EHM strukturasi (Fon Neyman arxitekturası)



Arifmetik–mantiqiy qurilma. (AMQ) Bu qurilmaning asosiy vazifasi xotiradan keluvchi ma'lumotni qayta ishlashdir. Zamonaviy EHM lar tarkibiga kiruvchi AMQ(arifmetik–mantiqiy qurilma) lar yuzgacha va undan ortiq amallarni bajara oladi. Oddiy arifmetik amallar - qo'shish, ayirish, ko'paytirish va bo'lishlardan tashqari AMQ ma'lumotni tahrirlash, sonlarni taqqoslash, hisoblash tarmog'ini aniqlash alomatlarini shakllantiradi va hokazo.

Fizikaviy tuzilishi bo'yicha AMQ sof elektron qurilma bo'lib, raqamli ma'lumotlarni juda yuqori tezlikda bajarish imkonini beradi. Zamonaviy EHM lar arifmetik va mantiqiy amallarni sekundiga millionlab bajaradi. Bu yuqori tezlikni amalga oshirish uchun qurilmalar o'rtasida ma'lumot almashish to'liq avtomatlashuvi kerak.

Xotira qurilmasi. Hisoblar, oraliq va yakuniy natijalar, boshlang'ich ma'lumotlar va EHM ning barcha ishlarini boshqaruvchi ma'lumotlarni qandaydir muddatga eslab qolish zarur. Shu sababli ixtiyoriy EHM xotiraga eslab qoluvchi qurilmaga ega bo'ladi.

Zamonaviy EHMlarda turli xil xotira qurilmalari mavjud.

Eng tez «eslovchi» xotira–joriy(operativ) xotiradir. Joriy xotirada ayni paytda aniq bir masalani yechish uchun zarur bo'lgan ma'lumotlarga saqlanadi. Ma'lumotlarni uzoq muddatga saqlash uchun nisbatan sekin tezlikda ishlovchi va katta hajmdagi tashqi xotira qurilmalari ishlatiladi.

Tashqi deb atalishiga sabab, bunday xotiralar joriy xotiradan qurilma nuqtayi nazaridan ajratilgan (disketa, vinchester, CD va hokazo).

EHM ishlashida ma'lumotlarning ayrim qismlari o'zgarmay qoladi. Masalan, ko'p ishlatiladigan fizika o'zgarmlari, jadvallar, maxsus funksiyalar koeffitsientlari, standart

programmalar va hokazolar doimiy eslab qoluvchi qurilmalarga yozilishi mumkin va undan faqat katta tezlikda o'qish mumkin. DQ(doimiy eslab qoluvchi qurilmalar) ga ma'lumot elektrik yoki mexanik usulda DQ yaratilish paytida yoziladi. Ayrim DQ lar qayta yozilishi mumkin. Masalan, SHEHM larda DQ da EHM ishga tushganda qurilmalarni tekshiruvchi va boshlang'ich yuklashni amalga oshiruvchi programmalar yozilgan bo'ladi.

Boshqaruvchi qurilma(BQ). Bu qurilma amallarni bajarilish tartibini aniqlaydi, EHM ishini koordinatsiya qiladi va mashinaning alohida qurilmalari orasidagi o'zaro munosabatlar «diplomatiyasini» olib boradi. BQ boshqaruvidagi har bir tugallangan elementar harakatni bildiradi: ikkita soni qo'shish, ikkita kattalikni solishtirish va hokazo. Aynan BQ AMQ ni aniq bir amalni bajarishga sozlaydi, xotiradan zarur bo'lgan ma'lumotlarni oladi, amal bajarilgandan keyin natijalarni xotiraga yozadi yoki o'qish-yozish qurilmasiga jo'natadi va nihoyat BQ programma bajarilishining keyingi qadami haqida qaror qabul qiladi.

BQ va AMQ munosabati shu darajada uzviy bog'liq va murakkabki, bu ikki qurilma bir guruhga birlashtirilib protsessor deb nomlanadi.

O'qish-yozish qurilmasi. (O'YOQ) O'YOQ odamni mashina bilan muloqot vositasidir. O'YOQ yordamida odam (inson) mashinaga programma va programmani boshqarish uchun zarur ma'lumotlarni va qulay ko'rinishda natijalarni oladi: jadval, grafika, sxema, tovush va hokazo.

EHM ga kiritilishi kerak bo'lgan ma'lumotlar qandaydir oraliq tashuvchilarda saqlanishi mumkin. Oraliq tashuvchilari sifatida perfokaratalar, magnit lentalar, magnit diskleri, egiluvchan disketalar, vinchester, lazer diskleri bo'lishi mumkin. Xuddi shunday EHM natijalari chiqarish (yozish) qurilmalar ro'yxatiga – ekran, printer, grafika quruvchisi, planshetlarni *misol* qilib keltirish mumkin. Disketalar, vinchester, lazer diskleri, modemlar bir paytda ma'lumot chiqarish va o'qish vositalari bo'ladi.

Tayyorlangan ma'lumotni katta tezlikda o'qish va yozish mumkin. Buning uchun EHM da diskovodlar – egiluvchi disketalar uchun, CD-ROM – lazer diskler uchun, printerlar – qog'ozga chiqarish uchun, modemlar – boshqa kompyuterlarga ma'lumot jo'natish qabul qilish uchun va boshqa maxsus o'qish-yozish qurilmalari yaratilgan.

Tizimli shinalar protsessor va EHM ning boshqa qurilmalari orasida ma'lumotlar almashinuvini ta'minlashga mo'ljallangan. 3 xil turdagi shinalar mavjud:

- boshqaruv shinasi,
- ma'lumotlar shinasi,
- adreslar shinasi.

Kompyuterning ishlash prinsipini birinchi bo'lib ingliz olimi Charlz Bebig va uning g'oyasini mukammallashgan ko'rinishini Djon Fon Neyman taklif qilgan. Uning prinsipi dastur asosida boshqariladigan avtomatik ravishda ketma-ket ishlash g'oyasidan iborat. Hozirda ko'p rusumli kompyuterlar shu g'oya asosida ishlaydi, lekin keyingi paytlarda ko'p prosessorli kompyuterlar, yani bir vaqtda dasturning bo'laklarini ketma-ket emas, parallel bajaradigan kompyuterlar ham yaratilganligini eslatib o'tish joyizdir. Shunday qilib kompyuter avvaldan tuzilgan dastur asosida ishlaydi, o'z navbatida dastur qo'yilgan masalani kompyuterda echish uchun qandaydir dasturlash tilida yozilgan buyruqlar operatorlar ketma-ketligidir.

Dasturlash tilida tuzilgan dasturlar mahsus tarjimon dasturlar yordamida kompyuter tiliga o'tkaziladi, kompyuter tili 0 va 1 lardan tashkil topgan, ma'lum qoidalar asosida yoziladigan ketma-ketliklardan iborat.

Djon Fon Heyman prinsipi bo'yicha avtomatik ravishda bajariladigan dastur avval kompyuterning xotirasiga kiritiladi, yuklanadi, xotirada turgan dastur asosida dasturni tashkil etuvchi har bir operator ketma-ket bajaradi.

Boshqaruv qurilmasi deb ataluvchi maxsus qurilma hozir operator bajarilishi va undan keyin qaysi operator bajarilishi ustidan nazorat o'rnatadi va uning bajarilishini taminlaydi. Amal arifmetik-mantiqiy esa posessor deb ataluvchi qurilmada bajariladi. Dastur ishlash natijasi to'g'iridan-to'g'ri ekranda yoki tashqi qurilma, video qurilma va shunga o'hshash qurilmalardan iborat bo'ladi.

Kompyuterlar asosan ikki qismdan tashkil topadi.

1.Hardware- kompyuterlarni tashkil etuvchilar, bular esa o'z navbatida kompyuterlarning qattiq qismlari hisoblanadi.

2.Software- kompyuterlarning dasturiy ta'minoti, bular o'z navbatida kompyuterlarning yumshoq qismlari hisoblanadi.

Nazorat savollari:

1.Informatika fanining predmeti nima?

2.Kompyuterlarning sinflari.

3.Kompyuterning ishlash printsiplari?

4.Ma'lumotni qayta ishlash sxemasi qanday?

5.Kompyuter so'zi qanday ma'noni anglatadi?

6.Super kompyuterlar haqida ma'lumot bering.

7. Kompyuter qurilmalarini ayting.

8. Xotira qurilmasi haqida ma'lumot bering.

MAVZU 2: SHAXSIY KOPMYUTERLARNING QURILMALARI

Reja:

1.Kompyuter qurilmalari

2.Kompyuter xotira qurilmalari

3.Tashqi qurilmalar

4.Xulosa

5.Foydalanilgan adabiyotlar

Tayanch so'zlar va iboralar: sistema bloki, monitor, klaviatura, sichqoncha, tezkor xotira, mikroprotssessor, kesh xotira.

Shaxsiy kompyuterlar asosan quydagi qurilmalardan tashkil topadi.

- ✓ Sistema bloki
- ✓ Manitor
- ✓ Klaviyatura
- ✓ Sichqoncha
- ✓ Tashqi qurilmalar

Sistema bloki –sistema bloki odatda dyesktop(yasi)yoki town (minora)ko'rinishida ishlab chiqariladi. Kompyuterlarning asosiy qismlari sistema blokida joylashgan bo'lib ular quydagilardir.

Tezkor xotira (ram-ixtiyoriy kirish mumkin bo'lgan mikroprosessor, qurilmalar nazoratchilari bular kantroyorlar, adapterlar, elektr manbai bilan taminlash bloki), yumshoq disk qurulmasi,qattiq disk qurulmasi,faqat o'qish uchun mo'ljallangan lazir disk qurulmasi shinalar madem va boshqa qurulmalar.Sestema blogiga uning parallel (LPT) va ketma-ket (COM) portlari orqali ko'plab tashqi qurulmalarni ulash mumkin.

Mikroprosessor. Mikroprosessor kompiyuterning amal bajaradigan qismi bo'lib, ma'lumotlarni berilgan dastur asosida qayta ishlaydi. Mikroprosessor 140 tacha turli arifmetik va mantiqiy amallarni bajaradi. IBM rusumli kompyuterlarda Intel tipidagi mikroprosessorlar ishlatiladi. Bu firma o'z faoliyati mobaynida Intel-8080, 80286, 80386, 80486,Pentium-3, Pentium-4 prosessorlarnigina bozorga chiqarmoqda.

MMX Pentium prosessori. Intel firmasining keyingi avlod prosessori sifatida 1997 yil yanvar oyidan boshlab chiqarilayotgan MMX Pentium prosessorini aytish mumkin. Dastaval, bu prosessor matrisalarni ko'paytirish uchun kengaytma nomi bilan atalgan bo'lsa , keyinchalik MultiMedia Extension- multimedia uchun kengaytma, deb atala boshladi. Bu yangi prosessorni ishlab chiqishdan maqsad, keyingi yillarda ommaviy qo'llanilib borilayotgan kompyuterning multimedia imkoniyatlarini har tomonlama barkamollik darajasiga ko'tarish, multimedia amaliyotlarni tez bajarishni ta'minlashdan iborat. Bu amallar qatoriga, jumladan, multimediami berilganlarini, ikki va o'lchovli grafikalarini tez bajarish kiradi. Shu bilanbirga bu prosessopr ko'paytirish va qo'shish amallarini ko'proq ishlatadigan amaliy dasturlarda amallarni tezroq bajarishga qaratilgan. Shuning uchun ham uni ko'proq matematik sopprosessori talab qilmaydigan, butun sonlar bilan ishlash bilan bog'liq masalalarni yechishda qo'llash maqsadga muvofiq bo'lsada, tajribalar uning har tomonlama ustunligini ko'rsatyadi.

Xotira- Xotira kompyuterda dasturlar va berilganlarni,amal natejalarini saqlaydigan qurilma. Xotiraning turlari ko'p:tezkor, doimiy,tashqi,kesh,video va boshqalar.

Tezkor xotira -kompyuterning muhum qismi bo'lib, prosessor undan amallarni bajarish uchun dastur, berilganlarni oladi va amalni bajarib, natijani yana unda saqlaydi. Shuni alohida ta'kidlash lozimki, kompyutero'chirilsa, tezkor xotirada saqlanayotgan dasturlar va berilganlar yo'q bo'lib ketadi. Shuning uchun ularni qattiq diskda yokidisketlarda saqlab qolish kerak.

Kompyuterishlab turganda elektr tokini ogohlantirmasdan o'chirish, umuman aytganda, katta zarar keltirishi mumkin. Barcha turdagi xotiralar uchun muhim tushuncha uning hajmidir. Kompyuterlarda ma'lumot birligining eng kichik o'lchovi sifatida bayt qabil qilingan bo'lib, 1 bayt 8 bitga teng. O'z navbatida bayt bir simvolni belgini tasvirlaydi. Familiyangizni kompyuterga kiritish uchun familiyangizda nechta harf bo'lsa, u xotirada shuncha bayt joyni egallaydi. Xotira hajmi birligi sifatida kilobayt qabil qilingan va K bilan belgilanadi. O'z navbatida bir kilobayt 1000 baytga teng. 1024 kilobayt esa 1 Mbaytga teng. Xotiraning katta-kichikligiga qarab u yoki bu dasturlar majmuini ishlata olish mumkin. Misol uchun 1Megabaytli kompyuterlarda imkoniyatlari ko'proq bo'lgan dasturlarni, xususan WINDOWS3.1,3.11 dasturlarini ishlatish mumkin. WINDOWS 95 to'laqonli ishlashi uchun 16 Megabayt xotira va kamida 486 prosessorli kompyuter, WINDOWS 98, 2000 operasion sistemasida ishlash uchun, FotoShop, Corell Drawe va boshqa hozirgi zamon dasturlarini ishlatish uchun kamida 32 Mbayt xotirali va PENTIUM-2,3 prosessorli kompyuterlar lozimdir. Bu ko'rsatgichlar yangi prosessorlar va dastur mahsulotlarining paydo bo'lishi bilan yil sayin oshib boradi.

Doimiy xotira. Kompyuterlarda berilganlar unga avvaldan joylashtirilgan doimiy xotira mavjud. Bunday xotiradan faqat o'qish mumkin. Shuning uchun ham u ROM deb ataladi. IBM PC kompyuterlarda bu xotira kompyuter jihozlarini ishlashini tekshirish, operasion sistemaning boshlang'ich yuklanishini ta'minlash, qurilmalarga xizmat ko'rsatishning asosiy funksiyalarini bajarish uchun ishlatiladi

Kesh xotira. Kesh xotira kompyuter ishlash tezligini oshirish uchun ishlatiladi. U tezkor xotira va mikroprosessor orasida joylashgan bo'lib, uning yordamida amallar bajarish tezkor xotira orqali bajariladigan amallardan ancha tez bajariladi. Shuning uchun kompyuter xotirasining ko'proq ishlatiladigan qismi nusxasini kesh xotirada saqlab turadi. Mikroprosessorning xotiraga muroajaatida, avvalo, kerakli dastur va berilganlar kesh xotirada qidiriladi. Berilganlarni kesh xotirada qidirish vaqti tezkor xotiradagiga nisbatan ancha kam bo'lgani uchun kesh xotira bilan ishlash vaqti ancha kam bo'ladi. PENTIUM-2,3

Kompyuterlarda kesh xotira hajmio 512Kni tashkil qiladi.

Videoxotira. Videoxotira monitor ekranida video ma'lumotlarni saqlab turish uchun ishlatiladi. Shuni aytish lozimki, videotasvirlar kompyuter xotirasida ko'p joy egallaydi. Shuning uchun video xotira hajmi qancha katta bo'lsa, shuncha yaxshi bo'ladi. Videoxotiraning 1 Mbaytdan kam bo'lmagani yaxshi.

Shina. Kompyuterda har bir qurilmaning ishini boshqaruvchi electron sxemalar mavjud bo'lib, ular adapterlar deb ataladi. Barcha adapterlar mikroprosessor va xotira orqali berilganlarni ayirboshlovchi magistral yo'l deb ataluvchi shinalar orqali bog'langan bo'ladi. Shunday qilib, oddiy so'z bilan aytsak, shinalar turli qurilmalarni bog'lovchi maxsuis simlardir. Kompyuterda bir qancha shinalar bo'lishi mumkin. Kompyuterlarning electron sxemasi electron

plata deb ataluvchi modullardan iborat. Uning modul tuzilishiga ega bo'lishi kompyuterlar remontini oson bajarish, uni foydalanuvchi ehtiyojiga qarab yig'ish va o'zgartirish imkoniyatini beradi.

Sistema platasi- kompyuterning asosiy platasi hisoblanib, unga bios, mikropro-sessor, tezkor xotira, kesh xotira, shinalar joylashtirilgan bo'ladi. Bundan tashqari, unda bazi bir qurilmalar, ishni bajaruvchi elektron shemalar, klaviatura, disk qurilmalari adapteri ham joylashgan bo'ladi. Hozirgi vaqtda shinalarning PCI/ ISA turi keng ishlatilmoqda, bunday shinalarning ma'lumot ayriboshlash tezligi yuqori bo'lib, u orqali kompyuterga ko'p tashqi qurilmalarni ulash mumkin.

Kompyuterda kiritish-chiqarish portlari kontrolyorlari mavjud bo'lib, ular sistema blokining orqa qismida joylashgan bo'lib u slot deb nomlanadi, bunga sichqoncha, printerlar ulanadi. Kiritish va chiqarish portlari parallel va ketma-ket bo'ladi va ular mos ravishda LPT1-LPT4 va COM1-COM3 deb belgilanadi.

Odatda LPT portga printer va COM portga faks-modem, sichqoncha va boshqa qurilmalar ulanadi.

Monitor- Monitor kompyuterda matn va grafik ma'lumotlarni tasvirlash uchun xizmat qiladi. Monitorlar rangli va rangsiz bo'ladi, kompyuter tarqatadigan nur umuman olganda zararli, shuning uchun ham bazi kompyuterlarda past radiasiya so'zlarini uchratish mumkin, lekin ularning inson organizimiga ta'siri tobora kamayib boradigan rusumlari yarilmoqda. Ohirgi vaqtda ishlab chiqarilayotgan 17-21 dyumli SVGA – katta video adapter monitorlarda nurlarning ta'sirini ancha kamaytirilishiga erishilganligini aytib o'tish mumkin. Tasvirlash qobiliyati ekranning gorizontali va vertikalidagi nuqtalar soni bilan belgilanadi.

Masalan, 14 dyumli monitorida tasvirlash qobiliyati 800x600, 15 dyumli monitorida 1024x768, 17 dyumli monitorida 1280x1024 va 21 dyumli monitorida esa 1600x1200. Tasvirlash qobiliyati 800x600 ga teng bo'lgan monitorlarda yahshi tasvir piksel 0,31mm ga, 1024x768 ga teng bo'lgan monitorlarda esa piksel 0,28 yoki 0,25 ga teng bo'lishi kerak, monitoring tez ishlashi uning adapteriga bog'liq bo'ladi.

Tashqi qurilmalar- quyidagi uskunalardan iborat: Printer, skaner, modem, strimer, grafik quruvchi va boshqalar.

Nazorat uchun savollar:

1. Tezkor xotirada qanday qurilmalar joylashgan?
2. Mikroprotsessorning vazifasi nimadan iborat?
3. Xotira turlarini ayting
4. Videoxotiraning vazifasini tushuntiring
5. Shinaning vazifasini tushuntiring

MAVZU - 3:

AMALIYOT TIZIMLARI VA DASTURIY TA'MINOTLAR

Reja

- 1. Operasion sistemalar**
- 2. Operasion sistema funksiyalari**
- 3. Dasturlash sistemalari**
- 4. Xulosa**
- 5. Foydalanilgan adabiyotlar**

Tayanch soʻz va iboralar: *operatsion sistema, dasturlash sistemalari, MS DOS, fizik resurslar, vinchester, dasturiy resurs*

Kompyuter ishlash uchun zaruriy shart –dasturlarning mavjudligidir.

Dastur ta'minoti 2ta guruhdan iborat:

- sistemaning ishlashi bilan bog'liq sistema dasturi;
- amaliy dasturlar.

Tizim dasturlari kompyuter ishlashi uchun zarur dasturlar bo'lib, u kompyuterning ishlashini boshqaradi, uning turli qurilmalari orasida muloqatni tashkil qiladi. Kompyuterdan foydalanishni osonlashtiruvchi sistema dasturlarning yadrosi operasion sistemalardir. Operasion sistema foydalanuvchi va kompyuter orasida bevosita muloqat o'rnatishni kompyuterni boshqarishni foydalanuvchi uchun qulaylik yaratishni, kompyuter resuslaridan oqilona foydalanish va hokazolarni ta'minlovchi dasturlardir.

Hozirgi paytda turli operasion sistemalar mavjud. Masalan: UNIX, MS DOS, PC DOS, DRD DOS, OS/ 2, WARP, WINDOWS 95, MACINTOSH va boshqalar. Bundan tashqari, xizmat qiluvchi dasturlar mavjud. Ular dastur utilitlari deb ataliob, yordamchi amallarni bajarib, kompyuter ishlashini qulaylovchi dasturlardir.

Amaliy dasturlar predmet sohadan olingan alohida masalalar va ularning to'plamini yechish uchun qaratilgan bo'lib, amaliy masalalarni yechish uchun mo'ljallangan. Bunday dasturlar majmui amaliy dasturlar paketi (ADP) deb ataladi.

Dasturlar odatda magnit yuritgichlarda joylashgan bo'ladi. Ammo operasion sistemalar va u bilan bog'liq dasturlar ancha katta hajmga ega bo'lgani tufayli keyingi paytlarda lazer disklariga yozilmoqda. Ba'zi bir sistemali dasturlar, masalan, kiritish-chiqarishning asosiy sistema

dasturlari to'g'ridan-to'g'ri kompyuterning doimiy xotirasida saqlovchi qurilmasiga yozilgan bo'ladi.

Operasion sistema

Shaxsiy kompyuterlarning operasion sistemalari yaratilish tarixiga nazar solsak, sakkiz razyadli shaxsiy kompyuterlar uchun yaratilgan birinchi operasion sistema CP/M-80 nom bilan tanilgan. Uning muallifi Digital Research kompaniyasining prezidenti Geri Kildell bo'lgan.

16 razyadli yangi kompyuterlar yaratish g'oyasini dasturlar yaratuvchi Microsoft kompaniyasining asoschisi va prezidenti, multimilliarder Bill Geyts ilgari surgan, u IBM firmasi bilan hamkorlikda ishlashga rozi bo'ladi.

Bill Geyts va Pol Allen BASIC dasturlash tili uchun tarjimon dastur yozishdi va u IBM firmasining MITS Altair kompyuteriga moslashtirildi. Shundan so'ng 16 razyadli kompyuterlar uchun operasion sistemalar yaratish jadallashdi va 1981 yilda shaxsiy kompyuterlar uchun birinchi yaratilgan CR/M operasion sistemasining ko'p g'oyalarini o'zida mujassamlashtirgan MS DOS operasion sistemasi 1981 yil avgust oyida paydo bo'ldi.

MS DOS 64 K bayt xotiraga ega bo'lgan kompyuterlarga mo'ljallangan bo'lib, o'zi 8 K bayt xotirani egallar edi. O'sha paytda yetarli deb hisoblangan bunday kompyuter xotirasi hozirgi paytda bir "oinchoqqa" aylandi. Chunki hozirgi zamon shaxsiy kompyuterlarining xotirasi bir necha Gegabaytlarga tenglashdi.

Mualliflar MS DOS ni rivojlantirishni davom etirib, uning MS DOS1.1, MS DOS 1. 25, MS DOS 2.0, MS DOS2-11 versiyalarini taklif etishdi va nihoyat, 1984 yilda MS DOS 3.0, IBM PCAT shaxsiy kompyuteriga 80286 mikroprosessorga asoslangan, 5,25 dyumli diskovodda ishlashga mo'ljallangan operasion sistema yaratildi. 1986 yilda Compaq Computer firmasi 80386 mikroprosessorga asoslangan IBM kompyuterini chiqaradi.

IBM firmasi esa 80386 mikroprosessorga asoslangan PC/2 kompyuterini yaratdi. Bu mikroprosessor asosida yaratilgan kompyuter nazariy jihatdan bir necha Gegabayt xotiraga ega bo'lishi mumkin edi. Ammo MS DOS esa 640K bayt xotiraga ega bo'lgan kompyuterlarga moslashgan edi. Shuning uchun MS DOS sistemasini kengaytirish ishlari davom etardi va 1987 yil MS DOS 3.3 yaratilib, u 3.5 dyumli, ya'ni 1,44 Mbaytli disklar bilan ishlash imkoniyatini berdi. 1987 yili IBM va Microsoft firmasi tomonidan bir vaqtda bir necha masalalar yechishga qodir bo'lgan OS /2 operasion sistemasdi ishlab chiqildi.

Operasion sistema funksiyalari

Operasion sistema bu boshqaruv dasturidir, operasion sistema bu kompyuterning fizik va dasturiy resurslarini taqsimlash va ularni boshqarish uchun ishlatiladigan dastur hisoblanadi,

kompyuter resurslari ikki hil ko'rinishda ish olib boradi bular fizik va dasturiy resurslarga bo'linadi.

Fizik resurslar bu:

- ✓ Xotira
- ✓ Vinchester
- ✓ Monitor
- ✓ Tashqi qurilmalar

Dasturiy resurslar bu:

- ✓ Kiritish va chiqarishni boshqaruvchi dasturlar
- ✓ Kompyuter ishlashini ta'minlaydigan boshqaruvchi dasturlar
- ✓ Berilganlarni tahlil qiluvchi dasturlar
- ✓ Drayverlar
- ✓ Virtual ichki va tashqi xotirani tashkil qiluvchi va boshqaruvchi dasturlar

Dasturlash sistemasi- dasturlash tillari va ularga mos til prosessorlari majmuasidan iborat bo'lib, dasturlarga ishlov berish va sozlashni ta'minlovchi dasturlar to'plamidan iborat. Dasturlash sistemasining tashkil qiluvchilari dasturlar, amaliy dasturlar to'plami singari operasion sistema boshqaruvi ostida ishlaydi.

Operasion sistemalardan qo'yidagi hususiyatlarga ega bo'lish talab etiladi:

1. Ishonchlilik- operasion sistema o'zi ishlayotgan qurilmalar bilan birga ishonchli bo'lishi kerak. Operasion sistema foydalanuvchining aybi bilan vujudga kelgan hatoni aniqlashi, uni tahlil qilishi va tiklash imkoniyatiga ega bo'lishi kerak.

2.Himoya- operasion sistema bajariloyotgan masalalarning o'zaro bir-biriga beradigan ta'sirdan himoyalash kerak.

3.Bashorat- operasion sistema foydalanuvchi so'roviga bashoratchilik bilan javob berish kerak.Foydalanuvchi buyruqlari sistema qabul qilingan qoidalar asosida yozilgan bo'lsa, ularning ketma- ketligi qanday bo'lishdan qat'iy nazar natija bir hil bo'lishi kerak.

4.Qulaylik- Foydalanuvchiga operasion sistemani taklif qilishdan maqsad resurslarni aniqlash va bu resurslarni boshqarish masalalarini echishdan ozod qilishdir.Sistemani inson psihologiyasini hisobga olgan holda loyihalash kerak.

5.Effektivlik- Resurslar taqsimotida operasion sistema foydalanuvchi uchun maksimal holda sistema resurslaridan foydalanish darajasini oshirish kerak.

6.Moslanuvchanlik- sistema amallari foydalanuvchiga qarab sozlanishi mumkin.Resurslar majmuasi operasion sistema effektivligi va samaradorligini oshirish maqsadida ko'paytirilishi yoki kamaytirilishi mumkin.

7.Kengaytiruvchanlik- Evaliyusiya jarayonida operasion sistemaga yangi fizik va dasturiy resurslar qo'shilishi mumkin.

8.Aniqlik- Foydalanuvchi sistema interfays darajasidan pastda sodir bo'ladigan jarayondan behabar qolishi mumkin. Shu bilan birga foydalanuvchi sistema haqida kerakli ma'lumotlarni bilishi mumkin.

Operasion sistemalar ishlashi vaqtida ko'plab resurslarga ega bo'ladi bu resurslar o'z navbatida qo'yidagilar hisoblanadi.

1. Matematik ta'minot resurslari- berilganlar va dastur bajarilishini nazorat qiluvchi , foydalanuvchidan himoyalangan funksiyalar majmuasidan iborat.

Bu resurslar orasida sistemali rejalashtirish sistema kutubhonalari fayllarni boshqarish va kiritish – chiqarishga hizmat qiluvchi servis dasturlar mavjud.

2. Kiritish- chiqarish qurilmalari va dasturlari. Kiritish va chiqarish qurilmalari kompyuter konfiguratsiyasiga bog'liq va ularning soni bir necha bo'lishi mumkin. Real vaqtida sistemaga bog'langan qurilma va qurilmalar soni kompyuterning portlari sonidan ko'p bo'la olmaydi. Har bir qurilmani ishga tushirish va u bilan berilganlarni almashish operasion sistema yordamida amalga oshiriladi, dastur odatda bevosita qurilma va uning fizik harakteristikasini hisobga olgan holda yaratilgan bo'ladi, ayrim hollarda biron bir kiritish-chiqarish qurilmasi o'rniga boshqasini ishlatish, xotira bilan berilganlarni o'zaro

almashish hatolikka yoki umuman berilganlarni almashmaslikka olib keladi, bunday holda bog'lanish amalga oshmagani sabablaridan biri- bu qurilma uchun qo'llaniladigan dasturni mos kelmasligi bo'lishi mumkin, bunday holda qurilmaga mos dasturni yuklab so'ng undan foydalanish tavsiya qilinadi.

3.Kommunikasiya –bu ikki sistemali daqsturni, buyruqni, dastur va buyruqni yoki buyruq va dasturni birlashtirish, Bunday ko'rinishdagi birlashtirish bir dastur yoki buyruqning natijasini boshqa dastur yoki buyruqqa kiritish imkoniyatini beradi. Yo'naltirilgan kiritish –chiqarish bilan kommunikasiya farqi:yo'naltirilgan kiritish-chiqarish bu berilganlarni o'qish yoki ularni "periferiya" qurilmasiga uzatishdir.Kommunikasiya esa-bu sistema dasturlari va buruqlari orasidagi o'zaro berilganlarni almashishdir, ya'ni berilganlarni uzatish operasion sistema ichida amalga oshiriladi.

4.Operasion sistemaning qo'shimcha funksiyalari- Berilganlarga ishlov berish, berilganlar kompyuter xotirasida turli ko'rinishda saqlanadi, bular avvaldan kelishilgan holda bo'ladi.

Masalan: dastur saqlanish prinsipi bilan berilganlarni saqlashi turli bo'ladi, biron bir matn muharriri yordamida hosil qilingan ma'lumot boshqa muharrir yordamida hosil qilingan ma'lumotning ichki ko'rinishidan farq qiladi, har bir ma'lumotning ichki tuzilishi avvaldan tanlab olingan kodlash usuli yordamida xosil qilinadi, kodlar turlari va kodlash usullari turlicha ularni qanday ko'rinishda tanlab olish va ishlatish bevosita sistema ijodkorlariga bog'liq.

5.Kompyuterning ishonchliligini ta'minlash- kompyuterning qurilmalari normal ishlashi uchun ma'lum shart- saroitlar bajarilishi talab qilinadi, bu elektr manbayi, parametrlaru, tashqi muhit, temperaturasi va boshqalaridir, bu shart-sharoitlardan chetga chiqish kompyuter apparaturasida uzilishga yoki noto'g'ri ishlashga olib keladi. Operasion sistemada apparatura noto'g'ri ishlashi natijasida paydo bo'ladigan hatolikni inkor qiluvchi dastur mavjud bo'lib, zaruriyat tug'ilganda sistema tashabbusi bilan bu dastur ishlaydi va hosil bo'lgan hatolikka ishlov beradi, bundan tashqari dasturiy resurslar noto'g'ri ishlashi natijasida hatolik vujudga keladi. Bu vaziyatda ishlatilayotgan dasturning to'g'ri yoki noto'g'ri ishlayotganligini aniqlash uchun turli uslublardan foydalaniladi, bunday uslublardan biri, qo'yidagicha, operasion sistema tarkibiga mahsus dastur kiritiladi va bu dastur berilganlarga ishlov berishdan avval va ishlov bergandan so'ng tekshiriladi, shu dastur kodlari yig'indisi tekshiriladi, agarda hosil bo'lgan kod avvaldan shu dastur uchun aniqlangan kodga teng bo'lsa, u holda dastur to'g'ri ishlaydi yoki ishlagan talqin qilinadi.

6.Proessor vaqti-topshiriq bajarilish uchun ketgan umumiy vaqt proessor va kutish vaqti majmuasidan iborat bo'ladi, proessor vaqti bevosita foydalanuvchi dasturiga ishlov beradigan vaqt bilan aniqlanadi.

7.Xotirani boshqarish- operasion sistema tarkibidagi maxsus dasturlar yordamida bajariladi, xotira ishchi dastur bilan yuklanganda sistema uchun qabul qilingan hajmdagi xotira ajratiladi yoki bo'lmasa topshiriqlar tilida ko'rsatilgandek joy ajratiladi shuni takidlash keraki operasion sistema asosini tashkil qiluvchi dasturlar xotirada doim saqlanib turadi, uning uchun xotirada maxsus joy ajratilgan va boshqa dasturlar yordamida bu joyga kirish operasion sistema himoya dasturi yordamida himoyalangan.

8.Dasturiy resurslar-bevosita operasion sistema ishini ta'minlaydigan va foydalanuvchi ishlatadigan dasturlar majmuasidan iborat bo'ladi. Yordamchi dasturlar hajmi foydalanuvchi ehtiyojiga qarab aniqlanadi, bu holda yordamchi dasturlar qancha ko'p bo'lsa shuncha yaxshi deyish noto'g'ri, chunki dastur qancha ko'p bo'lsa ularni saqlash qidirish va ishga tushirish shuncha murakkab bo'ladi.

9.Modem qurilmalari- Modem modulyasiya, demodulyasiya so'zlaridan oligan bo'lib, uzluksiz signallarni raqamli va raqamli ma'lumotlarni uzluksiz signalga almashtirib beradigan qurilmadir.Uning asosiy vazifasi kompyuterlar aro aloqani o'rnatishdir, u uzining

kommunikasion dasturlariga ega bo'lib, bu dasturlar yordamida ma'lumotlarni uzoq masofalarga uzatishi va qabul qilishi mumkin. Modemlar ichki va tashqi bo'lishi mumkin.

Kompyuterlarda telefon tarmog'i orqali axborot almashish maqsadida ishlatilayotganda, telefon tarmog'idan olingan signalni qabul qila oluvchi va uni raqamli axborotga aylantiruvchi qurilma lozim bo'ladi. Qurilmaning kirishida axborot modulayasisya qilinadi, chiqishda esa aksincha, demoduliyasiya holati vujudga keladi bu esa kompyuterlarning o'zaro aloqasini vujudga keltiradi.

Kompyuterning programmaviy ta'minoti

Birinchi EHM lar hisoblash markazlariga hech qanday programma ta'minotisiz, faqat texnik qurilmalar ko'rinishida keltirilardi. Shuning uchun ham EHM bajarishi lozim bo'lgan vazifalarning hammasi foydalanuvchi tomonidan tuzilgan programmada batafsil tasvirlangan bo'lishi talab qilinardi. Bu esa, foydalanuvchidan operator va programma tuzuvchi sifatida maxsus texnik bilimlarni va mashinaga bog'liq tillarda programma tuzishni bilishni talab etardi. Shuning uchun ham boshlang'ich paytlarda EHM dan ko'proq matematik-programmistlarga unumli foydalana olishardi.

Ma'lum vaqtdan keyin, EHM ni boshqarish bo'yicha maxsus texnik ishlarni, maxsus programmalar yozish orqali, kompyuterning o'ziga yuklash maqsadga muvofiqligi anglab yetildi. Bu sermashaqqat texnik vazifalarni mashinaga yuklash va programma tuzishni osonlashtirish maqsadida «Operatsion sistemalar» va «Yuqori bosqich algoritmik tillari» ni yaratish zaruriyati yuzaga keldi. Algoritmik tillarning paydo bo'lishi programmani mashina tiliga tarjima qiladigan translator va interpretatorlarning yaratilishini taqozo etdi.

O'z navbatida programmalar yozish ham ikki yo'nalish bo'yicha rivojlana boshladi. Konkret ilmiy-texnikaviy amaliy masalalarni sonli yechishga mo'ljallangan programmalar tuzish-tadbiqiy programmalash deb nomlandi. Programma yozishni va uni sozlashni (отладка) avtomatlashtirish va EHM resurslaridan unumli foydalanishni hamda inson va EHM o'rtasida qulay muloqotni (interfeys) ta'minlashga mo'ljallangan programmalar ta'minotini yaratishga-tizimli programmalash deb ataldi. Zikr etilgan vazifalarni bajaruvchi programmalar tuzish bilan shug'ullanuvchilarni mos ravishda, tadbiqiy va tizimli programma tuzuvchilar deb atashdi. Hozirgi kunda kompyuterlar sotuvga bazaviy programma ta'minoti bilan birgalikda chiqariladi.

Operatsion sistema, tizimli va xizmatchi programmalar to'plamidan iborat bo'lib, hamma programmalarining, EHM va tarmoqlar texnik qurilmalarining ishlashini hamda ularning resurs va imkoniyatlaridan unumli foydalanishni, shu bilan birgalikda

foydalanuvchi, programma ta'minoti va EHM o'rtasida o'zaro qulay muloqotni ta'minlash uchun xizmat qiladi.

Yuqoridagi fikrlarga tayangan holda operatsion sistemalarning asosiy vazifasi uch turdagi qulay muloqotlarni - interfeyslarni yaratishdan iborat deb qarash mumkin:

- foydalanuvchi bilan kompyuterning programma va apparat vositalari orasidagi interfeys (Foydalanuvchi uchun interfeys);
- kompyuterning apparat va programma ta'minoti orasidagi interfeys;
- programma ta'minotining turli ko'rinishlari orasidagi interfeys.

Kompyuter elektr tarmog'iga ulanganda, operatsion sistema (OS) tezkor xotiraga yuklanadi va EHM qurilmalarining ishga yaroqliligi va undan keyin bajariladigan barcha vazifalar OS tomonidan boshqariladi.

Disk va disketalarni formatlash, fayllarni saralash, arxivlash, matnlarni chop etish, tasodifan o'chirilgan fayllarni tiklash, viruslarni topish va yo'qotish kabi ko'p takrorlanuvchi amallarni bajaruvchi programmalar - *utilitlar* deb ataladi.

Kompyuterning printer, disk yurituvchi, displey, klaviatura kabi qurilmalarining ishlashini ta'minlovchi tizimli programmalar to'plamiga *drayverlar* deb ataladi.

EHM da tadbqiqiy va tizimli programmalarini bajarish paytida fayllar ustida yozish, o'qish, saqlash, izlash, o'chirish kabi ko'p martalab takrorlanuvchi ishlarni bajarish uchun qulay shart-sharoitlar yaratib beradi.

Hozirgi kunda juda ko'p operatsion sistemalar mavjud bo'lib, ularni quyidagi sinflarga ajratish mumkin:

- bitta yoki ko'p foydalanuvchilarning ishlashini ta'minlashga mo'ljallangan operatsion sistemalar;
- bir paytning o'zida bitta yoki bir nechta masalani yechish imkoniyatiga qarab, bir va ko'p masalalik OS lar;
- bir protsessorli va ko'p protsessorli OS lar;
- 8, 16, 32, 64 razryadli operatsion sistemalar;
- nografik va grafik OS lar;
- mahalliy va tarmoq operatsion sistemalari.

Eng ko'p tarqalgan operatsion sistemalar sifatida SR/M, MS DOS, OS/2, Windows, UNIX va MacOS larni sanab o'tish mumkin. SR/M va MS DOS lar bitta foydalanuvchi va bir masalali operatsion sistemalaridir. OS/2 va Windows esa ko'p masalali bir foydalanuvchiga mo'ljallangan. UNIX operatsion sistemasi ko'p masalali va ko'p foydalanuvchilarga mo'ljallangan.

Nazorat savollari:

1. Kompyuter ishlaши учун =андай шарт зарур?

2. Дастурий таъминот нечага былинади?

3. Тизим дастурлари =андай дастурлар?

4. +андай операцион системаларни биласиз?

5. Амалий дастурлар ша=ида маълумот беринг.

6. Амалий дастурлар пакети нима?

MAVZU 4-5: UTILITLAR VA TIZIMLI DASTURLAR.

R E J A:

1. Sistema dasturlari.

2. Utilitlar.

3. Virus va sistemali antivirus dasturlar.

Tayanch so`z va iboralar: sistema dasturlari, utilitlar, operatsion qobiqlar, Drayverlar, Formatlash vositalari, Virus dasturlar.

Sistema dasturlari qatoriga operatsion sistemalar, operatsion qobiqlar, fayl-menedjerlari va drayverlarni kiritgan edik. Ularning vazifalari quyidagicha:

Operatsion sistemalar kompyuter ishga tushgan zaxoti faolyat ko`rsatuvchi dasturiy vositalardir. OC lar asosan quyidagi vazifalarini bajaradi:

- Kompyuter qurilmalarining ishga tayyorligini tekshirib chiqadi va ularni ko`ngildagidek ishlashini nazorat qilib boradi. Agar qurilmalarning ishida nuqson sezsa, bu xaqda foydalanuvchiga xabar beradi.
- Kompyuterning ishi jarayonida uning xotira qurilmalari (tezkor xotira, qattiq disk, disketa) dagi bo`sh joylardan unumli foydalanishni tashkil qiladi. Ularga ma`lumotlarning yozilishi va o`qilishini nazorat qiladi va boshqaradi.

- Foydalanuvchi interfeysi orqali kompyuter va foydalanuvchi o'rtasida muloqot o'rnatadi. Foydalanuvchining buyruqlarini kompyuterga kiritadi va ularning bajarilishini nazorat qiladi.
- Foydalanuvchining talabi bo'yicha barcha boshqa dasturlarni ishga tushiradi va ularning bajarilishini nazorat qilib boradi.

Foydalanuvchi interfeysi – OC ning foydalanuvchi bilan muloqotini qulay va samarali tarzda olib borishga xizmat qiluvchi maxsus usullari va vositalaridir. Ular buyriqli, ob'yektga yo'naltirilgan va dasturiy interfeyslarga bo'linadi.

Ilova dastrlar-aynan bir turdagi OC boshqaruvida ishlashga mo'ljallab tuzilgan dasturlardir.(masalan MS DOS ilova dasturlari, WINDOWS ilova dasturlari va hokoza). Keyingi yillarda ishlab chiqarilayotgan OC larning ko'pchiligi boshqa OC larning ilova dasturlari bilan ham ishlash imkoniga ega.

. Hozirgi payitda eng ko'p tarqalgan OC lar MS DOS, OS/2, WIDOWS, UNIXlar hisoblanadi.

MS DOS OS Microsoft kompaniyasi tomonidan ishlab chiqarilgan eng dastlabki va eng ishonchli OC dir. Boshqa OCLar boshqaruvida ishga tushirish qiyinchilik paydo qilayotgan kompyutir yoki dastur MSDOS orqali osonlik bilan ishga tushadi. Uning kamchiligi foydalanuvchi bilan muloqat qilish usuli noqulayligidir. Bu kamchilik fayil-menijerlari ("qobiq" dasturlari) yordamida bartaraf qilinadi. Ularga NC va Volcov Commonder kabi vositalar misol bo'la oladi.

Dastlabki MSDOS OCi 1981yilda yaratilgan bo'lib IBM XT va IBM AT kompyutirlariga mo'ljallangan bo'lib u MSDOS 1.0 versiyasi edi.

Zamonaviy Pentium kompyutorlariga mo'ljallanga versiyasi MS DOS6.20esa 19__ yilda ishlab chiqarildi. OC ichida 1987 yilda IBM kompaniyasi tomonidan PS 2 kompyutlari maxsus ishcilar bo'lib ko'p dasturlar bilan bir vaqt ichida ishonchli ishlashi mumkin WINDOWS XP- WINDOWS OCLarining eng yahshi hususiyatlarini o'zida jamlagan tizimdir. Grafik va multivideo malumotlari bilan ishlash suratini oshiruvchi eng zamonaviy vositalarga ega 2002 yilda ishlab chiqarilgan.

UNIX ko'p vazifali va bir vaqtning o'zida ko'plab fayllardan fayllarga xizmat ko'rsata oladigan tizimdir qulaylik tamoni shundaki bu tizimda shaxsiy kampyuter tipi super kampyuterlargacha foydalanish mumkin. Malumotlar bazasi bilan ishlash lokal tarmoqlardan tortib to glabal tarmoqlargacha oddiy modem yordamida ulana olish imkonini baradi. Ko'plab ilova dasturlari mavjud.

Operatsion qobiqlar.

Ular grafik interfeys va multi dasturlashtirish vositalari yordamida foydalanuvchini MS DOS bilan muloqot qilishga imkon beruvchi vositalardir bularga MS WINDOWS 3.1 dasturini misol keltirish mumkin.

Fayl menedjerlar.

Ular foydalanuvchining MS DOS boshqaruvida kamyuter bilan qulay va ko'rgazmali muloqot olib borishini taminlovchi dasturdir. Ko'pincha "qobiq " dasturlar deb ham ataladi ularga NC,VC,WINDOWS commander kabi dasturlar misol bo'ladi.

Drayverlar.

Ular kamyuterbilan boshqa qurilmalar o'rtasida muloqot o'rnatishga xizmat qiluvchi dasturlar dir ular xarbir qurilma (klavyatura,printer, sichqoncha va xokazo) lar uchun aloxida ishlab chiqariladi, hu qurilmalarini kamyuter bilan xamkorlikda ishlashini ta'minlaydi va nazorat qilib boradi masalan Mouse.com-kamyuteri bilan sichqoncha o'rtasidagi muloqotni tayyorlaydi.

Utilitlar.

Utilitlarga arxivlash, formatlash vositalari anti viruslar deagnostika vositalari optemallash vositalari komunikatsya vositalari xtirani boshqaruvchi vositari misol keltirish mumkin. Arxivlash vositalari yoki arxivatorlar-maxsus usullar yordamida fayl xajmini qisib kichraytirishga ularning arxivlarini xosil qilishga xizmat qiluvchi vositalardir.

Arxivlangan fayllar birqancha qulayliklarga ega masalan:

Ulariga maxsus kamyuter viruslarning 90% dan ko'prog'I ta'sir etolmaydi. Matnli fayllar xajmi 50-60% qisqaradi. Ularni boshqa foydalanuvchilardan ximoya qilishning samarali usullari .Zamonaviy arxivatorlarga PKZIP, ARJ, PAK, RAR, WINRAR, WINZIP, kabi dasturlarni misol keltirish mumkin.

Formatlash vositalari.

Ma'lumot yozishdan oldin disklar satrlari sektorlarga sektorlar esa klasterlarga klasterlar esa ma'lumotlar yoziladigan yo'lakchalarga bo'linadi. Shundan keyin disklar ishchi xotiraga keladi bu esa ma'lumotlarning disda optimal

Xamda zarur payti kerakli ma'lumotlarni tezda izlab topishga imkon beradi disklar satxlarini shunday sektor va klasterlarga bo'lish diskni formatlash deyiladi va bu jarayonni maxsus

dasturlar yordamida bajariladi. Formatlangan 3.5 dyumlik disketada 18 sektor va 80 yo'lak bo'ladi bitta sektorning yo'klari sig'imi 51.2 baytga teng bo'lib ular klasterlar deyiladi.

Virus dasturlar va ularning turlari.

Kompyuter viruslari dep kompyuterning xotirasidagi ma'lumotlarni ishdan chiqarish maqsadida maxsus tuzilgan dasturiy vositalarga aytiladi dastlabki paytda ular disketalar orqali tarqalar edi. Keyingi yillarda kompyuter tarmoqlari orqali xam tarqalmoqda. Ularning buzg'unchilik faoliyati najijasida minglab kamyuterlar va kompyuter tarmoqlari ishdan chiqmoqda va ularning xotirasidagi ma'lumotlar yo'qolmoqda. Natijada ularning keltrayotgan zarari millionlab dollarlar bilan o'lchanmoqda.

- Virus dasturlar ta'sirida.
 - Fayllarning xajmi sezilarli ortishi.
 - Kompyuterning ish tezligi sekinlashishi.
 - Ekranda natija o'rnida turli belgilar paydo bo'lishi.
 - Chaqirilgan fayllar umuman ishga tushmasligi.
 - Kompyuter xotirasida begona fayllarning paydo bo'lishi.

Virus dasturlarni guruxlarga bo'lish mumkin.

- 1.Oddiy fayilli dasturlar fayllar bilan birga kompyuter hotirasiga kelib tushuvchi va shu fayil ishga tushirilgandagina o'z faoliyatini boshqaruvchi viruslar.
- 2.Rezident fayilli dasturlar chaqirilgan fayil bilan birga kompyuterning tezkor hotirasiga kelib tushuvchi va kompyuterning keyingi ishi davomida tezkor hotirada qolib tezkor hotiraga chaqirilgan barcha fayllarni zararlovchi viruslar.
- 3.Yuklovchi sektor viruslarri disklar yoki disketalarning yuklovchi sektorlarini ishdan chiqarishga mo'ljallangan ya'ni shu sektorda joylashgan sistema dasturlarini zararlovchi viruslar.
- 4.“Chuvalchang” viruslari boshqa dasturlarini zararlamaydi biroq o'z o'zidan nusxa olib ko'payuvchi viruslar ularning ishi natijasida kompyuter xotirasi virus dastur nusxalari bilan to'lib qoladi kompyuterning ishlash samaradorligini keskin pasaytiradi.

Antiviruslar.

Virus dasturlarini izlab topuvchi va ularni zararsizlantiruvchi dasturiy vositalardir. Ular o'zlarining ish usullariga va imkoniyatlariga ko'ra guruxlarga bo'linadi:

- detektor dasturlar va doktor dasturlar. Ular virus bilan zararlangan fayllarni izlab topadi xamda viruslarga davolashga xarakat qiladi masalan. Dialog-MGU firmasi yaratgan AVP, Dr Web dasturlari shular jumlasidan.

- Revizor dasturlari va doktor revizorlar. Ular fayllarning dastlabki xolatlar to'g'risidagi ma'lumotlarni (xajmi diskdagi o'rni va xokazo) eslab qoladi so'ngra tekshirish vaqtida bu ma'lumotlarni dastlabkisi bilan solishtiradi agar farq bo'lsa ular zararlangan deb xisoblaydi va virslarni aniqlash va faylni davolash faoliyatini boshlaydi.

Ularga misol qilib "diyalog MGU" firmasining ADinf dasturini keltirish mumkin.

Filtir dastur yoki rizident-dastur. Ular kompyutirning tezkor hotirasiga aftamatik ravishda yuklanib, kompyut ishi davomida shu joyda qoladi, hamdahotiraga chaqirilayotgan har bir dasturni viruslardan zararlanmaganligini uzluksiz tekshirib boradi, Flushot plus dasturishunday dasturlardan.

Dastur vakiinalar yoki imunizatorlar. Ular diskni va undagi fayllarni viruslarning tasiriga berilmovchi qilib takomillashtiruvchi hamda fan rejimida faoliyat olib boruvchi vositalardir. Zamonaviy antiviruslar dasturini ishlab chiqarishda aynan shunday vositalarga katta ahamiyat berilmoqda. Kiyingi yillarda yuqoridagi antivirus vositalarinning barcha hususiyatlarini o'zida mujasamlashtirgan universal vositalar ham ishlab chiqarilmoqda. Bular ichida eng ko'p tarqalgani Kasperiskiy laboratoriyasida yaratilgan Kasperiskiy dasturidir.

Diyagnostika vositalari. Kompyuter qurilmalari va magnit disklarning ishlash qobilyatlarini va holatini tekshiruvchi hamda ulardagi nuqsonli joylarni aniqlab, iloji boricha tuzatadigan dasturlar. Vazifasiga ko'ra ular ikkiga bo'linadi

- kompyuter qurilmalarining ish faoliyatini tekshiruvchi dasturlar. Masalan: Contral R
- magnit disklarning ish holatini tekshiruvchi dasturlar. Masalan: NDD

Optimallashtirish vositalari.

Qattiq va yumshoq disklarga fayllar klasterlarga yoziladi. Bunda quydagi hollar bo'lishi mumkin:

-fayllarning hajmi klaster sig'imida kichik bo'lishi mumkin, bunda klasterlarda bo'sh joylar hosil bo'ladi.

-fayilning hajmi klaster sig'imidan katta bo'lish mumkin, bunda shu fayl bir nechta klasterlarga bo'lib yoziladi.

Diskka bir marta ma'lumotlar yozilayotganda bitta klasterga sig'magan fayllarni bo'lib yozish tartib bilan amalag oshiriladi, ya'ni faylning ortib qolgan qismi qo'shni klasterga yoziladi. Diskdan ko'p marta foydalanilganda ko'p marta ma'lumotlar yozib yoki o'chirilganda disk ma'lumotlar yozmagan bo'sh joylar hosil bo'lishi hamda fayllarni klasterlarga bo'lib yozish tartibi buzilishi mumkin, natijada bitta faylning bo'laklari butun disk bo'ylab tartibsiz ravishda sochilib ketishi mumkin. Bunday holatda frakmentasiya diyeiladi. Buning natijasida

diskning umumiy sig'imi bo'sh joylar hisobiga kamayadi va fayillarni izlab topish murakablashadi, kompyuterning ish surati sezilarli darajada kamayadi.

Fragmentasiyalarni yo'qotish, ya'ni bosh joylarni malumotlar bilan to'ldirish, fayillarning bo'laklarini klasterlarga tartib bilan joylashtirishga defragmentasiya yoki oftemaleshtirish deyiladi. Buning uchun SpeeDisk, Fast Trax kabi dasturlardan foydalaniladi.

Nazorat uchun savollar:

1. Operatsion sistema nima?
2. Operatsion sistemalarning vazifalari nimalardan iborat?
3. Foydalanuvchi interfeysi nima?
4. Ilova dastrlar nima misollar keltiring
5. OC larning turlari?
6. Drayverlarga misollar keltiring.

II-bob. Algoritmik til va Paskal dasturlash tili.

Mavzu – 2.1:

Algoritm tushunchasi. Algoritmik til elementlari.

Reja:

1. Algoritmik til elementlari.
2. Murakkab buyruqlar.
3. Kattaliklar

*Tayanch so'z va iboralar: **algoritm, algoritmik til, murakkab buyruqLar, tarmoqlanuvchi algoritm, takrorlanuvchi alogoritm.***

Algoritmik til **algoritm**larni bir xil va aniq yozish va hamda ularni bajarish uchun ishlatiladigan belgilar va qoidalar sistemasidir.

Bir tomondan algoritmik til oddiy tilga yaqin bo'lsa, ikkinchi tomondan u matematik belgilar, son kattaliklar funktsiya belgilari, amallar , qavslar va boshqalarni o'z ichiga oladi. Har qanday til kabi algoritmik til ham o'z lug'atiga ega. Bu lug'atning asosini ibror algoritm

ijrochisining buyruqlari tizimiga kirgan buyruqlarni ifodalashda qo'llaniladigan so'zlar tashkil qiladi va ular soda buyruqlar deb ataladi. Sodda buyruqlar to'liq yoki qisqa shakldagi buyruq gapga o'xshash bo'lib, zarur bo'lganda formulalar va boshqa belgilarni o'z ichiga oladi.

Kattaliklarning nomini belgilashda ularning ma'nosini va vazifasini tushuntiradigan harflar, so'zlardan foydalanish mumkin. Kattaliklar natural, butun, haqiqiy, belgisi bo'lishi mumkin.

Algoritmik tilda yozilgan algoritmning umumiy ko'rinishi quyidagicha bo'ladi:

Alg algoritm nomi.

Boshl

qator (algoritm buyruqlari)

tam

Algoritm algoritmik tilda yozilgan saravhadan boshlanadi. Sarlavhada algoritm nomi, keyin qavs ichida vergullar bilan ajratilgan algoritmda qatnashadigan kattaliklarning turlari, argumentlar va natija tavsiflanadi. Algoritm sarlavhasi quyidagicha bo'ladi:

alg nom (tavsiflanadigan kattaliklar ro'yxati)

arg argumentlar ro'yxati

natija natijalar ro'yxati

Misol: Ikki butun musbat M va N sonlarning EKUBini topish algoritmining sarlavhasini yozing.

alg EKUB (nat M, N , EKUB)

arg M, N

natija EKUB

Bu erda alg, arg, nat, ... yordamchi so'zlar. Algoritmtdagi biror tugallangan amalni bajarish haqidagi ko'rsatma buyruq deyiladi. Birin ketin bajariluvchi oddiy buyruqlar ketma-ketligi qator (seriya) deyiladi.

Algoritmlar chiziqli, tarmoqlanuvchi va takrorlanuvchi turlarga bo'linadi.

Algoritmlarni blok-sxemalar ko'rinishida tasvirlash ham mumkin.

Murakkab buyruqlar

Algoritmda oddiy buyruqlarning bitta seriyasidan iborat algoritmlar chiziqli bo'ladi. Tarmoqlangan va siklik algoritmlarni algoritmik tilda yozish uchun qo'shma gaplarga o'xshash murakkab buyruqlardan foydalaniladi.

Tarmoqlanish buyrug'i quyidagicha yoziladi:

agar shart

bo'lsa 1-qator

aks holda 2-qator

hal bo'ldi

Shartga bogʻliq holda tarmoqlanish buyrugʻiga kiruvchi qatrlardan faqat bittasi bajariladi.
Tarmoqlanish buyrugʻi qisqa shaklda ham ishlatiladi:

agar shart

boʻlsa qator

hal boʻldi

Toʻliq va qisqa shakllar quyidagi blok-sxemalar koʻrinishida ifodanadi:

Misol: Eritma muhitini Aniqlash algoritmini tuzing. Bunda eritmaga lakmus qogʻozi tushiriladi. Uning rangiga qarab eritma ishqorli, kislotali yoki neytralligi aniqlanadi.

Takrorlanish buyrugʻi

Takrorlanish buyrugʻi kompyuterda masala echishda muhim ahamiyatga ega, undan foydalanib juda uzun amallar ketma-ketligini bajarish haqida kompyuterga koʻrsatma berish mumkin. Takrorlanish buyrugʻi umumiy holda quyidagicha yoziladi:

Toki shart

sB

kator

sO

Uning blok-sxemasi quyidagicha boʻladi:

Misol: Bashe oʻyini. 15 ta buyum bor. Raqiblar navbat Bilan yurishadi, har yurishda oʻyinchi bita, ikkita yoki uchta buyum olishi mumkin. Oxirgi buyumni olgan oʻyinchi yutqazadi. Birinchi oʻyinchingin yutish algoritmini yozilsin.

Alg Bashe oʻyini

Boshl ikkita buyum olinsin;

Toki buyumlar soni toʻrttadan ortiq

SB

Raqibning yurishiga imkon berilsin;

Raqib olgan buyumlar soni K esda saqlansin;

4-K ta buyum olinsin;

SO

Raqibning yurishiga imkon berilsin.

Tam

Kattaliklar.

Kattaliklar o'zgaruvchi va o'zgarmas bo'ladi. Algoritm bajarilishi davomida qiymati o'zgarmaydigan kattaliklar o'zgarmas kattaliklar deyiladi. Algoritmning bajarilishi davomida qiymati o'zgaradigan kattalik o'zgaruvchi kattalik deyiladi. Har bir kattalik o'z nomiga ega bo'ladi. Nom tanlashda kattalikning ma'nosi va vazifasini tushuntiruvchi istalgan so'zlarni ham ishlatish mumkin. Masalan, ko'paytma, qadamlar soni va xakozo.

Qiymatlari matnlardan iborat kattaliklar matnli kattaliklar deyiladi. Uzgaruvchilarning tiplari qisqacha nat, but, haq, harf va hokazo so'zlar bilan belgilanadi.

Oraliq kattaliklar.

Misol: kvadrat tenglamani echish algoritmini yozing.

alg KVT (haq a,b,c, x1,x2, harf u)

arg a,b,c

natija x1,x2, u

boshl haq D

D:=b²-4ac

Agar D<0

Bo'lsa u:="echim yo'q"

Aks xolda u:="echim bor"

hal buldi

tam

Algoritmni tahlil qilaylik. Boshl xizmatchi so'zidan keyin haq D yozuvi paydo bo'ldi. D kattalik algoritmnining argumenti ham natijasi ham emas. Uni oraliq o'zgaruvchi deyiladi. D:=b²-4ac yozuvi D o'zgaruvchiga b²-4ac ning qiymati berilsin de bo'qiladi. "[:=" qiymat berish buyrug'i deyiladi.

Misol: Ikkita butun M va N sonlarning EKUBini topish algoritmini tuzing.

Alg EKUB (nat M,N, Ekub)

arg M,N

natija EKUB

Boshl nat x,y

X:=M, y:=N

Toki

SB

Agar x>y

Bo'lsa x:=x-y

Aks xolda u:=y-x

xal buldi

SO

tam

MURAKKAB SHARTLAR

ifodani hisoblash algoritmi tuzilsin.

Alg misol (haq x, y harf R)

arg x

natija y, P

Boshl nat x, y

Agar $x=0$ yoki $x=1$

Bo`lsa $P := "y \text{ ning qiymati Aniqlanmagan}"$

Aks xolda $P := "y \text{ ning qiymati Aniqlangan}"$

xal buldi

tam

Bu algoritmda $x=0$ yoki $x=1$ shart murakkab shart deyiladi. Murakkab shartlarni yozish uchun va, yoki, emas xizmatchi so`zlaridan foydalaniladi.  $a$  va  $b$  shartlar desak,  $a$  ham  $b$  ham o`rinli bo`lsa  $a$  va  $b$  shart o`rinli.

Nazorat uchun savollar:

1. Algoritmik til nima?
2. Algoritmik tilda yozilgan algoritmning umumiy ko`rinishi qanday?
3. Algoritm sarlavhasi qanday yoziladi, misollar keltiring.
4. Tarmoqlanish buyrug`ining umumiy shakli va blok-sxemasini tushuntiring.
5. Takrorlanish buyrug`ining umumiy shakli va blok-sxemasini tushuntiring.
6. Kattaliklar, oraliq kattaliklarga misollar keltiring.
7. Murakkab shartlarning ishlatilishiga misollar keltiring.
8. Algoritm ta'rifini ayting, misollar keltiring

MAVZU 2.2:

DASTURLASH TILLARI VA VA ULARNING SINFLARI. PASKAL PROGRAMMALASH TILI

REJA:

1. Programmalash tillari haqida
2. Paskal tiliga kirish
3. Programmalash tillari va texnologiyalari

Endi bizning oldimizda algoritmlarni EHM tushinadigan tilda yozish masalasi turadi, buning uchun maxsus algoritmik tillar mavjud. EHM paydo bo'lganidan beri juda ko'plab algoritmik tillar yaratilgan. Ularni shartli ravishda uch tipga ajratish mumkin:

- Quyi darajadagi programmalash tillari (mashina tillari)
- O'rta darajadagi programmalash tillari
- Yuqori bosqichli programmalash tillari

Quyi darajadagi programmalash tillarida buyruqlar va amallar ma'lum kodlar (raqamlar) bilan ifodalangan bo'lib, ular EHM qurilmalari adreslari bilan bevosita ishlashga mo'ljallangan va mashina tili deb ham yuritiladi.

O'rta darajadagi programmalash tillarida amallar va buyruqlarni (komandalar)ni ifodalash uchun odam tushunishi uchun qulay bo'lgan har xil qisqartma so'zlardan foydalaniladi. Bunda ham programma tuzuvchi EHM qurilmalari adreslari bilan ishlashi lozim. Bu qisqartmalar-mnemokodlar deyiladi, bu turdagi programmalash tillari assemblerlar deb ataladi.

Yuqori darajadagi programmalash tillarida ko'rsatmalar inson tiliga yaqin bo'lgan so'zlar va birikmalardan iborat bo'lib, programma tuzish uchun juda qulay va biror maxsus amallardan tashqari hollarda adreslar va qurilmalar bilan bevosita bog'liq ko'rsatmalarni bilish zarur emas. Yuqori bosqich programmalash tillariga *misol* sifatida FORTRAN, Algol, RL-1, BASIC, PASCAL, C, C++ tillarini keltirish mumkin.

Informatsion texnologiyalarining jadal suratlar bilan rivojlanishi, Internetning paydo bo'lishi, yangidan yangi va ma'lum maqsadlarga mo'ljallangan programmalash tillari va muhitlari yaratildi va yaratilmoqda. Ularga *misol* sifatida dBase, KARAT, LISP, FoxPro, Simula, Delphi, Visual Basic, Visual S++, HTML, Java, Java Script kabi tillar va programmalash muhitlarini keltirish mumkin. Hozirgi kunga kelib obyektga yo'naltirilgan va vizual programmalash texnologiyalari keng tarqalmoqda.

Har bir algoritmik tilning o'z alfaviti maxsus belgisi va birikmalari, operatorlari, konstruksiyalari bo'ladi va ular ma'lum bir sintaktik va semantik qonun qoidalarga bo'yinadi.

Sintaksis—bu alfavitdan foydalanilgan holda biror algoritmik tilda ishlatilishi mumkin bo'lgan konstruksiyalarni aniqlovchi qoidalar sistemasidir. Bu konstruksiyalar orqali algoritmlar ifodalanadi. Shunday qilib, sintaksis har bir simvollar ketma-ketligining berilgan tildan foydalanish mumkinligini yoki mumkin emasligini aniqlab beradi.

Semantika—biror algoritmik til konstruksiyalarini ma'nosi jihatidan tekshiruvchi qoidalar sistemasidir. Biror algoritmik tilning sintaktik qoidalarini ifodalash uchun maxsus yordamchi tilga—metatil deb ataladi. Eng keng tarqalgan metatillardan biri Bekus–Naurning metalingvistik formulalari va sintaktik diagrammalaridir. Bir algoritmik tilning qonun qoidalarini aniq va bir qiymatli aniqlash uchun maxsus tushincha va belgilar zarur bo'ladi. Tilning har bir tushinchasi

uchun yagona metoformula mavjud bo'lishi kerak va uning gap qismida kiritilayotgan tushincha, ya'ni metao'zgaruvchi ko'rsatiladi. O'ng tomonda esa, metoo'zgaruvchining qabul qilishi mumkin bo'lgan qiymatlar to'plami keltiriladi. Odatda metoo'zgaruvchilar maxsus $\langle \rangle$ kavslar ichida yoziladi. Masalan: $\langle \text{son} \rangle$, $\langle \text{arifmetik ifoda} \rangle$. Metoformulaning chap va o'ng qismlari maxsus $::=$ metosimvol bilan ajratiladi va u "ta'rif bo'yicha" degan ma'noni anglatadi. Masalan, quyidagi metoformula

$$\langle \text{o'zgaruvchi} \rangle ::= A \setminus V$$

o'zgaruvchi ta'rif bo'yicha A yoki V harfidir degan ma'noni ifodalaydi.

$$\langle \text{ifoda} \rangle ::= \langle \text{o'zgaruvchi} \rangle \mid \langle \text{o'zgaruvchi} \rangle + \langle \text{o'zgaruvchi} \rangle \mid \\ \mid \langle \text{o'zgaruvchi} \rangle - \langle \text{o'zgaruvchi} \rangle$$

metoformula esa, yuqoridagi $\langle \text{o'zgaruvchi} \rangle$ metoformulasiga bog'liq holda $\langle \text{ifoda} \rangle$ sifatida quyidagi 10 ta ifodadan ixtiyoriy bittasi bo'lishi mumkin degan ma'noni anglatadi:

A, V, A + A, A + V, V + A, V + V, A - A, A - V, V - A, V - V.

Eslatib o'tamiz vertikal $\mid$ chiziq yoki degan ma'noni ifodalaydi. Faraz qilaylik biz $\langle \text{ikkilik kod} \rangle$ degan tushunchasini kiritmoqchimiz va ikkilik kod deganda 0 va 1 raqamlardan tashkil topgan ixtiyoriy ketma-ketlikni nazarda tutamiz. Umuman olganda, 0 va 1 ning o'zlari ham ikkilik kod va ularning yoki 0 va 1 raqamlaridan birortasini yozsak, yana ikkilik kod paydo bo'ladi yuqorida keltirilgan fikrlarni quyidagi metaformulalar yordamida oddiy va qisqa ko'rinishda ifodalash mumkin.

$$\langle \text{ikkilik raqam} \rangle ::= 0 + 1$$

$$\langle \text{ikkilik kod} \rangle ::= \langle \text{ikkilik raqam} \rangle + \langle \text{ikkilik kod} \rangle \langle \text{ikkilik raqam} \rangle$$

metoformulalarda ishlatiladigan figurali qavs { }, uning ichidagi konstruksiyaning ko'p marta takrorlanishini ifodalaydi. Yuqoridagi ikkilik kod tushunchasi figurali qavslar yordamida quyidagiga kiritilishi mumkin.

$$\langle \text{ikkilik raqam} \rangle ::= 0 + 1$$

$$\langle \text{ikkilik kod} \rangle ::= \langle \text{ikkilik raqam} \rangle \{ \langle \text{ikkilik raqam} \rangle \}$$

2. Paskal tiliga kirish

Paskal programmalash tili N. Virt (Shveytsariya Informatika instituti professori) tomonidan 1968 yilda yaratilgan. 1992 yilda Borland firmasi Paskal tilining SHEHMdagi MS-DOSda amal g'iluvchi Turbo-Pascal programmalash paketlarini chiqardi. TP juda keng tarqaldi va programmalash tarixida foydalanuvchilar sonining oshish sur'ati bo'yicha bu tilga teng keladigani yo'q. Buni Paskal tilida algoritmlarning fundamental va eng muhim

konsepsiyalarining oydin va tushunishga oson shaklda berilganligi bilan tushintirish mumkin. Programma tuzish paytida programma tuzuvchi uning ixtiyoriga programmalash tili tomonidan beriladigan kategoriyalar bilan fikr yuritadi.

Paskal tilini yaratishda ikki maqsad nazarda tutilgan:

- tushinarli va tabiiy ravishda ifodalangan bir kator fundamental tushunchalarga asoslangan programmalashni o'rgatishga qulay bo'lgan tizimli predmet sifatida yaratish;
- mavjud bo'lgan hisoblash mashinalarida tilni ishlatishning ishonchliligi va samaradorligi.

Paskal programmalash tili eng muhim ikkita tamoyilga asoslangan:

- strukturali programmalash tamoyili (unga klassik strukturali programmalashning asosiy boshqaruvchi strukturalarini amalga oshirish vositalari kiradi);
- ma'lumotlarni tashkil qilishning strukturali tamoyili (u K. Xoar tomonidan taklif etilgan strukturali turlarning to'liq to'plamini oddiy o'zgaruvchilar, massivlar, ketma-ket fayllar, yozuvlar, variantli yozuvlar, ko'rsatkichlar va undan berilganlarning yangi turlarini yaratishning taraqqiy topgan vositalarini o'z ichiga oladi);

Paskal programmalash tili, zamonaviy programmalashda muhim ahamiyatga ega bo'lgan, programmalarning to'g'riligini isbotlovchi analitik usullarning taraqqiy topishida katta rol o'ynadi. U hozirgi kungacha programmalashning to'g'riligini isbotlashga imkon beradigan programmalash sistemalariga ega bo'lgan yagona tildir.

Paskal programmalash tili, programmaning qismlari yoki butun programma to'g'risida turli xil mulohazalarni isbotlashni ta'minlaydigan, yaxshi matematik apparatga asoslangan. K. Xoarning aksiomatik sistemalari yordamida ifodalanadi. Programmalarni ishlashini "sozlash"ning eski usullari o'rniga, hozirgi paytda paydo bo'layotgan programmalarning to'g'riligini tekshiradigan avtomatik sistemalardan foydalanish maqsadga muvofiqdir.

Albatta, hamma o'zgaruvchilarni tasvirlashni talab qilinishi ortiqcha bo'lib ko'rinishi mumkin, lekin programma yozish osonligiga qaraganda, uning ishonchli bo'lishi muhimroqdir.

1982 yil A. Eddiman rahbarligidagi ishchi guruh tomonidan Paskal programmalash tilining Britaniya standarti yaratildi, u bugungi kunda xalqaro standartga ham aylandi.

Programmalash tillari va texnologiyalari

Programmalash tillari sun'iy tillar hisoblanadi, ularda sintaksis va semantik qoidalar qat'iy aniqlangan bo'ladi. Shu sababli Programmalash tillari tabiiy tillardan farqli ravishda jumalarni ko'pmazmunli va erkin talqin etishga yo'l qo'ymaydi. Buning asosiy sababi tildagi har bir ko'rsatma mashina tilidagi aniq bir buyruqqa o'tadi.

Sintaksis-bu qoidalar to'plami bo'lib, Programmalash tilida ruxsat etilgan belgilarning ketma-ketligi va asosiy ichki tuzilishni aniqlaydi.

Semantika-til birliklariga (so'z, so'z birikmalari, jumllariga) beriladigan qiymatlardir.

EHLar endi yuzaga kelgan paytda programma tuzishda, faqat mashina tillarida, ya'ni sonlar yordamida EHM bajarishi kerak bo'lgan amallarning kodlarida kiritilgan. Bu holda mashina uchun tushinarli sanoq, sistemasi sifatida 2 lik, 6 lik, 8 lik sanoq sistemalari bo'lgan. Programma mazkur sanoq sistemasidagi sonlar vositasida kiritilgan.

Yuqori bosqichli programmalashda, mashina tillariga qaraganda mashinaga moslashgan (yo'naltirilgan) belgili kodlardagi tillar hisoblanadi. Belgilar kodlashtirilgan tillarning asosiy tamoyillari shundaki, unda mashina kodlari ularga mos belgilar bilan belgilanadi, hamda xotirani avtomatik taqsimlash va xatolarni tashhis qilish kiritilgan. Bunday mashina moslashgan til - ASSEMBLER tili nomini oldi.

EHM faqat mashina tilini-buyruqlar, operand va sonlarning ikkilik sanok, sistemasidagi ko'rinishini «tushinadi». Shu sababli, Assembler tilida yozilgan programmalar uchun ularni mashina tiliga o'tkazuvchi «tarjimon» kerak bo'ladi.

Assembler tilidagi programmani mashina tiliga *o'tkazuvchi* maxsus programma translator deyiladi, ayrim hollarda uni assembler deb ham atashadi.

Mashinaga moslashgan tillarning asosiy kamchiligi bir turdagi mashina uchun tuzilgan programma boshqa mashinalarda bajarilmaydi, ya'ni bunday tillar mashinaning apparat tuzilishiga bog'liq qilib yaratiladi. Masalan, Praves mashinasi uchun tuzilgan programma IBM yoki YAMAHA turidagi mashinalar uchun o'tmaydi va aksincha.

Programmalash tillarining keyingi bosqichida protsedurali tillar joylashadi. Bu tillarning asosiy xususiyati shundaki, ularning sintaksis va semantikasi konkret EHM (protessor) buyruqlari tarkibiga bog'liq emas. Tuzilgan programmani konkret EHM ga bog'lashni translator amalga oshiradi.

Programmaning boshlang'ich matni operativ xotiraga kiritilgandan keyin u translatsiya qilinadi. Natijada, aynan shu mashina «tushinadigan» buyruqlar ketma-ketligi hosil bo'ladi va ularni bajarish asosida EHM masalani yechadi.

Programmani tarjimasi va uni bajarish jarayoni ikki usulda amalga oshirilishi mumkin:

Birinchi usul-komplilator deb nomlanuvchi usulda programmani EHMda bajarilishi-tarjima jaryoni to'liq, tugagandan keyin amalga oshiriladi. Bunda tarjima programmasini operativ xotirada saqlab turishga hojat yuq, shu sababli xotirani tejashga erishiladi.

Ikkinchi usul – interpretatsiya - programmadagi ayrim operatorlar ular tarjima qilingan zahoti bajariladi, shundan keyin navbatdagi operator tarjima qilinadi, bajariladi va hokazo. Bu rejimda xotirada interpretator programmasi ham bo‘lishi kerak, natijada qo‘shimcha xotira ishlatiladi va programmaning ishlashi kompilyatsiya qilinganga nisbatan sekin ishlaydi. Interpretatorning afzallik tomonlari programmalarni sozlash paytida ko‘rinadi, ya’ni yo‘l qo‘yilgan xatolar tuzatilishi mumkin. Kompilatorida bunday imkoniyat yo‘q.

Protsedurali tillarning umumiy xususiyatlarini ko‘raylik. Bu tillarda programmadagi yozuvlar umumiy qabul qilingan matematik yozuvlarga juda yaqin va tushunish uchun yengildir. Masalan, 7 va 5 sonlaring yig‘indisini hisoblash amalini programmalash tillarida yozilishini ko‘raylik:

1) mashina tilida

Buyruq	Operand adresi
0011 1111	1101 0001
0000 0101	1101 0010
0000 0110	1101 00111
0000 0111	1101 0100
1000 0000	1101 0101

2) mashinaga moslashgan tilda

MOV AX, 7

ADD AX, 5

MOV A, AX

3) protsedurali tillarda $A=7+5$.

Ko‘rinib turibdiki, protsedurali tillardagi yozuv tabiiy yozuvga yaqin.

Birinchilar qatorida yuzaga kelgan protsedurali tillar ichida FORTRAN (FORMula TRANslation - formulalarni boshqa ko‘rinishga aylantirish). O‘zining sodda tuzilishi, samarali translatsiyalanishi bu tilni nafaqat shu kungacha saqlanib qolishiga sabab bo‘ldi, balki uning ilmiy tadqiqot, injener-texnik masalalarini yechishda eng samarali til sifatida rivojlanib, keng tarqalishiga sabab bo‘ldi.

Fortran tili Beysik tilining asosiga qo‘yildi. Beysik so‘zi inglizcha «VASIC-Veginners All-purpose Symbolic Instruction Sode» ma’noni anglatadi. Bu jumla quyidagicha tarjima kilinadi: boshlovchilar uchun ko‘pmaqsadli belgili buyruqlar tili.

Beysik tili mutaxassis bo‘lmagan odamlarni programmalash texnologiyalariga jalb qilish imkoniyatini berdi. Beysik tilida programma operatorlarining kamligi, tuzilishining soddaligi uni

programmalashga o'rgatish tili sifatida keng tarqalishiga sabab bo'ldi. Bu til asosan maktablarda o'rgatiladi.

Algol-60 tili Fortranga nisbatan takomillashgan til, unda tuzilgan programmalar moslashuvchanlik va yuqori ishonchlilik xususiyatlari bilan xarakterlanadi.

Prolog va Lisp tillari sun'iy tafakkur sohasidagi masalalarni yechishga moslashgan. Bu tillar belgili ma'lumotlarni qayta-ishlash orqali matematika va mantiqiy masalalarni yechishga eng qulay tillardir.

1971 yilda paydo bo'lgan Paskal tili programmalash texnologiyalariga strukturali programmalash g'oyasini olib kirdi. Natijada bitta masala bir nechta bo'laklarga bo'lingan holda alohida yechilishi va oxirida bitta programmaga jamlanish imkoniyati yuzaga keldi.

Si tili zamonaviy kompyuterlar xususiyatlarini o'zida aks ettirgan va samarali programmalar yaratishda Assemblerga murojaat qilmaydigan til sifatida namoyon bo'ldi. Bu til UNIX operatsion sistemasida yozilgan.

Ayni paytda programmalash texnologiyasining istiqbolli yo'nalishlaridan biri obyektga yo'naltirilgan programmalash tillaridir (объектно-ориентированное программирование).

Obyektlar ko'p ishlatiladigan programma modullaridir. O'z navbatida obyektlar ikkita qismdan tashkil topadi: metodlar va o'zgaruvchilar. Metodlar protsedura va funksiyalar to'plami bo'lib, obyekt ishlash algoritmini aniqlaydi. O'zgaruvchilar turli xil qiymatlarni qabul qiladi.

Obyektga moslashgan programmalashning asosiy mohiyati shundaki, sodda obyektlardan vorislik orqali murakkab obyektlar shajarasini yaratish mumkin. Bunda biror masalani yechish deganda masala algoritmgiga mos obyektlar shajarasini qurish tushiniladi.

Obyektga yo'naltirilgan tillarning yaqqol namunasi - bu Delphi vizual programmalash muhitidir. Bu muhit Turbo Paskal tili asosida yaratilgan bo'lib, unda programma yaratish muhitidagi mavjud komponentalarni loyihalash asosida bo'ladi. Shuning uchun ham bosh programma Rgoject deyiladi. Delphining yana bir muhim xususiyati vizual programmalashdir, ya'ni programma yaratilayotgan paytda tuzuvchi hosil bo'ladigan programma shaklini ko'rib turadi va grafik interfeys yordamida komponentalar shaklini o'zi ma'qul topgan joyiga qo'yishi mumkin.

Visual Basic tili ham obyektga moslashgan til hisobalanadi va Windowsning ofis programmalarida makroslar yozish uchun til sifatida ishlatiladi.

Internet uchun programmalar yaratish uchun NTML (Nureg Text Markur Language) 1989 yilda yaratilgan bo'lib, WWW sahifalarini yaratish uchun asosiy til bo'lib qolmoqda. Bu til yordamida sahifa poligrafik usulda yaratishilishi mumkin, unda matn, audio, video, animatsiya va boshqa ma'lumotlar joylashishi mumkin.

Java tili kompyuter tarmoqlarida amal qiladigan programmalarni yaratishga mo'ljallangan, ya'ni Java tilida yozilgan programmlar global tarmoqlarda dinamik reklamalarni (animatsiya, teletayp lentlari) global tarmoqda joylashtirish uchun ishlatiladi, Java tilining afzalligi shundaki, u turli operatsion sistemalarda ham birdek amal qilaveradi. Java programmasi bir paytda Windows, UNIX va Macintosh operatsion sistemalarida ishlashi mumkin.

Quyidagi jadvalda programmalash tillari haqida ma'lumotlar keltirilgan.

Til	Yaratilgan yili	Mualliflar	Tashkilot, firma
Ada	1979-80	Jean Ichbian	Cii-Honeywell (Fransiya)
Algol	1960		International Commitee
ARL	1961-1962	Kenneth Iverson, Adin Falkoff	IBM
DELPHI	1995		Borland
BASIC	1964-1965	JohnKemeny, Thomas Kurtz	Dartmouth Colleje
C	1972-1973	Dennis Ritchie	Bell Laboratories
C++	1980	Bjarne Strostrup	Bell Laboratories
COBOL	1959-1961	Grace Murray Hopper	
Fort	1971	Charles H.Moore	
FORTTRAN	1950-1958	John Backus	IBM
HTML	1989	Tim Berners-Li	CERN, Jeneva
LISP	1956-1960	John MCCarthy	
LOGO	1968-70	Seymour Papert	MassachusetS Institute of Techn.
Pascal	1967-1971	Niklaus Wirth	Federal Institute of Technology (Shveytsariya)
PL1	1964-1966		
PROLOG	1978	Alan Kalmeroe	
SIMULA	1967	Ole-Yoxan Dal, Kristen Nigaard	Norvegiya XM
Java	1995	Djeyms Gosling	Sun Microsystems

Mavzu 2.3

Paskal dasturlash tili

Reja:

- 1 Turbo Paskal tili alfaviti
2. Paskal tilida ma'lumotlarning turlari:
 - a). Standart turdagi ma'lumotlar (INTEGER, REAL, CHAR, BOOLEAN);
 - b) Sanaladigan va cheklangan turdagi ma'lumotlar;
 - v) Murakkab turdagi ma'lumotlar.
3. Paskal tilidagi standart funktsiyalar, arifmetik va mantiqiy ifodalar.
4. Xulosa.

Kirish

XVII asrda yashagan va dunyoda birinchi xisoblash mashinasini yaratgan frantsuz olimi Blez Paskal sharafiga shunday deb nomlangan PASKAL tili Shvetsariyaning Syurix shahridagi oliy texnika maktabining professori Nikols Virt tomonidan 70-yillarda yaratilgan bulib 1979 yilda standart Paskal deb tasdiklangan. Uzining soddaligi, mantiqiyli va samaraligi tufayli bu til butun dunyoga tez tarkaldi. Xozirgi paytda barcha xisoblash mashinalari, xususan, mikro EXM lar xam shu tilda ishlash imkoniyatiga ega. Dasturlar matnining tugriligini osonlik bilan tekshirish mumkinligini, ularning ma'nosi yakkol kuzga tashlanishi va oddiyli bilan ajralib turadi. Paskal tili ancha murakkab va kup vakt oladigan xisob ishlari bajarishga muljallangan tartiblashtirilgan dasturlar tuzishga imkon beradi. Yana bir afzalligi shundan iboratki, foydalanuvchi xatolikka yul kuymasligi uchun yoki xato yozib kuygan bulsa, tez tuzatib olishi uchun dasturda ishlatiladigan uzgaruvchilar oldindan kaysi turga (toifa) mansubligi dasturning barcha elementlari xakida ma'lumot tavsiflash bulimida mujassamlashgan buladi. Operator soni esa minimal darajada kamaytirilgandir. EXMda xisoblash uchun programma asosan ikki kismdan iborat buladi:

berilganlarni tavsiflash (bular ustida amallar bajariladi); amallarni tavsiflash (bu amallar berilganlar ustida bajarilishi kerak). Amallar operatorlar yordamida beriladi, berilganlar esa ta'riflar va tavsiflar yordamida beriladi.

Turbo Paskal tili alfaviti

Alfavit – bu Paskal tilida ishlatishi mumkin bo‘lgan belgi va simvollar to‘plamidir. Turbo-Pascal muhitning hamma belgilari ASCII (American Standard Code for Information Interchange) kodining standart belgilaridan tashkil topgan.

Tupbo Paskal tili alfavitiga quyidagilar kiradi:

- 1) lotin harflari: A, B, C, ..., Z va a,b,c,...,z;
- 2) arab raqamlari: 0, 1, 2, ..., 9;
- 3) maxsus belgilar: +, -, *, /, =, [,], <, >, ., , , (,), :, ;, @, ^, \$, ', {, }, #, (probel) ;
- 4) arifmetik amallar belgilari: +, -, *, /, div, mod;
- 5) taqqoslash amallari: <, <=, >, >=, =, <>;
- 6) Mantikiy amallar: NOT (inkor etish) , OR (mantikiy kushish) , AND (mantikiy kupaytirish);
- 7) kalit so‘zlar: absolute, and, array, begin, case, const, constructor, destructor, div, do, downto, else, end, external, file, for, forward, function, goto, if, implementation, in, inline, interface, interrupt, label, mod, nil, not, object, of, or, packed, program, procedure, record, repeat, set, shl, shr, string, then, to, type, unit, until, uses, var, virtual, while, for.

8. Tuplamlar ustida amallar: * (tuplamlarni kesishishi $\cap$), + (tuplamlarni birlashtirish, $\cup$), - (tuplamlarni ayirmasi), IN (tuplamga tegishlilik).

Xamma berilganlar quyidagi turlarga bulinadi: sodda tuzilishga (strukturaga) ega bulgan skalyar va strukturali. Paskal tilida Uzgarmaslarni 4 ta turi ishlatiladi: Butun(INTEGER), Xakikiy (REAL), Mantikiy (logik) (BOOLEAN) va simvolli (belgili) CHAR.

Nom (identifikator) Xarf bilan boshlanib, Xarf va rakamlar ketma-ketligidan iborat. Programmada nom uzgarmaslarni (konstanta), turlarni, funktsiya uzgaruvchilarini, kism programmalarni, fayllarni yozish uchun ishlatiladi. Kupgina EXM larda nomni maksimal uzunligi 8 tadan oshmasligi kerak.

Paskal tilida standart nomlar xam bor. Bular -TRUE, FALSE, INTEGER, REAL, BOOLEAN, CHAR, INPUT, OUTPUT, ABS, SQR, SIN, COS, EXP, LN, SQRT, READ, WRITE , WRITLN va x.k. Bu nomlarni boshka maksadlarda ishlatish mumkin emas.

Standart funktsiyalar

Standart funktsiyalar juda kup xollarda masala va misollarni echish uchun dastur tuzayotganda tez-tez uchrab turadi. Masalan: trigonometrik, logarifmik, eksponenta, ildiz chikarish va Xokazolar. Xar bir EXM uchun bunday funktsiyalarni xisoblash dasturi tuzilgan va EXM xotirasiga kiritilib, ular standart funktsiyalar deyiladi. Bu funktsiyalarga murojaat

kilinganda avval funktsiyalarni nomlari keyin dumalok kavs ichiga ularning argumentlari yoziladi. Kup ishlatiladigan standart funktsiyalarning ba'zilarini keltiramiz:

$(x(- ABS(x)$	$ex-EXP(x)$	
$x^2 - SQR(x)$	$\ln x-LN(x)$	TRUNC(x)- sonni butun kismga ajratish
		ROUND(x)-sonni yaxlitlash,
$x^{100}-EXP(100*LN(X))$		PRED(x) -oldingi elementni topish
		SUCC(x)- keyingi elementini topish
$\sin x-SIN(x)$	$\sqrt{x}-SQRT(x)$	
$\cos x-COS(x)$	$\arctg x -ARCTAN(x)$	

Ifodalar

Ifoda kiymatlarni xisoblash ketma-ketligini va bajariladigan amallarni aniklaydi. Ular kavs va amal belgilari bilan ajratil-gan uzgarmaslardan, uzgaruvchilardan, standart funktsiyalardan iboratdir. Programma strukturasi esa kuyidagi oltita bulimdan iborat buladi. Bulimlar albatta kuyidagi tartibda joylashgan bulishi kerak: belgilar (metkalar); uzgarmaslar; turlar; uzgaruvchilar; protsedura va funktsiyalar; operatorlar.

Paskal tilida suz deb bir nechta belgilar ketma-ketligi tushuniladi. Xizmatchi suzlar deb Paskal tilidagi standart nom tushuniladi. Bu nom maxsus ma'noni anglatadi va uni ma'lumotlarga berib bulmaydi. Maslan, PROGRAM, BEGIN, END va xokazo. Paskal tilidagi ma'lumotlarning elementlari bulib uzgaruvchilar, uzgarmaslar, izoxlar xizmat kiladi. Uzgaruvchilar deb xisoblash jaraenida uz kiymatini uzgartiradigan katta- liklarga aytiladi. Uzgaruvchilarningnomlari (identifikatorlar) - xarflardan yoki xarf va rakmlardan iborat buladi, belgilar soni 8 tadan oshmasligi kerak. Uzgarmaslar (sonst) - deb xisoblash jarayonida uz kiymatini uzgartirmaydigan kattaliklarga aytiladi. Bularga xam uzgaruvchilar kabi nom beradi. Izoxlar - dasturning ma'lum kismini tavsiflash uchun ishlatiladi va bu katorda xech kanday amal bajarilmaydi, ya'ni dasturning biror blokini yaxshirok tushunishga xizmat kiladi.

Izox (*,*) simvollari orasida beriladi. Katta *, * va [*,*] lar xam ishlatilishi mumkin. Paskal tilida ma'lumotlarning toifa-lari ikki xil buladi. Oddiy skalyar va murakkab. Skalyar toifa uz navbatida uzgaruvchi va standart toifalarga bulinadi. Uzgaruvchi toifaga kayd kilingan va chegaralangan toifalar kirsa, standart toifaga butun - INTEGER, xakikiy - REAL, mantikiy - BOOLEAN, belgili - CHAR toifalar kiradi. Murakkab toifaga esa tup-lamlar, massivlar, yozuvlar, fayllar, belgilar katori kabi toifalar kiradi. INTEGER - BUTUN tipidagi ma'lumotlar fakat butun sonlarni kabul kiladi. Ular Paskal tilida kuyidagicha ifodalanadi : 7,10,89 va x.k.

REAL - XAKIKIY turdagi ma'lumotlar xakikiy sonlarni kabul kiladi va ikki xil kurinishda buladi:

a) Kuzgalmas nuktali xakikiy sonlar (3.56, 0.88, 150.45)

b) Kuzgaluvchi nuktali xakikiy sonlar (0.546 = 546E - 3
96.78 *10² =96.78E2 ba x.k.)

CHAR - BELGI turidagi ma'lumotlar kiymati sifatida barcha belgilarni kabul kiladi. Bu ma'lumotlar apostrof ichiga olib eziladi. Masalan ('F','A','R','5','8' ba x.k.).

BOOLEAN - MANTIKEY turidagi ma'lumotlar. Bu nom ingliz matematigi Jorj Bul sharafiga shunday deb atalgan va shu turidagi uzgaruvchilar fakat ikkita kiymat kabul kiladi TRUE eki FALSE. Bu ma'lumotlar ustida solishtirish amallari (>,<,<=>=>),

$$y=e^{\sin(x)+\cos x}+\operatorname{tg}(x+3)$$

$$y=\operatorname{EXP}(\operatorname{SIN}(\operatorname{SQRT}(X))+\operatorname{COS}(\operatorname{SQR}(X)))+\operatorname{SIN}(X+\operatorname{SQRT}(3))/\operatorname{COS}(X+\operatorname{SQRT}(3)).$$

MANTIKEY IFODALAR

Mantikiy ifodalar uz ichiga mantikiy uzgarmaslarni, mantikiy uzgaruvchilarni, mantikiy amal belgilarini, mantikiy natija beradigan standart funktsiyalarni xamda DIV v a MOD funktsiya-larni kamrab oladi. Mantikiy ifodada xam amallar ustivorligiga kura bajariladi.

1. Kavs ichidagi amal va mantikiy inkor amali (NOT).

2. Mantikiy kupaytirish (AND),DIV ,MOD amallari .

3. Mantikiy kushish (OR) va solishtirish amallari .

1 3 2 6 5 4

Misol. (A>3) AND (A=C+3) OR NOT (B=5)

:

1) FALSE

2) FALSE

3) FALSE

4) FALSE

5) TRUE

6) TRUE

Nazorat savollari:

1. Ma'lumotlarning toifalari turini ayting.

2. Skalyar toifa necha turga bulinadi ?
3. Uzgaruvchi toifa necha turga bulinadi ?
4. 4.Standart turdagi toifani sanab bering.
5. Murakkab turdagi toifani sanab bering.
6. Mantikiy ifodalar tarkibi kanday ?
7. Paskal tilining boshka tillardan farki ?
8. Xizmatchi suz nima ? Sanab bering.
9. Izox nima ? Uzgaruvchi nima ? Uzgarimas nima ?

Mavzu 2.4: Paskal tili dasturning asosiy tarkibiy qismi

Reja :

1. Dastur tushunchasi
2. Paskal tilida dastur tarkibi.
3. Dasturning asosiy bulimlari.
4. Operator tushunchasi. Paskal tildagi operatorlar tasnifi.
5. Sintaksis diagrammalar.
6. Xulosa.
7. Foydalanilgan adabiyotlar ruyxati.

- 1.N.Kultin Samouchitel Programmirovaniye v Turbo Pascal 7.0 i Delphi. Moskva Sank-Peterburg. 1999 g.
- 2.V.G.Abramov i dr. Vvedenie v yazk Paskal.-M.:Nauka,1988.
3. Sagatov M.V.,Yakubov O.X. Informatika (maruzalar matni) Toshkent- 2000 y.
- 4.D.B.Polyakov, I.Yu.Kruglov. Programmirovaniye v srede Turbo Paskal (versiya 5.5)-M.:MAI AG`O «ROSVUZNAUKA», 1992.

Kirish

Xozirgi kunda dasturlash tillarini u eki bu belgisi buyicha sinflash mumkin. Dasturlash tilining EXM ga bogliklik darajasi buyicha sinflash eng umumiy xisoblanadi. Bu belgiga karab dasturlash tillari mashinaga boglik va mashinaga boglik bulmagan tillarga bulinadi . Mashinaga boglik tillar uz navbatida mashina tillari va mashinaga muljallangan tillarga ajratiladi. Dasturlash tilining mashina tiliga yakinligi darajasini ta'riflash uchun til darajasi tushunchasi kullaniladi. Mashina tili nolinchii daraja deb kabul kilingan bulib, sanok boshi xisoblanadi. Odamning tabiiy tili eng yukori darajadagi til deb karaladi. Mashinaga boglik bulmagan tillar xam ikki turga bulinadi: birinchisi protseduraga muljallangan , ikkinchisi muammoga muljallangan. Protseduraga muljallangan tillar turli masalalarni echish algoritmlarini (protseduralarini) tavsiflashga karatilgan, shuning uchun ular kupincha oddiy kilib algoritmik tillar deb ataladi. Ushbu tillar echilaetgan masalalar xususiyatlarini tula xisobga oladi va EXMning turiga deyarli boglik emas. Bu xildagi tillar tarkibi mashina tiliga karaganda tabiiy tilga, masalan , ingliz tiliga yakinrok. Xozirgi kunda xisoblash, muxandis-texnik , iktisodiy, matnli va sonli axborotlarni taxlil kilish va boshka masalalarni echish tillari ma'lum. Masalan: FORTRAN tili 1954 yili

ishlab chikilgan bulib FORMULA TRANSLACION- formula tarjimoni degan ma'noni anglatadi va ilmiy, muxandis-texnik masalalarni xisoblashda kullaniladi. ALGOL tili 1960 yili yaratilgan bulib, ALGORITMIC LANGAUGE- algoritmik til degan ma'noni anglatadi va ilmiy-texnik masalalarni xisoblashlarga muljallangan.

KOBOL tili 1959 yili yaratilgan bulib, Common Businees Oriented Langaue - savdo- sotik masalalariga yunaltirilgan til degan ma'noni anglatadi. Korxona va tarmokning moddiy boyligini, moliyasini, ishlab chikargan maxsulotini xisobga olish bilan boglik iktisodiy masalalarni echish uchun ishlatiladi.

PLG`1 tili 1964 yilda yaratilgan bulib, Programming LangaueG`1-

1- tartib rakamli dasturlash tili ma'nosini anglatadi. Ushbu til universal tillar turkumiga kiradi. Bu tilda ishlab chikilgan dasturlar EXM ni yangisi bilan almashtirilganda kaytadan tuzib chikilishi zarur emas.

BEYSIK (BASIC) -Beginner's All Purpose Sumbolic Instruction Code-boshlovchilar uchun kup maksadli dasturlash tili. Xisoblash algoritmlarini ezish uchun kullaniladigan algoritmik til.

1. Dastur deb maxsus tilda yozilgan EXM da masala echish uchun kadamma kadam yozilgan anik va mukammal instruksiyaga aytiladi. Paskal tilidagi dastur tarkibi ikki asosiy kismdan iborat buladi. Bular dastur sarlovxasi va dastur tanasi. Dastur tanasidan keyin nukta kuyiladi, bu nukta dastur oxirini bildiradi. Dastur sarlovxasi dastur tanasidan nukta vergul bilan ajraladi:

<Dastur > ::q<dastur sarlovxasi >; <dastur tanasi >

dastur sarlovxasi PROGRAMM xizmatchi suz bilan boshlanadi bu suzdan keyin shu dasturga berilgan nom yoziladi. (bu nom dastur ichida ishlatilishi mumkin emas). Dastur nomidan keyin yumalok kavs ichida dastur parametrlar ruyxati beriladi. Bu parametrlar fayllar nomi bulib ular orkali dastur tashki muxit bilan boglanadi. Mavjud bulgan kuplik fayllar orasidan ikki standart matnli fayllar ajratilgan. Bular dasturni boshlangich ma'lumotlar va ma'lumotlarni kayta ishlash natijalari bilan boglaydigan INPUT BA OUTPUT lardir.

Dastur sarlovxasi yozilishini kuyidagi misolda kuramz.

PROGRAM PASKAL (INPUT,OUTRUT);

Paskalni kup versiyalarida INPUT BA OUTRUT yozilmasligi xam mumkin. Umumiy xolda dastur tanasi (bloki) 6 bulimdan iborat va ular aniklangan ketma - ketligida joylashgan bulishi shart. <blok>::q<belgilar bulimi >

<konstantalar bulimi>

<toifalar bulimi>

<uzgaruvchilar bulimi >

<protsedura va funktsiyalar bulimi>

<operatorlar bulimi>

Dasturni asosiy vazifasi bu EXM ga malumotlarni (berilganlarni) kayta ishlash buyicha amallar berish. Bu amallar operatorlar yordamida beriladi. Shuning uchun xam operatorlar bulimi asosiy xisoblanadi va xar kanday dasturda ishtirok etishi shart. Operatorlar bulimidan oldingi bulimlar ba'zi dasturlarda ishtirok etmasliklari xam mumkin. Ular tavsifsiflash bulimlari deyiladi.

Belgilar bulimi.

Dasturni ixtiyoriy operatorning oldiga belgi va ikki nukta kuyib kolgan operatorlardan ajratish mumkin. Operatorning belgisi bittadan ortik bulishi mumkin emas. Operatorning belgilari bir xil bulishi mumkin emas. Operatorning belgisi uning nomi sifatida karaladi. Dasturda ishlatilayotgan xar bir belgi avval belgilar bulimida tavsiflash yuli bilan e'lon kilinishi kerak :

<belgilar bulimi >:: q<bush > label <belgi > ,<belgi >;

Belgilar bulimi label (belgi) xizmatchi suz bilan ochiladi, undan keyin belgilar ya'ni ketma - ket vergul bilan ajratilgan konstanta shaklida yoziladi. Belgilar bulimida e'lon kilinayotgan belgilar ixtiyoriy ketma-ketligida yozilishi mumkin. Dasturdagi ketma-ketligiga rioya kilmasa xam buladi. Belgilar (0-9999) orasidagi sonlardan olinadi.

Misol.

label 4, 5, 25

Konstantalar, uzgarmaslar bulimi.

Konstanta deganda oydin bir u yoki bu toifadagi kiymatni tushunamiz, dastur ishlash jarayonida bu kiymat uzgarmaydi shu sababli uzgarmas deb xam yuritiladi. Paskal tilida kuyida konstanta katori kurinishi xam bulishi mumkin:

< liter katori > ::q`< litera >< litera >`

misol "Informatika"

Konstantalar tavsiflarining xammasi konstantalar bulimi tarkibida bulishi kerak.

< konstantalar bulimi > ::q <bush> !const<

< konstanta tavsifi > ;<konstanta tavsifi>;

< konstanta tavsifi > ::q <konstanta nomi> q

q < konstanta > Konstantalar bulimi const xizmatchi suz bilan boshlandi va nukta vergul (;) bilan tugaydi.

Misol:

const Aq25; Vq4.5; KAFq"INFORMATIKA";

Toifalar bulimi

<toifalar bulimi>::q <bush> ! type < toifa tavsifi>;<toifa tavsifi>;

Avvalgi ma'ruzalardan bizga ma'lumki Paskal tilida 4 standart toifadagi qiymatlar bor bulib: integer (butun), real (xakikiy), char (matnli), boolean (mantikiy). Bu toifadagi qiymatlar bilan bir katorda Paskalda boshka toifadagi qiymatlarni xam ishlatish mumkin. Lekin standart toifalardan farkli ravishda boshka toifalar dasturda anik tavsiflanishi kerak. Bu tavsiflashni quyidagilarda kurishimiz mumkin:

<toifani tavsiflash>::q<toifa nomi>q<toifa>

<toifa >::q<toifa nomi><toifa berilishi>

Ikki metaformulani ma'nosi shundayki toifani tavsifida <tip> komponentasi sifatida standart toifa nomi xam bulishi mumkin.

Misol

TYPE

Butunq integer; keyinchalik v a r n,m: butun;

kilib ishlatga buladi.

Toifalarning xamma tavsiflari toifalar bulimida berilgan bulishi kerak. Toifalar bulimi type (toifa) xizmatchi suzi bilan boshlanadi, keyin toifalar tavsifi yoziladi. Toifalar tavsifi bir biridan nuqta vergul (;) orkali ajratiladi.

<toifalar bulimi>::q<bush>!type <toifa tavsifi>;<toifa tavsifi>;

Misol: type mantiqqboolean; hafta(dush, sesh, chor, pay, jum, shan,yaksh); ish kuniqses...jum;

Bu misolda 3 toifa tavsiflari bor. Birinchi katorda standart mantikiy toifaga boshka nom berilayapti - mantiq. Ikkinchisida yangi kayd kilingan toifa kiritilayapti va unga hafta deb nom berilayapti. Uchinchisida yangi chegaralangan toifa kiritilayapti unga ishkuni deb nom berilayapti.

Uzgaruvchilar bulimi.

Dasturda ishlayotgan xar bir uzgaruvchi oldindan elon kilingan bulishi kerak. Ushbu vazifani bajarish uchun Paskal tilida <uzgaruvchilar tavsifi > tushunchasi bor. Bu tavsifda xar bir ishlayotgan uzgaruvchiga nom beriladi. Keyinchalik dastur bajarilish jarayonida tavsiflangan uzgaruvchiga boshka toifadagi qiymat berilsa bu xatto deb xisoblanadi.

Aloxida uzgaruvchini tavsifi quyidagicha buladi.

<uzgaruvchi nomi >:<toifa >

Misol :x : real; y: integer;

Birinchi katorda dastur jarayonida ishlatishga real toifadagi x uzgaruvchi kiritilyapti, ikkinchisida u nomli butun qiymatli uzgaruvchi kiritilyapti. Paskal tilida bitta tavsif yordamida dasturda bir necha bir xil toifadagi uzgaruvchilarni kiritish mumkin.

Misol:

x,y,r,h:real;

Bu tavsif ish jarayoniga 4 ta x,y,z,h nomli xakikiy uzgaruvchilarni kirityapti. Uzgaruvchilar bulimi var (variable uzgaruvchi) xizmatchi suz bilan boshlanadi.

<uzgaruvchilar bulimi>::q<bush> var <uzgaruvchilar tavsifi>; <uzgaruvchilar tavsifi>;

<uzgaruvchilar tavsifi>::q<uzgaruvchi nomi>, <uzgaruvchi nomi>: <toifa>

Misol. var i,j,k: integer; x,h,sum, way: real;n,m:integer;day: xafta;

Dasturda ishlatilayotgan uzgaruvchilar fakat bir marotaba tavsiflanishi kerak degan fikr xato xisoblanadi.

Protsedura va funktsiyalar bulimi.

Boshka bulimlardan farkli ravishda bu bulim maxsus xizmatchi suz bilan belgilanmaydi. Bu bulimni boshi procedure yoki function suz bilan boshlanadi. Protsedura tavsifi xar doim funktsiya tavsifidan oldin turadi. Bu bulim dasturda standart protsedura va funktsiyalardan tashkari protsedura va funktsiyalar ishlatilayotgan bulsa, agar ularga dasturdan murojaat kilinayotgan bulsa ishlatiladi.

Operatorlar bulimi.

Bu bulim dasturning asosiy bulimi xisoblanadi.

<operatorlar bulimi>::qbegin<operator>;<operator>end.

Dasturni bajarish bulimi operatorlik kovuslari begin va ehd ichiga joylashgan operatorlar ketma-ketligini bajarishga kiritiladi. Operatorlar bir-biridan (;) yordamida ajritiladi. Shuni aytish kerakki protsedura va funktsiya bulimidagi algoritmlar fakat operatorlar bulimidan murojaat kilingandayok bajariladi. Operatorlar yordamida masalani echish algoritmini bajarishga kerak bulgan (berilganlar) ustunidagi amallar tavsiflanadi. Funktsional vazifalari buyicha Paskal tilidagi operatorlar kuyidagi guruxlarga bulinadi: uzlashtirish, kiritish- chikarish, boshkarish, funktsiya va protseduralar tavsifi (kism dastrurlari).Uzlashtirish operatorlari-dasturda uzgartiruvchilarga ma'lum kiymatlarini uzlashtirish uchun ishlatiladi. Kiritish-chikarish operatorlari – tezkor xotiraga boshlan-gich ma'lumotlarni kiritadilar va yozuvga chikaruvchi kurilmalarga xisoblashlar natijasini chikaradilar. Boshkarish operatorlari –dastur operatorlarini ketma-ket bajarili-shini tashkil etadi. Funktsiya va protseduralarni aniklash operatorlari – dasturini kism-larga bulib bu kism-larni aniklash va nomlash imkoniyatini beradilar.

Uz tarkibiga kura operatorlar ikki turga bulinadilar: sodda va tarkiblashgan.

Uzining tarkibida boshqa operatorlar bulmagan operatorlar sodda operatorlar deyiladi. Sodda operatorlarga uzlashtirish, utish va protseduraga murojaat operatorlari kiradi. Tarkiblashgan operatorlarga shartli tanlash takrorlash va tarkibiy operatorlar kiradi. Tarkibiy operator BEGIN va END xizmatchi suzlar orasiga olingan operatorlar ketma-ketligidan iborat. Sintaksis diagramma xar biri tavsiflanayotgan smeta uzgaruvchini kiymati bulgan konstruktsiya-larni strukturasini grafik kurinishda tasvirlaydi.

Diagrammani aloxida elementi bulib, tilning asosiy simvoli yoki tushunchasi bulishi mumkin. Xar bir elementdan bitta yoki bir nechta strelka chikadi va undan keyingi elementga boradi. Xech kanday chikmaydigan strelka diagrammaga kirish xisoblanadi. Agarda stelka xech kanday elementga karalmagan bulsa unda u diagrammadan chikish xisoblanadi. Diagrammalarda asosiy simvolarni tushunchalardan ajratish uchun asosiy simvollar shunday uzi yoziladi, tushunchalar <> ga olinadi.

Nazorat savollari:

1. *Qanday dasturlash tillarini bilasiz ?*
2. *Dastur nima ?*
3. *Dasturning tarkibiy kislari nimalardan iborat ?*
4. *Dastur kanday asosiy bulimlardan iborat ?*
5. *Operator nima ?*
6. *Sintaksis diagramma nima ?*

Mavzu 2.5: Paskal tilida oddiy masalalarni dasturlash.

R e j a :

- 1.*Dasturlashtirishda bajariladigan xarakatlar kontseptsiyasi.*
- 2.*Uzlashtirish operatori.*
 - 2.1. *Arifmetik uzlashtirish operatori.*
 - 2.2. *Mantikiy uzlashtirish operatori.*
 - 2.3. *Belgili uzlashtirish operatori.*
- 3.*Tarkibiy va bush operatorlar xamda, nuktali vergul belgisi xakida tushuncha.*
- 4.*Ma'lumotlarni oddiy kiritish va chikarish.*
- 5.*Oddiy masalalarni dasturlashga misollar.*
- 6.*Xulosa.*

Kirish.

Shuni eslatib utish lozimki, dastur bu-masalani echish jarayonini ifodalovchi, ma'lum bir tilda ifodalangan, algoritm bulib, EXM bajaruvchi vosita xisoblanadi. Paskal tilida yozilgan dastur tugridan-tugri EXM ga emas, balki biror bajaruvchiga muljallanganki, unga kanday toifadagi kiymatlar bilan ishlay olishi va kushimcha kursatmalarsiz kanday amallarni bajara olishi belgilab berilgan bulishi kerak. U yoki bu masalani echish uchun kattaliklar ustida bajarilishi kerak bulgan amallarni kursatish uchun algoritmik tilda operator tushunchasi xizmat kiladi.

1. Dasturlashtirishda bajariladigan xarakatlar kontsepsiyasi.

Dasturdagi xar bir operator ma'lumotlarga ishlov berishning mustakil, mantikan tugallangan boskichini ifodalaydi. Paskalda operatorlarning 8 ta turi kuzda tutilgan. Xar kanday masalani echish (bu berilgan ma'lum kattaliklardan) anik koidalar buyicha, berilgan kattaliklardan boshka kattaliklardan boshka kattaliklarni keltirib chakarish jarayonidan iboratdir. Yangi kattaliklarni xosil kilish koidasi Paskalda ifodalar orkali beriladi. Oddiy xollarda masalaning echimini bitta formula yordamida kursatish mumkin. Masalan, tugri burchakli uchburchakni berilgan a va v katetlari uzunligi yordamida gipotenuza uzunligini topish uchun formula yordamida xisoblash utkazish kifoya. Mana shunday, berilgan ifodaga karab, yangi kiymatni xisoblash koidasini berish uchun paskal operatorlaridan biri-uzlashtirish operatori ishlatiladi. Bunda berilgan formula buyicha xisoblangan kiymat biror uzgaruvchiga beriladi. Dastur tuzish jarayonida fakatgina uzlashtirish operatorlari bilan chegaralanishning iloji yuk. Dasturda xisoblashlarning barcha yullari kuzda tutilgan bulishi va kanday xollarda u yoki bu xisoblash yuli tanlanishi kerakligi xakida axborot berilishi zarur. Xisoblash jarayonlarni tashkil etish uchun tanlash operatorlari va utish operatorlari kuzda tutilgan. Bu kategoriyaga yana ulash operatorini xam kiritish mumkin: bu xam tanlash operatori ammo u yordamida amallar emas, balki kayta ishlashga beriladigan ma'lumotlar tanlanadi. Paskaldagi ma'lumotlarning aralash toifasi sababli bu operator mavjuddir. Paskal tili etarlicha samaradagi dasturni olish uchun zamonaviy EXM ning umumiy tavsifdagi tomonlarini xisobga oladi. Kup oddiy algoritmlarni dasturlashtirishda asosan uzlashtirish, kiritish - chikarish operatorlari-dan foydalaniladi. Kuyida shu operatorlarni kurib chikamiz.

2. Uzlashtirish operatori.

Masalaning echish jarayoni kator bajariluvchi boskichlarga bulinib ketadi. Bu boskichlarning xar birida ma'lum kiymatlar buyicha yangi kiymatlar xisoblanadi. Bu xisoblangan kiymatlarning

ba'zilar esa oralik qiymatlar bulib, keyingi boskichlar uchun boshlangich qiymat bulib xisoblanadi. Yangi qiymatlar xisoblash uchun ifoda tushunchasi xizmat kiladi, xar bir ifoda bitta qiymatni xisoblash koidasini belgilaydi. Xisoblangan qiymatni xisoblash jirayonini keyingi boskichida foydalanish uchun eslab kolish zarur, bunday eslab kolish xisoblangan qiymatni ma'lum uzgaruvchiga berish yuli bilan amalga oshiriladi. Bunday amalni bajarish, asosiy operatorlardan biri xisoblanuvchi qiymat berish ya'ni uzlashtirish operatori bilan bajariladi. Uzlashtirish operatori sintaxis jixatdan kuyidagicha aniklanadi: $\langle \text{uzlashtirish operatori} \rangle :: q \langle \text{uzgaruvchi} \rangle : q \langle \text{ifoda} \rangle$

Bu erda 2 ta belgidan iborat bulgan asosiy belgi “:q” “uzlashtirish” deb ukiladi. Uzlashtirish operatorining bajarilishida “:q” belgisining ung tomonidagi ifoda xisoblanib, belgining chap tomonidagi uzgaruvchi tomonidan uzlashtiriladi. Shunday kilib, uzlashtirish operatori mustakil mantikiy tugallangan xisoblash jarayonini ifodalaydi. Uzlashtirish operatori bajarilishi natijasida ma'lum uzgaruvchilar keyingi boskichlarda ishlatilishi mumkin bulgan yangi joriy qiymatni kabul kiladi. Turli toifadagi ifodalar aslida kup uxshashliklarga ega bulib, ularning barchasi operandlardan, amal belgilaridan, amallarni ixtiyoriy ketma-ketligini bajarish imkoniyatini beruvchi kavslardan tashkil topadi. Bunda operandlar uch turli: uzgarmas, uzgaruvchan va xisoblanadigan buladilar. Uzgarmas operand dastur tuzilayotganda ma'lum bulgan va bajarilishi davomida uzgarmas bulgan qiymat bulib, boshkacha aytganda uzgarmas operand - bu biror toifadagi konstantadir. Uzgaruvchi operand qiymati dastur bajarilishi davomida aniklanadi va uzgarishi mumkin. Ammo bu operand katnashgan ifodaning xisoblanishidan oldin uning qiymati aniklangan bulishi lozim. Bunday operandlar bulib Paskal tilining uzgaruvchilari xisoblanadi. Sintaxis jixatdan uzgaruvchi - qiymatni uzlashtiradigan, shu uzgaruvchi nomi sifatida kullaniladigan identifikatordir. Xisoblanadigan operand qiymati xatto xisoblashdan avval xam aniklanmagan bulib, ifoda xisoblanishi jarayonida aniklanadi. Xisoblanadigan operandlar sifatida Paskal funktsiyalarini kursatish mumkin.

Arifmetik uzlashtirish operatori uzgaruvchiga arifmetik toifadagi, ya'ni REAL yoki INTEGER toifadagi qiymatni berish uchun xizmat kiladi. Bunday operatorning ung tarafida arifmetik ifoda, ya'ni shu toifalardagi qiymatni xisoblash koidasini beruvchi ifoda bulishi kerak. Agar uzlashtirish operatorining chap kismidagi uzgaruvchi REAL toifaga ega bulsa, arifmetik ifoda REAL toifadagi, xamda INTEGER toifadagi qiymat xam bulishi mumkin (bu xolda olinadigan butun qiymat avtomatik tarzda xakikiy qiymatga aylantiriladi). Agar chap kismidagi uzgaruvchi INTEGER toifaga ega bulsa, unda Arifmetik ifoda, albatta, shu toifadagi qiymatni saklash kerak. Arifmetik ifodaning barcha operandlari REAL eki INTEGER toifada bulish-lari kerak. Asosiy operandlar sifatida uzgarmaslar (ishorasiz son eki konstanta nomi), uzgaruvchilar va funktsiyalar ishlatiladi. Arifmetik ifodaga kuyidagi misollarni (ifodaning ung tarafida uning

xisoblash tartibi , xar bir ifoda natijasi va ishlatilgan funktsiyalar qiymatlari toifalarini xisobga olgan xolda) keltirish mumkin.

$2*3Q4*5$	$((2*3)Q(4*5)q26)$
$9 \text{ div } 4G^2$	$((9\text{div}4)G^2q1.0)$
$40G^5G^{10}$	$((40G^5)G^{10}q0.8)$
$_sqrt(sqr(3)Q32G^2)$	$(-sqrt(sqr(3)Q(32G^2))q-5.0)$
$((2Q4)G^{10}Q2G^4)*2$	$((((2Q4)G^{10})Q2G^4))*2q2.2)$

Arifmetik uzlashtirish operatoriga misollar:

(bunda , x ,a,b,s,r-REAL toifadagi uzgaruvchilar, piq3.14159

bulgan xakikiy sonning nomi):

x:q0 (x uzgaruvchiga nolga teng bulgan qiymat berilayapti);

i:qiQ1 (i ning joriy qiymati bir birlikka orttirilayapti);

s:qsqrt(a*aQb*b) (a va b katetlar gipotenuzasi uzunligi xisobla-nayapti);

x:q 2*pi*r (r radiusli aylana uzunligi x xisoblanayapti).

Kuyidagi ezuvlar sintaksis buyicha uzlashtirish operatorlari emas eki Paskalda bunday ezib bulmaydi :

3:qiQ2 (chap tomonda konstanta kullanilishi mumkin emas);

xq2*ri*r ("q"belgi uzlashtirish operatorning belgisi emas);

iq5G^4(butun sonli uzgaruvchiga xakikiy qiymat berilmaydi);

x:qa*-b G^2 (ketma- ket ikki amal belgisini ezish mumkin emas);

Agar uzlashtirish operatorining chap tarafida BOOLEAN toifadagi uzgaruvchi kursatilgan bulsa operatorning ung tarafida mantiqiy qiymat (TRUE eki FALSE) ni xisoblovchi mantiqiy ifoda berilishi kerak.

Mantikiy ifodada konstanta, uzgaruvchi va funktsiyalar ishlatiladi, birok mantiqiy amalning xar bir operandi BOOLEAN toifada bulishi kerak. Mantikiy ifodada mantiqiy amal operandi bulib munosabat xizmat kiladi.

Bu tushunchani oddiylashtiraylik;

<arifmetik ifoda><takkoslash amali> <arifmetik ifoda> kurinishdagi ezuv - munosabatdir.

Bunda <takkoslash amali>::q<g`<g`qg` g`>g`. Agar arifmetik ifodalarning munosabati tugri berilgan bulsa, munosabat TRUE qiymatiga, aks xolda FALSE qiymatiga ega buladi.

Masalan, 3<5 munosabati TRUE, 3>q5 munosabati FALSE qiymatiga ega. Munosabatda katnashadigan ixtieriy arifmetik ifoda xakikiy eki butun sonli bulishi mumkin. Butun son xakikiy son bilan takkoslanganda oldin xakikiy songa utkaziladi, deb karaladi.

Mantikiy ifodaga misollar.

(d,b,s-mantikiy, x, u-xakikiy, k-butun sonli uzgaruvchi):

$x < 2 * u$	(munosabat)
TRUE	(konstanta)
D	(uzgaruvchi)
Odd9(k)	(funktsiya)
not not d	(mantikiy kupaytuvchi)
$(x > y \wedge 2)$	(mantikiy kupaytuvchi)
d and (xqu) and b	(mantikiy kushiluvchi)
(s or d) and (xqu) or not	(oddiy ifoda)

Mantikiy ifoda qiymatini hisoblash oddiydir.

Masalan, $d \text{ or } (x * u \wedge 2 > xQu) \text{ and not } b \text{ or } (x > 2 * r)$ ifoda qiymatini hisoblashda, bu erda keltirilgan barcha uzgaruvchilarning urniga ularning joriy qiymatlarini olish va ifodada berilgan amallar-ni, ularning makomi va kuyilgan kavslarni hisobga olgan holda, bajarish kerak. Agar uzlashtirish operatorining chap tomonida CNAR toifadagi uzgaruvchi kursatilsa, uning tomonida belgili ifoda, ya'ni CNAR toifa, ? boshqacha holda belgini anglash koidasi berilishi kerak. Belgili ifoda sifatida faqatgina shu toifadagi konsata, uzgaruvchi eki funktsiya berilishi mumkin. Belgili uzlashtirish operatoriga misollar:

(sum ,alpha , betta - CNAR toifadagi uzgaruvchilar):

sum : q“Q”;

alpha :qsum ;

betta: qsucc (sum).

Shu avvaldan ma'lumki, umuman, Paskal tilida arifmetik ifoda, mantikiy ifoda va x. k. tushunchalari yuk bulib, faqatgina, barcha toifadagi ifodalarni ularning ichiga oluvchi, bitta sintaksis buyicha aniqlanuvchi tushuncha - <ifoda>gina mavjud.

Nazorat uchun savollar:

1. Uzlashtirish operatori haqida ma'lumot bering.
2. Arifmetik uzlashtirish operator. haqida ma'lumot bering.
3. Mantikiy uzlashtirish operatori haqida ma'lumot bering.
4. Belgili uzlashtirish operatori haqida ma'lumot bering.
5. Tarkibiy va bosh operatorlar xamda, nuktali vergul belgisi xakida tushuncha.

Mavzu 2.6: Paskal tilida ma'lumotlarni oddiy kiritish, chikarish va tarkibiy operator

Reja:

1. Tarkibiy operator

2. Ma'lumotlarni oddiy kiritish va chikarish

Таянч сыз ва иборалар: READLN , WRITE, OUTRUT

1. Tarkibiy operator

Tarkibiy operator keltirib chikaruvchi operatorlar toifasidan xisoblanadi. Tushunishni osonlashtirish uchun Paskal dasturida u eki bu sintaksis konusturuktsiya kismida yagona operator kullanishi talab etiladi. Ammo algoritmda esa ba'zan, operatorlar ketma-ketligi ezilishiga tugri keladi. Tarkibiy operator ma'lum bir operatorlar ketma- ketligini operator kavslari begin (boshi) va end (oxiri) xizmachl suzlari orasiga olib birlashtiriladi.

<tarkibiy operator>::qbegin <operator>; <operator> end. Yukoridan kurinadiki yagona (tarkibiy) operatorga keltirilaetgan operatorlar soni bir dona bulish mumkin. Agar ularning soni ikki eki undan ortik buladigan bulsa, ularning orasi nuktali vergul bilan ajratiladi. Aytib utish lozimki, Paskalda nuktali vergul operatorlarni ajratuvchi sifatida ishlatiladi, ya'ni na avvalgi va na keyingi operator tarkibiga kirmaydi. Til buyicha tarkibiy oprator tarkibiga kiruvchi operatorlarga chikariluvchi, shuningdek tarkibiy operatorlardan biri bulishi xam mumkin. Ya'ni tarkibiy operator rekursiv xarakterga ega.

Tarkibiy operatorlarga misollar :

```
begin i:qD end ; begin y:qx G`2; x:qQh end
```

```
begin k:q2; begin i:qD; xisobchi :qD end end
```

Tarkibiy operatorlarda uning tarkibiga kirgan operatorlar ezilish ketma-ketligida bajariladi. Ixtieriy Paskal dastur tanasining operatorlar kismida xech bulmaganda birgina tarkibiy operator mavjud buladi. Bush operator -xech kanday xarakatni bajarmaydigan operatordir. Bush operatorga koida buyicha operator mavjud bulishi kerak bulgan joydagi yozuvning yukligi tugri keladi. Undan sung nuktali vergul, kuyish kerak.

Masalan;

```
A:qB;R:q2;; K:q7.2
```

Bu erda uchinchi operator bushdir. Tarkibiy va bush operatorlar shartli operatorlarda tez-tez kullaniladi.

Ma'lumotlarni oddiy kiritish va chikarish

Sonli uzgaruvchilarga ularning qiymatini berishda qiymat berish operatoridan foydalanish mumkin.

Masalan:

A:q5 ; B:q-6.143

Dasturni uzgaruvchilarning turli qiymatlarida bajarish uchun READ-kiritish operatori muljallangan.

Kiritish operatori quyidagicha kurinishga ega

READ (a1,a2,...,an); Bunda, a1,a2,...,an - uzgaruvchi qiymatlarini ketma-ket standart INPUT protsedura faylidan oluvchi uzgaruvchilar. Aytaylik, A,B,C uzgaruvchilarga dastur bajarilishi davomida kuyidagi qiymatlarni berish kerak bulsin: Aq5,Bq17,Cq6.2.

Operator READ(A,B,C) kurinishga ega bulib, sonlar qiymatlarini dastur bajarilishi davomida kuyidagicha kiritish mumkin:5 17 6.2

ENTER.

Agar uzgaruvchi REAL toifada aniklangan bulsa, uning qiymatini butun son yoki xakikiy son kurinishida kiritiladi. Mashinaning uzi butun sonni xakikiy songa utkazib oladi.

Masalan,VAR A,B:REAL;

READ(A,B) operatorining ishlatilishida 4 5 ENTER kurinishida butun sonlarni kiritish mumkin.

READLN

Bunda ma'lumotlar kiritilgandan keyin satrga utish ta'minlanadi. Bunga kushimcha ravishda READLN (a1,a2,...,an); kiritish operatori xam mavjud, bunda avval a1,a2,...,an ga qiymat kiritilib, sung keyingi satrga utiladi. Bu operator oldingi ikki operatorga teng kuchlidir. EXM xonasidagi ma'lumotlarni displey ekraniga chikarish operatori -WRITE dir. Operatorning yozilishi quyidagichadir:

WRITE (a1,a2,...,an)

Bunda a1,a2,...,an oddiy uzgaruvchilar, yoki apostrof ichiga olingan belgilar katori bulishi mumkin va ular standart OUTPUT protsedura fayliga chikariladi.

WRITE (B ning qiymatiq',B) operatori displey ekraniga:

B ning qiymati q ,

va undan sung V uzgaruvchining qiymatini chikaradi. Butun va xakikiy sonlarni chikarish uchun WRITE operatorida FORMAT larni kursatish mumkin. U ikki nukta orkali uzgaruvchidan sung kursatiladi. Masalan WRITE (Y:5:2) operatori bilan Y ning iymatini chikarishda, Y ning xamma qiymatini chikarish uchun 5 ta pozitsiya ajratilishi, ulardan ikkitasi kasr kismi uchunligini anglatadi. Butun sonlarni chikarishda kasr kismi formati kursatilmaydi. Aytaylik, Nq179 butun sonli qiymatni chikarish kerak bulsin. Chikarish operatori buning uchun kuyidagi kurinishda bulishi mumkin: WRITE (' Nq',N :3)

Bu erda sonni tasvirlash uchun 3 pozitsiya ajratilgan. Agar formatni 3 dan ortik berilsa, masalan, WRITE ('Nq',N:5)

Unda sondan oldin ikkita bush joy tashlanadi: Nq 179, manfiy son uchun esa bitta bush joy tashlanadi: Nq -179.

Paskal tilida boshkacha chikarish operatorlari xam ishlatiladi

Parametrlarsiz chikarish operatori

WRITELN

display ekranida yangi satrga utishni taminlaydi.

WRITELN (a1,a2,...an)

chikarish operatori oldin a1,a2,..., an larning qiymatlarini chikaradi, sung yangi katorga utishni taminlaydi.

Shunday kilib, bu xam kuyidagi ikki operatorga ekvivalent WRITELN(a1,a2,...,an); WRITELN
Masalan, A,B,C ning qiymatlarini kiritish uchun kuyidagicha kursatish mumkin:

WRITE ('A,B,C qiymatlarni kiriting ');

READ(A,B,C)

Shunday kilib, A,B,C ning qiymatlarini kiritilishidan oldin ekranga kuyidagi xabar chikariladi:A,B,C qiymatlarni kiriting va shundan sunggina qiymatlarni kiritish mumkin,masalan, 5 17 6.2 ENTER.

Oddiy masalalarni dasturlash

Bazi bir oddiy masalalarning dasturlarini tuzamiz.

1.R radiusli sharning xajmini kuyidagi formula buyicha xisoblansin

$V = \frac{4}{3} \pi R^3$

Dastur kuyidagi kurinishda buladi.

PROGRAM E10(INPUT,OUTPUT);

CONST

PI=3.14;

VAR

R:REAL;Shar radiusi

V:REAL;Shar xajmi

BEGIN

WRITELN ('R radius qiymatini kiriting:');

READ(R);

$V := \frac{4}{3} * PI * R * R * R$;

WRITELN;

WRITELN ('natija:');

WRITELN ('Shar xajmiq',V:8:3);

END

2.Kuyidagi karshiliklardan tashkil topgan zanjirning umumiy karshiligini xisoblang: Ketma-ket karshilik R ketq $R1QR2$, parallel karshilik R parq $R1*R2G` (R1QR2)$ bulsin. $R1$ ni $R1$, $R2$ ni $R2$, R ketni $RKET$, R larni $RPAR$ deb belgilaylik. Zanjir karshiligini xisoblash dasturini tuzamiz:

PROGRAM ER(INPUT,OUTPUT);

VAR

$R1,R2$:REAL; karshiliklar

$RKET$:REAL; ketma-ket ulanish

$RPAR$:REAL; parallel ulanish

BEGIN

WRITELN (' $R1$ va $R2$ kiymatlarini kiriting:');

READ($R1,R2$);

$RKET$:q $R1QR2$;

$RPAR$:q $R1*R2G` (R1QR2)$;

WRITELN;

WRITELN ('Ketma-ket ulangan zanjir, Rq' , $RKET$:8:2);

WRITELN ('parallel ulangan zanjir , Rq' $RPAR$:8:2)

END

Nazorat savollari:

1. Kuyidagi belgilar ketma-ketligidan kaysilari ta'minlash operatori xisoblanadi:

a) $a:qb$; b) $a:qsQ1$; v) $a:qb-sqr(2)$;

g) $a*xQb :q0$; d) $z:q0$; e) $z:qzQ1$;

j) $z:qzQ1,2$; z) $y:qy$; i) $-y:qy$?

2. x va u uzgaruvchilarning kiymatlari mos xolda 0,3 va 0,2 teng bulsin. Ta'minlash operatori bajarilgach, bu uzgaruvchilarning kiymatlari kanday buladi ?

a) $x:qxQ2*u$; $u:quG`2$;

b) $y:q-y$; $x:qxQy$; $y:qyQ1$;

v) $x:q1$; g) $u:qxQu$?

3. Kuyidagi belgilar ketma-ketligidan kaysilari kiritish operatorlari ?

a) read (x,y,z); b) read x,y,z ; v) read (x);

g) $x:qread (x)$; d) read ($a:v$); e) read (a,vQs)?

4. Kuyidagi operatorlar ketma -ketligi bajarilgach , x va u uzgaruvchilarning kiymati kanday buladi ? Berilganlar 5.2 va 18.7 dan iborat.

read (x,y) ; t:qx; x:qu; u:q t,

5. Kuyidagi belgilar ketma-ketligidan kaysilari chikarish operatorlari bula oladi?

a) write (x, y); b) write (x,xQ1,xQ2); v) read(a); g) write (100) ;

d) print (y ,z); e) print u, z; j) write (x y z); z) wite (xQ2,2).

6. Kuyidagi operatorlar ketma-ketmaligi bajarilsa, natijaga kanday sonlar chikariladi?

x:q(sin(1)-1)Q2*abs(-2))G`cos(2-2);

u:qx*(skr(2)) ; write (x ,y).

7. Kuyidagi operatorlar ketma -ketligi bir xil kiymat kiritilganda xamma vakt xam bir xil natija buladimi ?

a) read (x,y); x;qx*x: write (x,y) xamda

read(x,y); write (sqr(x)); write(y).

v) read (x,y); z:q sqr(x)Q1; write (xG`y,yG`z,x) xamda

read (x,y); write (xG`sqr(x)Q1,yG`sqr (x)Q1,x)

8) Ikkita operatorlar ketma-ketligi :

u:quQv; v:q2*v xamda

v:q2*v; u:quQv , berilgan. Ular bir-biridan operatorlar ketma-ketligi bilan farklanadi. u va v ning barcha boshlangich kiymatlarida bu operatorlar ketma-ketligi u va v ning kiymatining bir xil uzgarishiga olib keladimi?

Mavzu 2.7-2.8: Tarmoklanuvchi xisoblash jarayonlarini algoritmash va dasturlash

Reja:

1.Kirish

2.Munosabat amallari va ularni kattaliklar orasidagi shartlar sifatida kullanishi.

3.Shartli utish operatorlari.

3.1.Shartsiz utish operatorining tulik kurinishi.

3.2.Shartli utish operatorining kiska kurinishi.

3.3. Tanlash operatori CASE.

4.Utish operatori.

5.Xulosalar.

6.Foydalanilgan adabiyotlar ruyxati.

1.Kirish.

Kup masalalarning echimi malum shart yoki shartlarning kuyilishiga nisbatan bajariladi. Bunday masalalarning algoritmik tavsiflari bizga oldingi mavzulardan malum va ular tarmoklanadigan xisoblash jarayonlari kurinishida tasvirlanadi. Tarmoklanish xisoblash jarayonlarining tula va kiska shartli kurinishi mavjud.

Tarmoklanuvchi xisoblash jarayonlari tarkibida yana tarmoklanishlar bulishi mumkin. Bunday tarmoklanishlari bor bulgan xi- soblash jarayonlari murakkab tarmoklanuvchi xisoblash jarayonlari deb ataladi. Bir operatorida bir nechta mantikiy ifodani tekshirish va mantikiy ifoda bajarilishiga karab utishni tanlash mumkin. Bu turdagi algoritmning tarkibi tarmoklanishning umumiy lashgani bulib, bir nechta echimdan bittasini tanlash asosida ishlaydi.

2.munosabat amallari va ularning kattaliklar orasidagi shartlar sifatida kullanishi.

Oldindan ishlab chikilgan algoritm asosida ishlayotgan dasturlar EXM da bajarilish davomida kattaliklar xisoblanibgina kolinmasdan balki ularni xossalarini taxlil kilish, bir-biriga takkoshlashga xam tugri keladi. Masalan, sonlar juft yoki tok, oddiy yoki murakkabligini, istalgan ikkita kattalikni teng yoki teng emasligini tekshirish va xokazolar. Algoritmik tilda kattaliklarni istalgan xossasi shu ondagi kiymatlari uchun bajarilishi yoki bajarilmasligi ShART deyiladi.

Masalan, aqv va vq3 tenglik aq3 va vq3.007 bulganda bajarilmaydi, ammo aq3 va vq3 bulganda bajariladi. N «tub son» degan xossa Nq19 uchun algoritmni yozishda algebraik ifodalardan foydalanilganda shart shunday bulishi kerakki, ijrochi kattaliklarni istalgan kiymatlarini shartni bajarish yoki bajarilmasligini tekshirish imkoniyatiga ega bulsin. Bunday tekshirishlar mantikiy ifodalarni taxlil kilish bilan bajarilishi mumkin. Mantikiy ifoda shunday ifodaki, uning kiymati ifodadagi kattaliklarning (uzgar mas va uzgaruvchi) kiymatiga karab ROST yoki YoLGON bulishi mumkin. (Paskal tilida TRUE-ROST, FALSE- YoLGON ni anglatadi).

Mantikiy ifodalarga kuyidagi munosabatlar misol bula oladi:

$A > B$, AqB , $X < 4,55$, $2QZ > 0$, $XQY < q1$ va $x.k$.

Kattaliklar orasidagi munosabatlarni shartlar sifatida kiritish algoritmni yozishda «shart» tushunchasiga anik matematik shakl berish imkonini beradi. Shartlar oddiy va murakkab bulishi mumkin. Agar kattaliklar orasidagi shart fakat bitta munosabat orkali berilgan bulsa, «oddiy shart» deb ataladi. Munosabat amallari sonli va matnli kattaliklarni solishtirishda ishlatilishi mumkin. Paskal tilida kuyi-dagi munosabat amallaridan foydalaniladi.

Amal belgisi	Tekshirilayotgan munosabat	Misol
q	teng	$2q2$; XqU
< >	teng emas	$2 < > 3$ $X < > U$

<	kichik	$2 < 3$	$X < U$
>	katta	$5 < 4$	$X > U$
<q	katta emas	$1 < qZ$	$X < qU$
>q	kichik emas	$1 > qZ$	$X > qU$

Munosabatning bajarilishi yoki bajarilmasligiga karab, u rost yoki yolg'on bulishi mumkin. Masalan, $2 < 3$ munosabati rost. $4 > 5$ munosabati yolg'ondir. Kattaliklar orasidagi shartlar VA, YO'KI, EMAS (Paskal tilida AND, OR, NOT) mantik amallari belgilari orkali boglanuvchi bir necha munosabatlardan iborat bulsa «murakkab shartlar» deb ataladi.

Masalan, $(U > q0 \vee X > Z) \wedge U < -3$ xamda $U > q0$ va $(X > 2 \vee U < 3)$

Masalan, $X < q5$

$X < 11$ tizim, $X > q5$ va $X < 11$ kurinishda yoziladi.

Matematik yozilish Algoritmik tilda yozilishi

1) $6 < qx < 10$ $x > q6$ $x > q6 \text{ AND } x < 10$

$X < 10$

2) $aqbq0$ $aq0$ $aq0 \text{ AND } bq0$

$bq0$

3) $1 < X < 4 \vee X > 1$

$X < q4$

$x > 1 \text{ AND } x < 4$

4). (x, y) nuqta $Y > X^2$

$Y < 2$

$y > qx^2 \text{ AND } y < 2$

Mantikiy amallar, oddiy mantikiy ifodalardan murakkablarini xosil kilishda ishlatiladi. AND amalining natijasi uning ikkala argumenti xam rost bulsagina rost buladi. OR amalining natijasi rost bulishi uchun argumentlardan birining rost bulishi etarli.

NOT amalining natijasi argumentning inkor kiymati ya'ni argument rost bulsa natijasi yolg'on. Argument yolg'on bulsa natijasi rost buladi. Bularni kuyidagi jadvaldan kurish mumkin

(R – rost, E - yolg'on)

X	U	$X \text{ AND } U$	X	U	$X \text{ OR } X$	NOT X	NOT U
R	R	R	R	R	R	E	E
R	E	E	R	E	R	E	R
E	R	E	E	R	R	R	E
E	E	E	E	E	E	R	R

Masalan: $4 < 5 \text{ AND } 5 < 100$ mantikiy ifoda rost

$\sin(X) > 1 \text{ AND } X < qX$ ifoda yolg'on

$2^0 > 0 \text{ OR } \cos(X) < q5$ ifoda rost

Shunday jarayonlar mavjudki, ularda bajarilayotgan operatorlardagi shartga karab dasturni u yoki bu kismiga utishga tugri keladi. Ana shunday jarayonlarga dastur tuzish uchun shartli uzatish operatoridan yoki oddiy utish operatoridan foydalanish mumkin. Utish operatorining umumiy kurinishi kuyidagicha;

GOTO n; Bu erda n belgi bulib, 0-999 oralikdagi sonlardan tanlanadi. Paskal tilida belgi butun yoki xarf bulib xech kanday ifodaga ega bulmaydi. Belgi tavsiflash bulimida albatta tavsiflangan bulishi shart. Utish operatori yordamida boshkaruv n – chi belgiga uzatiladi.

Misol:

GOTO 25;

:

:

:

25:y:qx*x;

Belgini dasturda: bilan ajratiladi. Bu operatorninig notugri kulanilishi dasturning bajarilishi xalakit beradi. Shuning uchun dasturda bu operatorning kamrok uchrashi maksadga muvofikdir. Paskal tilida sharli utish operatorining ikki turi mavjuddir: Shartli utish operatorining tulik kurinishini kurib chikamiz. Uning metoformulasi kuyidagicha yoziladi:

< tulik shartli operator>::q IF< mantikiy ifoda> THEN

<operator > ELSE < oporator >

yoki oddiyrok

IF < mantikiy ifoda > THEN operator 1 ELSE operator 2

Bu erda IF (agar),

THEN (u xolda) va ELSE (aks xolda) degan xizmatchi suzlar.

Tulik kurinishdagi shartli utish operatorining tarkibini yana kuyidagicha yozish mumkin.

IF B THEN A1 ELSE A2

Bu erda V mantikiy ifoda ;

A1, A2 -operatorlar .

Shuni takidlab utish kerakki, agar matikiy ifodalar, biz yukorida aytganimizdek, mantikiy amallar yordamida (AND,OR, NOT) murakkab kurinishga ega bulsa, ular kavslarga olib yoziladi .

Masalan (x>y) and (x>z).

Operatorning ishlash tartibi kuyidagicha: Agar keltirilgan mantikiy ifoda TRUE (rost) kiymatni kabul kilsa THEN xizmatchi suzidan keyingi operator bajariladi, aks xolda ELSE Xizmatchi suzidan keyingi operator bajariladi.

Misol IF y>o THEN D:qSQRT (y) ELSE D:q U ;

Agar y > o bulsa DqU , buladi , aks xolda D:q y, buladi.

IF (x y) AND (x> Z) THEN (X:qo) ELSE y:qo;

Ayrim algaritmlarda ba'zan shunday xol uchrase mumkinki, bunda Xisoblash jarayonida ayrim amallar ba'zi bir mantiqiy ifoda bajarilganda xisoblanadi, agar mantiqiy ifoda bajarilmasa xech kanda amal bajarilmaydi. Bu xolda operatorning kiska kurinishda ifodalash mumkin. Uning metaformulasi yozilishi quyidagicha:

< kiska shartli operator> :: q IF <mantiqiy ifoda> THEN < operator > yoki oddiyrok
IF <mantiqiy ifoda > THEN <operator >

Kiska kurinishdagi shartli utish operatorining umumiy kurinishi quyidagicha :

IF B THEN A

Bu erda V mantiqiy ifoda, A operator.

Operatorning bajarilish tartibi quyidagicha :

agar V- mantiqiy ifoda TRUE (rost) qiymat kabul kilsa, A-operator bajariladi, aks xolda keyingi turgan operator bajariladi.

Misol IF X <0 THEN T:qX*X;

Kiska kurinishdagi shartli utish operatorining umumiy kurinishidagi A-operator urnida uz novbatida yana kiska kurinishdagi shartli utish operatori bulishi mumkin va u quyidagi kurinishga ega buladi.

IF B1 THEN IF B2 THEN A11 ;

Bu operator bajarilishi natijasida V1 mantiqiy ifoda tekshiriladi agar TRUE qiymat kabul kilsa V2 mantiqiy ifoda tekshiriladi, u xam rost bulsa (TRUE) A11 operator bajariladi.

Agar mantiqiy ifodalar V1 yoki V2 yolg'on bulsa, (FALSE) shartli utish operatoridan keyingi operator bajariladi. Tulik kurinishdagi shartli utish operatoridagi A2 operator urnida tulik kurinishdagi shartli operator bulishi mumkin.Operatorning umumiy kurinishi quyidagicha :

IF B1 THEN A1 ELSE IF B2 THEN A2 ELSE A3 ;

Paskal tilining koidalariga binoan ELSE xizmatchi suzi xar doim uzidan oldin keladigan eng yakin xizmatchi suziga tegishli buladi . Bu operatorning bajarilishi natijasida V1 mantiqiy ifoda tekshiriladi agar TRUE qiymat kabul kilsa, A1 operator bajariladi, aks xolda V2 mantiqiy ifoda tekshiriladi.V2 mantiqiy ifoda TRUE qiymat kabul kilsa A2 operator bajariladi, aks xolda A3 operator bajariladi.

Misol :

IF X<qA THEN Z:q SIN(x) ELSE IF x>B

THEN Z :q SIN(x) G` sos (x) ELSE Z:qcos (x)

Agar birinchi shart bajarilsa Z:qsine (x) xisoblanadi, aks xolda ik-kinchi shart x> B tekshiriladi va bu shart bajarilsa

Z:qSIN (x)G` COS (x) xisoblanadi, aks xolda Z:q COS (x) xisoblanadi .

Agar kurilayotgan masalada shartni tekshirish operatorida bir necha amallar bajarilishi talab etilsa operator kavs – BEGIN va END lardan foydalanish mumkin.

Misol :

x va u uzgaruvchilarga 5 va 23 qiymatlarini uzlashtirish kerak, agar $A < 0$ bulsa $x \leftarrow 5$, $u \leftarrow 23$ aks xolda x va u o'ni uzlashtirsin, u xolda IF operator-lari quyidagicha yoziladi .

Misol :

```
IF A < 0 THEN BEGIN X:5; Y:23
```

```
ELSE BEGIN X:0; Y:0; end
```

Shunday operatorlari kavs va BEGIN va END lari tashkil etishda xam kullash mumkin. Bu xolda BEGIN va end operatorlarining aniq va maksadga muvofik ishlashini ta'minlaydi.

Misol :

```
IF A OR B THEN
```

```
BEGIN IF X < 0 THEN X: 1 end ELSE
```

```
BEGIN IF X < 0 THEN x: 1 Q0.5 ELSE Y:1 Q0.5 END
```

Juda kup xisoblash jarayonlarida katta tarmoklash uchrashi mumkin. Tarmoklanishning shartli operatori orkali amalga oshirilishni yuqorida kuringiz. Ammo ayrim xollarda tarmoklanishni ikki tarmok buyicha emas, balki k ($k > 2$) yul buyicha olib borish talab kilinadi. Umuman olganda buni bizga tanish shartli operator yordamida amalga oshirish mumkin.

```
IF B1 THEN A1 ELSE
```

```
IF B2 THEN A2 ELSE
```

.....

```
IF BK THEN AK
```

Lekin bu xolda shartli operatorning yozilishi noqulay. Kup xollarda dasturchi uchun shartli operatorning umumiylik kurinishi – tanlash (variant) operatorini ishlatish kulay. Tanlash operatorining metaformulasi quyidagicha yoziladi:

```
<tanish operatori>::CASE<operator selektori> OF <tanlash ruyxati elementi>,<tanlash ruyxati elementi>END
```

bunda

```
<operator selektori>::<ifoda><tanlash ruyxati elementi>::<tanlash belgisi>:<tanlash belgisining ruyxati>:<operator><tanlash belgisining ruyxati>::<tanlash belgisi><tanlash belgisi><tanlash belgisi>::<uzgarmas>.
```

Tanlash operatorining umumiy kurinishini oddiyroq xolda

```
CASE C OF
```

```
M1: A1
```

M2 :A2

.....

Mn : An; end;

Kurinishda yozish mumkin.

Bu erda CASE –xizmatchi (tanlash) suzi , OF (dan) S- selektor Mi - operatorlar belgilari, Ai-operatorlar.

CASE operatori tarmoklanish jarayonini berilgan bir nechta operatoridan birini tanlash yuli bilan amalga oshiradi. Tanlash operatorida barcha operatorlar, shu jumladan bajarilishi uchun tanlangan operator xam anik ravishda keltiriladi (berilgan operatorlar ketma ketligi chegaralangan). Bajarilishi kerak bulgan operator – operatorlar ketma ketligidan operator selektorining kiymatiga kura aniklanadi. Operator selektori sifatida xakikiy bulmagan skalyar kurinishdagi xar kanday ifoda bulishi mumkin. Operatorning ishlashida uning tarkibidagi xar bir operator tanlash belgisi deb ataluvchi belgi taminlanadi. Bu belgi operatorning bajarilishi uchun zarur bulgan selektorning maxsus kiymatini kabul kiladigan selektorning tavsifiga mos konstantadir. Operator bir nechta mavjud kiymatlar bilan ishlashi uchun unda tanlash belgilari ruyxati keltirilishi kerak. Operator bajarilishida dastlab selektorning kiymati xisoblanadi. Sungra selektorning kiymatiga mos belgili operator bajariladi. Agar operatorlar ketma- ketligida bunday belgili operator topilmasa, dasturda xato kayd etiladi. Shuning uchun dastur bajarilishi jaraenida selektorning kiymatiga mos keladigan maxsus belgili operator - operatorlar ketma- ketligida bulishi shart. Bunda tanlash operatorida beriladigan belgilar-belgilarni tafsiflash bulimida keltirilmaydi. Tanlash operatoridagi belgili operatorlar oddiy belgiga xam ega bulishlari mumkin. Bu xolda oldin tanlash belgilari sungra oddiy belgilar eziladi. Shuni xam inobatga olish lozimki tanlash operatoriga fakat SASE xizmatchi suz orkali kirish mumkin, ya'ni tanlash operatoridan tashkaridagi utish operatori orkali bu operatoriga murojaat kilish mumkin emas. Tanlash operatoriningbajarilishi uning tarkibidagi operatorlar ketma-ketligidagi bitta operatorning bajarilishiga olib keladi. Shuning uchun ularning biridan biriga GOTO operatori erdamida utish xato demakdir.

Shartli utish operatorining kuyidagi kurinishi

IF B THEN AI ELSE A2

Tanlash operatorining kuyidagi operatoriga ekvivalentdir;

CASE B OF

TRUE: AI;

FALSE: A2;

END;

Kiska kurinishdagi shartli utish operatorining kuyidagi kuri- nishi IF B THEN A tanlash operatorining kuyidagi kurinishga ekvivalentdir.

CASE B OF

TRUE: A;

FALSE:

END;

Misol:

CASE T OF

‘*’,’G’:R:q1;

‘Q’,’-’:R:q2

END;

Bu operatorning bajarilishi natijasida, agar T belgisi uzgaruvchi Q yoki – belgi qiymatlarini kabul kilsa R uzgaruvchi 2 qiymatni agar T- uzgaruvchi * yoki G` belgini kabul kilsa R uzgaruvchi 1 qiymatni kabul qiladi.

Misol: $Ax^2 + Bx + C = 0$ kvadrat tenglamaning ildizlarini topish dasturi tuzilsin.

PROGRAM corni ; label 20;

Var A,B,C,D,E,F,X,X1,X2,Z:real ;BEGIN

Read (A,B,C);

If A=0 THEN BEGIN

X:=B/C;

WRITELN (X);

GOTO 20

END;

ELSE BEGIN

D:=B*B-4.0*A*C;

Z:=2.0*A;

E:=B/Z;

F:=SQRT(ABS(D))/Z END;

IF D>0 THEN BEGIN

X1:=E+QF;

X2:=E-QF;

WRITELN (X1,X2); END;

ELSE IF D=0 THEN

BEGIN X:=E; WRITELN(X); END ELSE WRITELN ('ECHIMI YUK')

20 : END

Nazorat savollari:

1-Misol. Kuyidagi belgilar ketma ketligi shartli operator buladimi?

- a) if $x < y$ then $x:q0$ else $y:q0$
- b) if $x > y$ then $x:q0$ else read (y)
- v) if $x > qy$ then $x:q0; y:q0$ else write (z)
- g) if $a < b$ then 100 else $z:q5$
- d) if $a < b < c$ then $z:qzQ1$
- e) if $\text{sqrt}(z) < q3.17$ then $z:qzQ1$
- j) if $a < > b$ then $z:qzQ1; a:qbQ1$

2-Misol. Kuyida keltirilgan shartli operatorlarning kaysilari tugri ezilgan?

- a) if $xQy < z$ then $x:qxQ1$ else if $y > z$ then $z:q0$ else $y:q0$,
- b) if $xQy < z$ then if $y > z$ then $z:q0$ else $y:q0$ else $z:q0$,
- v) if $xQy < z$ then if $y > z$ then $z:q0$ else $y:q0$?

3-Misol. Berilgan uchta sondan kichigini topish programmasini ezing.

4-Misol. Uchta son berilgan. Shu sonlardan musbatlarini kvadratga kutaradigan, manfiy sonlarni uzgarishsiz koldiradigan programma tuzing.

Mavzu 2.9-2.10: Takrorlanuvchi jarayonlarni dasturlash.

Takrorlanish operatorlari.

Reja:

- 1.Kirish.
- 2.Takrorlanish jarayonlarining uch xil kurinishlari va Paskalning ularga mos operatorlari.
 - 2.1.Avval sharti tekshiriladigan (old shartli –«xozircha») takrorlanish jarayoni.
 - 2.2.Sharti keyin tekshiriladigan (keti shartli –«gacha») takrorlanish jarayoni.
 - 2.3.Parametrli takrorlanish jarayoni.
3. Murakkab takrorlanish jarayonlari.
- 4.Xulosa.

Таянч сыз ва иборалар:

I.Kirish

Shu paytgacha kurib chikilgan operatorlarning xammasi anik bir marotoba bajariladigan buyruklar tizimini tashkil etadi. Kurib chikilgan operatorlar orkali fakat oddiy xisoblashlarnigina bajarish mumkin. Xakikatdan bunday oddiy kurinishdagi masalalarga dastur tuzib, uni EXM ga kiritib, xatosi bulsa uni tugirlab natija olishga sarflangan vaktни yarmisini sarflangan xolda, bunday masalalarni oddiy kalkulyatorda xam, dastur tuzmasdan xisoblash mumkin.

Amaliyotda murakkab jarayonlarni dasturlashda ma'lum buyruklar ketma-ketligini ma'lum shartlar asosida kayta-kayta bajarish zaruriyati tugiladi. Yukorida utilgan maruzalardan bilamizki, ma'lum bir uzgaruvchining turli kiymatlarida ma'lum buyruklar tizimining biror bir konuniyatiga asosan kayta-kayta bajarilishi takrorlanuvchi xisoblash jarayoni (tsikl) deb ataladi.

Takrorlanuvchi xisoblash jarayonining takror-takror xisoblana-digan kismini takrorlanishning tanasi deb ataladi.Takrorlanish ichida kiymatlari uzgarib boradigan uzgaruvchini takrorlanish uzgaruvchisi yoki takrorlanishning boshkaruvchi uzgaruvchisi (tsikl parametri) deb yuritiladi. Takrorlanuvchi jarayonning algoritmi umumiy xolda kuyidagilarni uz ichiga olishi kerak:

- 1.Takrorlanishni tayyorlash – takrorlanishni boshlashdan oldin, takrorlanishda katnashadigan uzgaruvchilarning boshlangich kiymatlari yoki takrorlanish uzgaruvchisining boshlangich kiymati urnatiladi, takrorlanish uzgaruvchisining uzgarish kadami belgilanadi.
- 2.Takrorlanish tanasi – takrorlanish uzgaruvchilarining turli kiymatlari uchun takror bajariladigan amallar ketma-ketligi kursatiladi.

3. Takrorlanish uzgaruvchisiga yangi qiymat berish – xar bir takrorlanishdan avval uzgaruvchiga uzgarish kadamiga mos ravishda yangi qiymat beriladi.
4. Takrorlanishni boshkarish – takrorlanishni davom ettirish sharti tekshiriladi, takrorlanishning boshiga utish kursatiladi.
2. Takrorlanish operatorining uch xil kurinishlari va Paskalning ularga mos operatorlari.

Paskal algoritmik tilida uch xil kurinishda takrorlanuvchi xisoblash jarayonini tashkil qilish mumkin va bu jarayonlarni dasturlash uchun maxsus operatorlar belgilangan:

- avval sharti tekshiriladigan (sharti oldin kelgan "xozircha") takrorlanish jarayoni, bu maxsus WHILE operatori orkali amalga oshiriladi;

- sharti keyin tekshiriladigan ("...gacha") takrorlanish jarayoni, bu jarayonni maxsus REPEAT operatori orkali amalga oshiriladi;

- parametrli takrorlanish jarayoni, bu jarayon maxsus FOR operatori yordamida amalga oshiriladi.

Yukorida keltirilgan jarayonlar ustida aloxida-aloxida tuxtalib utamiz.

Avval sharti tekshiriladigan (old shartli) takrorlanish jarayoni.

Takrorlanuvchi jarayonining bu kurinishi takrorlanish soni oldin-dan nomalum bulgan xollarda, yani takrorlanishdan chikish malum shartga boglik bulgan xollarda ishlatiladi. Takrorlanishning bu jarayonida takrorlanishdan chikish sharti takrorlanish tanasini bajarishdan oldin tekshiriladi.

Ushbu operatorning umumiy kurinishi kuyidagichadir:

<Sharti avval tekshiriladigan takrorlanish operatori>::qWHILE <mantikiy ifoda >DO
<operatorlar>yoki

WHILE L DO M bu erda

WHILE –xozircha, do-bajarish ma'nosini anglatuvchi xizmatchi suzlar, L-mantikiy ifoda, M-operatorlar yoki operatorlar guruxi u takrorlanish tanasini belgilaydi. Takrorlanish tanasida bitta yoki bir nechta operatorlar guruxi bulishi mumkin. Bunda operatorlar guruxi albatta Begin va End operatorlari orasida yozilishi kerak.

Operatorning bajarilishi kuyidagicha:

L mantikiy ifodaning qiymati xisoblanadi.

Agar L mantikiy ifoda rost qiymatga ega bulsa, M operatori bajariladi va bu operator L mantikiy ifodaning qiymati yolg'on bulgungacha kayta-kayta bajariladi.

Agar L mantikiy ifodaning qiymati birinchi tekshirishdayok yolg'on bulsa M operatori biron marta xam bajarilmaydi va boshkaruv WHILE operatoridan keyingi operatorga uzatiladi. Agar L mantikiy ifoda rost bulib, EXM M operatorni bijarish davomida, biror sababga kura takrorlanishdan chikish talib etilsa, u xolda Paskal algoritmik tilida EXIT operatori orkali amalga oshiriladi.

1-misol. Bizga A xakikiy son berilgan bulsin. Shunday eng kichik butun musbat K sonini topish talab kilinsinki , bu son

$$3k > A$$

shartini bajarsin.

Echish. Masalani dasturini tuzish uchun 3 ifodaning kiymatini saklaydigan kushimcha uzgaruvchi kattalik kirgizishimiz lozim. Agar, biz bu kattalikni U identifikatori bilan belgilasak, u xolda $K=0$ da $u=1$ dan boshlab.bitta kadam bilan uzgarishida 3 formulani (darajaga kutarishni) $U=U*3$ rekkurent formula bilan almashtiramiz. U xolda EXMni takrorlanishdagi xisobdan chikish sharti

$U > A$ buladi.

Yukorida kurib chikilgan operatoridan foydalanib, ushbu misolni echish dasturini tuzamiz.

Program wel :

Var k; integer;y m; real;

Begin

U:=1; k:=1; m:=30;

While y<=m do

begin

y:=y*3;k:=k+1;

writeln ('k=';k,'y=';y)

end;

end.

Natija, M=30 son uchun eng kichik K son 4 ekan.

2-misol : Ushbu ifodaning kiymatini topish dasturi WHILE operatori yordamida tuzilsin:

$y = \dots$; $Xb \leq x \leq x0$, $h=0.1$, $x=q1.2$; $x0=1.8$;

bu erda xv va x0 mos ravishda x kattaligining boshlangich va oxirgi kiymatlari. h uning uzgarish kadami .

program prwel (input,output);

uses printer;

var

x,y,xb,xo,h: real;

begin

writeln('xb=,xoqh');)

read (xb,xo,h);

x:qxb;

while x<=xo do

```

begin
y:=(x*x*x-4*x+1)G'(x+1);
writeln(ist,'yq;Y);
x:=x+h;
end
end.

```

Sharti keyin tekshiriladigan takrorlanish jarayoni

Takrorlanish jarayonining bu kurinishi xam takrorlanish soni oldin-dan malum bulmagan xollarda ishlatiladi. Bunday jarayonda biror buyrukni yoki buyruklar tizimi berilgan shart bajarilgunga kadar takror va takror bajariladi. Bu jarayonni avvalgisidan farki shunda-ki, bunda takrorlanishning tanasi xech bulmaganda bir marta bulsa xam bajariladi, chunki takrorlanishdan chikish sharti tikrorlanishning tanasi bajarilgandan keyin tekshiriladi.

Bunday takrorlanish jarayonini dasturlashda maxsus REPEAT operatoridan foydalaniladi.

Ushbu operatorning umumiy kurinishi kuyidagicha:

<Sharti keyin tekshiriladigan takrorlanish operatori>: :qreperat<operatorlar guruxi>until<mantikiy ifoda >
yoki

REPEAT M UNTIL L; Bu erda

REPEAT-takrorlamok.UNTIL-‘gacha’manosini anglatuvchi xizmatchi suzlar;

M-takrorlanish tanasi, uning tarkibida bitta operator yoki operator-lar guruxi bulishi mumkin.

Operatorlar guruxini Begin va End orasiga olish shart emas.

Operator kuyidagi tartibda bajariladi:

-takrorlanish tanasiga kirgan operatorlar birin-ketin bajariladi. Sung L mantikiy ifodaning kiymati topiladi, yani shart tekshi-riladi. Agar bu shart bajarilsa, L ning kiymati TRUE (rost) bulsa, boshkaruv takrorlashdan tashkariga, UNTIL suzidan keyingi operatorga uzatiladi. Aks xolda takrorlanish davom etadi. Takrorlanish jarayoning bunday kurinishi yukorida keltirilgan takrorlanishdan shu bilan fark kiladiki, bunda takrorlanish tanasi xech bulmaganda bir marta bajariladi.

Misol Oldingi kursatilgan 2-misolga REPEAT operatorini kullab, Paskal algoritmik tilida dastur tuzilsin.

Progpam preper (input.output);

Uses printer;

Var

X,y,xb,xo,h: real ;

Begin

```

Writeln('xbq,xoq,h');
Read(xb,xo,h);
X;qxb;
Repeat
Y:q(x*x*-4*xQ1)G`(xQ1);
Writeln (ist,'yq',Y);
X:qxQh;
Until x>xo;
End.

```

Parametrli takrorlanish jarayoni.

Yukorida keltirilgan takrorlanish operatorlarini odatda takrorlanish soni nomalum bulgan xollarda ishlatish maksadga muvofikdir. Takrorlanish jarayonining takrorlanish soni uni bajarilishidan oldin malum bulsa, xamda takrorlanish uzgaruvchisining kiymatlari orasidagi fark (uzgarish kadami) 1ga yoki -1ga teng bulsa parametrli takrorlanish operatoridan foydalanish kulay.

Bu buyrukning dasturdagi umumiy yozilish konstruktsiyasi kuyidagi kurinishga egadir:

<parametrli takrorlanish operatori >::=

FOR <boshkaruvchi uzgaruvchi kattalikning nomi >:q<boshkaruvchi uzgaruvchi kattalikning boshlangich kiymati >TO (DOWNT0) <boshkaruvchi uzgaruvchi kattalikning oxirgi kiymati >DO <operatorlar guruxi>

Yoki FOR V:=LI TO (DOWNT0) L2 DO M

Bu erda FOR –operatorning nomi; V-takrorlanish uzgaruvchisining nomi; L1, L2-mos ravishda V ning boshlangich va kuyi kiymati ; TO- “gacha” manosini anglatuvchi xizmatchi suz; DOWNT0 –V ning kiymati -1ga uzgarganda TO ning urniga ishlatiladi; DO-bajarilsin degan manoni anglatadi; M-takrorlanish tanasi. Tanada bir nechta operator bulsa, ular Begin va end xizmatchi suzlari orasida yozziladi.

Takrorlanish uzgaruvchisining toifasi REAL va cheklanmagan INTEGER toifasidan tashkari barcha oddiy toifalarda berilishi mumkin va uning kiymatlari orasidagi fark bitta kattalikka teng. L1 va L2 V ning toifasidagi kattaliklar bulib, uning urnida uzgaruvchi nomllarini, ifodalarni va uzgarmaslarni ishlatish mumkin.

Operatorning ishlash tartibi kuyidagicha:

-takrorlanish tanasi takrorlanish uzgaruvchisining xamma kiymatlari uchun (boshlangich kiymatdan kuyi kiymatigacha) kayta-kayta takrorlanadi. Takrorlanish operatorida TO ishlatilganda $V > L2$ sharti, DOWNT0 ishlatilganda $V < L2$ sharti bajarulgungacha davom etadi.

Shuning uchun operatorning birinchi kurinishida $L1 < L2$, operatorning ikkinchi kurinishida $L1 > L2$ bulishi kerak.

-suzli konstruktsiya ishlatiladi. Bu operatorlarni quyidagi kurinishlarda yozish mumkin. Misol.

1-xol for x:q a to b do y:=m uzgaruvchilar

for x:=b downto a do y:=m

2-xol for x:=qa+b to c*k do y:=k ifodalar

for x:=qc*k downto a Qb do y:=m

3-xol for x:=1 to 15 do y:=m uzgaruvchilar

for x:=15 downto 1 do y:=m

2-Misol: yqx ifodani $x=10$ dan boshlab 100 gacha 1 ta kadam bilan uzgartirib, u ifodani qiymatlarini topish dasturini tuzing.

Echish: Bu misol uzgaruvchili takrorlanish jarayoniga mansub bulib, bu erda:

x-boshkarish uzgaruvchisi ;

11-boshkarish uzgaruvchisining boshlangich qiymati 10 ga teng;

12-boshkarish uzgaruvchisining oxirgi qiymati 100 ga teng;

Parametrli takrorlanish jarayonga tegishli for operatorini kullab, masalani echish dasturining bulagini yozamiz :

For x:=1 to 100 do y:=x*x

Kuyida ushbu operatorlardan foydalanib misollar echamiz.

1-Misol: 2-misolni for operatoridan foydalanib “to” yordamchi suzli konstruktsiyasidan foydalanib dastur tuzilsin.

Program prfor (input,output);

Uses printer;

Var

X,y,xh,hx: real ;

n,i :integer;

begin writeln ('xh=,hx=');

read (xh,xo,hx);

n:=trunc((xo-xh)/hx) Q1;

x:=xh;

for i:=1 to n do

begin

y:=q(x*x*x-4*xQ1)G'(xQ1);

writeln (i,'y=',y);

x:=xQh;

end

end.

2-Misol: 2-misolni for operatoridan foydalanib “downto” yordamchi suzli konstruktsiyasidan foydalanib dastur tuzilsin.

Program prfot (input ouput);

Uses printer;

Var

X,y,xb,xo,hx: real ;

N,i:integer;

Begin

Writeln('xb=,xo=,hx=');

Read(xb,xo,hx);

N:qtrunc ((xo-xb)G`hx)Q1 ;

X:=xb;

For i:=n downto 1 do

Begin

Y:q(x*x*x-4*xQ1)G`(xQ1);

Writeln (ist,'y=',y);

X:=xQhx;

End;

End.

Bulardan tashkari, lotin alfaviti tartiblangan bulgani uchun bu malumotlarni sikl parametri kiymatlari sifatida ishlatish mumkin, masalan:

For t:q'a'to 'd' do y:qcQt

Yoki For t:q'd' downto 'a' do y:qcQt Bu erda, t-tsikl parametri;

11-sikl parametrining boshlangich kiymati, a-ga teng;

12-sikl parametrining oxirgi kiymati, d-ga teng;

m-urniga esa y:qcQt ifoda tugri keladi.

Misol –1:

Program prch (input,ouput);

Uses prtnter;

Yar

T,y:char;

Begin

For t:='a'to 'z' do

Begin

Writeln (ist,t,ord (t));

End;

Readln

End.

Misol-2

Program prch (input,output);

Uses printer;

Var

I: integer ;

Begin

For i:=97 to 122 do

Writeln (ist,i,'=',chr (i));

End.

4.Murakkab takrorlanish jarayonlari

Yukorida keltirilgan operatorlar asosida murakkab takrorlanishlar tashkil kilish mumkin. Agar takrorlanish jaryonlarining tanasi takrorlanish strukturasidan tashkil topgan bulsa, u xolda bunday takrorlanish ichma-ich joylashgan yoki murakkab deb ataladi, yani boshkacha kilib aytganda, bitta takrorlanish ichiga bir yoki bir necha boshka takrorlanishlar kirsas murakkab strukturali takrorlanishli dasturlar xosil buladi. Boshka takrorlanish jarayonlarini uz ichiga olgan takrorlanish tashki takrorlanish deb ataladi.

Takrorlanish jarayonlarini ichida joylashgan takrorlanish ichki takrorlanish deb ataladi.

Zqsin $xQcos$ y funksiyaning kiymatini x va

U uzgaruvchilarining $xv < qx < qxo$ va $uv < qyo$ kiymatlarida xisoblash dasturini tuzing.

PREOGRAM FUNG;var

I,j,nx,ny: integer;

X,xo,hx,hn,z:real;

Begin

Read(xo,hx,yo,hy,yn);

Writeln('x0q';xo,'hxq'hx,'xn);

Writeln ('yoq',hy',yn,);

Px:qtrunc((xnqxo)G`hx),Q1

Ny:qtrunc ((yn-yo)G`hy)Q1;

X:qxo;

For i:q1 to nx do

Begin*tashki takrorlanishni boshlash*

Y:qyo;

```

For j:q1 to ny do
Begin*ichki takrorlanishni boshlash*
Z:qsin(x)Qcos(x);
Writeln ('xq',x,'yq'y, 'zq',z);
Y:qyQhy
End; *ichki takrorlanishni tugatish*
X:qxQhx
End; *tashki takrorlanishni tugatish*
End.

```

Nazorat savollari:

Kuyidagi operatorlar ketma - ketligi sikl operatori xisoblanadimi?

- a) while x<0 do x:qxQ0.5
- b) while x< 0 do x:qx-100
- v) while 0< y <1 do y:q sqr (y)Q0.01
- g) while a> 0 do y:q2Qy
- d) while a>b do a:q a-1; b:qbQ1?

2. Kuyidagicha boshlangan sikl operatori tugallanadimi?

while abs (x)Q1>0.793 do ... ?

3. X uzgaruvchining kanday boshlangich kiymatlarida sikl operatori tugallanadi?

- a) while x<1.3 do x:qsqr(x)
- b) while abs(x)>q1 do x:qx-1
- v) while 2*x>x do x:qx-1
- g) while sqr (x)>q0 do x:qsin(x)Q1.315?

4. Kuyidagi operatorlar bajarilgach s uzgaruvchi kanday kiymat kabul kiladi.

- a) s:q0 ; i:q0;
while i<5 do i:=iQ1; s:=sQ1G` i; 0.2
- b) s:q0 ; i:=1;
while i>1 do begin s:q sQ1G` i; i:qi-1 end; 0.0
- v) s:q0 ; i:q1;
repeat s:=s+1G` i; i:=i-1 until i<=1 1.0
- g) s:=1; n:=1;
for i:=2 to n do s:=s+1G` i 1.0

Mavzu 2.11: Malumotlarning murakkab toifasi.Qayd kilingan va chegaralangan toifalar.

Reja:

1. Kirish. Malumotlarning kayd kilingan toifasi. Ularni tavsiflash va ulardan dasturda foydalanish.
- 2.Malumotlarning chegaralangan toifasi. Ularni tavsiflash va ulardan dasturda foydalanish.
- 3.Misollar.
- 4.Xulosa.
- 5.Foydalanilgan adabiyotlar ruyxati.

1.Sagatov M.V.,Yakubov O.X. Informatika (maruzalar matni)

Toshkent- 2000 y.

2.D.B.Polyakov, I.Yu.Kruglov. Programmirovaniye v srede Turbo

Paskal (versiya 5.5)-M.:MAI AG`O «ROSVUZNAUKA», 1992.

Kirish.

Biz shu vaktgacha dasturlarda malumotlarning standart toifalari bilan ishlab keldik. Bu toifalardagi kiymatlar diapazoni Paskal tili tomonidan belgilanadi. Malumotlarning toifasi shu toifadagi kattaliklarning kabul kilishi mumkin bulgan kiymatlar tuplamini belgilaydi. Malumotlarnig toifasini belgilash uz navbatida bu toifadagi kattaliklar ustida bajariladigan amallarni xam belgilaydi. Masalan, xakikiy toifadagi kattaliklar ustida kushish, ayirish, kupaytirish va bulish amallarini bajarish mumkin, lekin Div, Mod amallarini bajarish mumkin emas.

Lekin Paskal algoritmik tili programma tuzuvchisiga yangi, nostandart toifalarini kiritish, yani malumotlarning uziga kulay bulgan kiymatlarini ishlatish imkoniyatini beradi. Bularga odiy toifaga kiruvchi kayd kilingan va chegaralangan toifalar kiradi.

1. Malumotlarning kayd kilingan toifasi. Ularni tavsiflash va ulardan dasturda foydalanish.

Malumotlarning kayd kilingan toifasi. Bu toifadagi kiymatlar cheklangan sonda va katiy tartibda kayd etish ,keltirish yuli bilan beriladi.Bu toifadagi uzgaruvchilar fakat shu toifada keltirilgan kiymatlarning birini kabul kilishi mumkin.Odatda kayd kilingan toifa kiymatlari echilayotgan masalaning mazmuniga kura belgilanadi.

Kayd kilingan toifalarni xar doimgidek, TYPE (toifalarni tavsif-lash) va VAR (uzgaruvchilarni tavsiflash) bulimida tavsiflash mumkin.

TYPE bulimida:

TYPE kayd kilingan toifa nomi(konstantalar ruyxati);

VAR uzgaruvchi nomi: kayd kilingan toifa nomi;

Bu erda konstanta sifatida foydalanuvchi tomonidan beriladigan qiymatlar olinadi va ular ruyxatda vergul orkali keltiriladi.

Masalan:

TYPE BULq(FALSE, TRUE); WEEKq(Sunday, Monday, Tuesday, Wednesday, Friday,saturday);

Animalq(Dog, cat, gow);

VAR B1, B2:BUL; D1:WEEK; D2:ANIMAL;

Bu tavsifga asosan operatorlar bulimida kuyidagi operatorlarni ishlatish mumkin:

D1:q Monday; B1:qTRUE; D2:qCAT;

Kuyidagi operatorlarni esa ishlatib bulmaydi:

B1:qDOG; D1:qFALSE; D2:qSunday;

Chunki, takidlaganimizdek, xar bir uzgaruvchi fakat uzi aniklangan toifa qiymatlarining birini kabul kilishi mumkin edi.

VAR bulimida:

VAR uzgaruvchi nomi: (konstantalar ruyxati);

Bu erda konstantalar ruyxatida kayd kilingan toifa qiymatlari vergul orkali beriladi.

Masalan:

VAR DAY, D3:(Monday, Tuesday,WEDN,Friday); SEZ,D4: (SUMMY,AUTUMN);

Kayd kilngan toifada konstantalar tartiblangan, yani xar bir konstanta uzining tartib rakamiga ega va u 0 dan boshlanadi. Yukorida keltirilgan misollardagi WEEK toifasida konstantalar mos ravishda kuyidagi tartib rakamiga ega:

Sunday-0; Monday-1;Tuesday-2 va x.k. Konstantalarning tartiblanishi uz navbatida bu konstantalar ustida malum amallarni bajarish imkonini beradi va ular uchun bazi bir standart funktsiyalar belgilangan:

1). PRED(Z)- bu erda Z kayd kilingan toifadagi uzgaruvchi yoki shu kayd kilingan toifadagi konstanta. Bu funktsiya z dan bitta oldingi konstantani aniklab beradi.

Masalan:

DAY:qTuesday; D3:qPRED(DAY);

Operatorlari bajarilishi natijasida D3 uzgaruvchisi MONDAY

Kiymatini uzlashtiradi.

2). SUCC(Z) – bu funktsiya Z dan bitta keyingi konstantani aniklab beradi.

Masalan:

SEZ:qSUMMY; D4:qSUCC(SEZ);

Operatorlarning bajarilishi natijasida D4 uzgaruvchisi AUTUMN

Kiymatiga ega buladi.

3). ORD(Z)- bu funktsiya Z ga mos kelgan kiymatning tartib rakamini aniklab beradi.

Masalan: SEZ:qSUMMY; D:qORD(SEZ); operatorlarining bajarilishi natijasida D ning kiymati 0 ga teng buladi (D:INTEGER).

Bulardan tashkari, kayd kilingan toifa kiymatlari ustida solishtirish amallarini xam bajarish mumkin. Masalan, Monday<Tuesday mantiqiy ifodaning kiymati TRUE ga teng, chunki Monday-0 tartib rakamiga, Tuesday-1 tartib rakamiga ega.

Misol:

```
PROGRAM RR;
```

```
TYPE DAYq(MON,TUS,WED,THUR,FRI,SAT,SUN);
```

```
VAR WEEKDAY: DAY;TEL: INTEGER;
```

```
BEGIN WEEKDAY:qWED; WRITELN ('1 TIMEq',WEEKDAY);
```

```
WEEKDAY:qSUCC(WEEKDAY); WRITELN ('2 TIMEq',WEEKDAY);
```

```
WEEKDAY:qWED;
```

```
WEEKDAY:qPRED(WEEKDAY); WRITELN ('3 TIMEq',WEEKDAY);
```

```
TEL:q ORD(WED); ERITELN('ORD-NUMBERq',TEL)
```

```
END.
```

Dasturning bajarilishi natijasida kuyidagi kurinishda natija olinadi:

1 TIMEqWED

2 TIMEqThursday

3 TIMEqTUS

ORD-NUMBERq2

Yukorida keltirilgan misollardan kurinib turibdiki, kayd kilingan toifa kiymatlarini READ operatori yordamida xotiraga kiritib va WRITE operatori yordamida bosmaga chikarib bulmaydi. Kayd kilingan toifadagi uzgaruvchilarga kiymatlar uzlashtirish operatori yordamida uzlashtiriladi.

Malumotlarning chegaralangan toifasi.

Bazi bir xollarda uzgaruvchi uzi aniklangan toifaning xamma kiymatlarini emas, balki malum chegaradagi kiymatlarnigina kabul kilishi mumkin. Bu xollarda uzgaruvchi uchun chegaralangan toifa belgilash mumkin. Chegaralangan toifa oldindan aniklangan toifaga cheklanish kuyish bilan beriladi. Oldindan aniklangan toifa chegaralangan toifaga nisbatan baza toifasi vazifasini bajaradi. Baza toifasi sifatida REAL toifasidan tashkari barcha standart toifalarni ishlatish mumkin. Chegaralangan toifa kuyidagi kurinishda tavsiflanadi:

```
TYPE <chegaralangan toifa nomi>q1-konstanta..2-konstanta;
```

```
VAR <uzgaruvchi>:<chegaralangan toifa nomi>;
```

Bu erda 1-konstanta va 2-konstanta mos ravishda baza toifasining birinchi va oxirgi qiymatlari. Chegaralangan toifadagi uzgaruvchi baza toifasining birinchi qiymati va oxirgi qiymatlari orasidagi ixtiyoriy qiymatlarning birini kabul qilishi mumkin.

Masalan:

```
TYPY CHISq1..100;
```

```
SIMq'A'..'F';
```

```
FASLq(KUZ,QISH,BAHOR,YOZ);
```

```
SEZqKUZ..BAHOR;
```

```
BULqFALSE..TRUE;
```

```
VAR C:CHIS;F:FASL;S:SEZ;B:BUL;
```

Bu misolda CHIS toifasi uchun INTEGER toifasi, SIM toifasi uchun CHAR toifasi, SEZ toifasi uchun FASL toifasi, BUL toifasi uchun BOOLEAN toifasi baza toifasi vazifasini bajaradi.

Chegaralangan toifani aniklashda quyidagi koidalarga amal qilish kerak:

Chegaralangan toifaning birinchi va ikkinchi konstantalari bir xil toifada bulishi kerak.

Birinchi konstanta qiymati ikkinchi konstanta qiymatidan kichik bulishi lozim.

Masala. Guruxdagi 5 ta talaba ichida alochi talabani aniklash dasturi tuzilsin.

```
PROGRAM RR;
```

```
TYPE
```

```
FAM=(AHMEDOV, SOBIROV, VALIEVA, SODIQOV, RAVILOV);
```

```
FAN=(MAT,HIM,FIZ,ING);
```

```
BAHO=1..5;
```

```
VAR STUD,I:FAM;J:FAN; B:BAHO;S:REAL;
```

```
BEGIN WRITELN('TALABANING BAXOSINI KIRITING');
```

```
FOR I:=AHMEDOV TO RAVILOV DO
```

```
BEGIN S:0;
```

```
FOR J:=MAT TO ING DO
```

```
BEGIN READ(B); S:=SQB END;
```

```
READLN;
```

```
IF SG`4>=5 THEN BEGIN WRITELN('Alochi talaba ');
```

```
CASE STUD OF
```

```
AHMEDOV: WRITELN ('Axmedov');
```

```
SOBIROV : WRITELN ('Cobirov');
```

```
VALIEVA: WRITELN ('Valieva');
```

```
SODIQOV: WRITELN ('Sodikov');
```

```
RAVILOV: WRITELN ('Ravilov');
```

```
END
```

END;

END

END.

Bu dasturda FAM- talabalarning familiyalari keltirilgan toifa, FAN-fanlar nomi keltirilgan toifa va BAHO-baxolar mezoni kursatilgan toifalar yaratilgan. Dasturni bajarish jarayonida xar bir talabaning xar bir fandan olgan baxosi 5 ballik tizimda aloxida katorda kiritiladi. Natijada ekranga

Alochi talaba

Yozuvidan keyin talabaning familiyasi chikariladi.

Mavzu 17-18: Massivlar. Malumotlarning murakkab toifasi.

Muntazam toifa.

Reja :

Kirish.

- 1.Massiv tushunchasi va indeksli uzgaruvchilar.
2. Bir va ko'p ulchamli massivlar .
- 3.EXM xotirasiga massiv elementlarni kiritish va chikarishni tashkil kilish.
- 4.Massiv elementlari ishtirokida masalalarni echish.

Foydalanilgan adabiyotlar ruyxati.

- 1.N.Kultin *Samouchitel Programmirovaniye v Turbo Pascal 7.0 i Delphi. Moskva Sank-Peterburg. 1999 g.*
- 2.V.G.Abramov i dr. *Vvedenie v yazk Paskal.-M.:Nauka,1988.*

Kirish.

Yukorida takidlab utilganidek Paskal tilida ishlatiladigan malumotlar 2 turga, oddiy (REAL, INTEGER, BOOLEAN, CHAR, kayd kilingan va chegaralangan) va murakkab turdagi malumotlarga (muntazam toifadagi malumotlar, yozuvlar, faylli turlar, tuplamlar va kursat-kichlar) bulinadi. Oddiy turdagi malumotlarning boshka turdagi malumotlardan asosiy farkanuvchi belgisi, ularning tartiblangan- ligi va yaxlitligidadir, yani masalan INTEGER turiga mansub ixtiyoriy kattalik aloxida rakamlarga bulinmaydigan yaxlit kattalikdir (INTEGER toifasidagi kattalik sifatida bitta butun son tushuniladi).

- 1.Massiv tushunchasi va indeksli uzgaruvchilar

Sonlar ukiining ixtiyoriy bulagini butun sonlar ketma-ketligi deb karasak, u xolda shu ketma-ketlik 1-nchi rakami tugrisida gapirish, bu elementlar (butun sonlar) tuplamiga «butun son» degan umumlashtiruvchi nom berish mumkin. Shunday umumiy nomga ega, tartiblangan kattaliklar tuplami muntazam toifadagi malumotlar yoki massivlar deb yuriti-ladi. Ularning tashkil etuvchilari massiv tashkil etuvchilari yoki elementlari toifasi esa bazaviy toifa deb nomlanadi. Masalan, kuyidagi tartiblangan xakikiy sonlar ketma-ketligi :1.5,14,7 –5.0, 8.5, 0.45 xakikiy sonlar massivi, bu massivning bazaviy toifasi esa xakikiy (REAL)toifadir.

Massiv tushunchasi A –umumiy nomga ega va bir toifadagi kuzgalmas kattaliklar ketma-ketligidan iborat bulgan sonli vektor

$A(5) \text{ q}(a_1, a_2, a_3, a_4, a_5)$

misolida tushuntirish mumkin. Bu erda a_i -massiv elementlaridir. Ularni ifodalashda indeksli uzgaruvchilardan foydalaniladi.

Matematika kursidan malumki, indeks uzgaruvchilarning tartiblangan ketma-ketlikdagi urnini bildirib, kavslar ichiga olinib yoki massiv nomidan bir muncha pastda kursatilar edi, masalan $A(1)$ yoki A , umumiy xolda esa A , bu erda $i \in 1, 2, 3, \dots, n$

Paskal tilida uzgaruvchining indeksi turtburchak kavslar ichiga olinib yoziladi, yani $A[1]$ q1.5, $A[2]$ q14.7, $A[3]$ q-5.0 $A[4]$ q8.5, $A[5]$ q0.45,

Agar dasturda massiv ishlatilayotgan bulsa, u xolda uni uzgartiruvchilar bulimi VAR yoki toifalar bulimi TYPE da tavsiflash zarur, masalan VAR bulimida massiv kuyidagi kurinishda tavsiflanadi:

VAR <massiv nomi>: ARRAY [<indeks toifasi >]

OF <komponent toifasi >.

Bu erda <massiv nomi> - ixtiyoriy identifikator; ARRAY (massiv) va OF (-dan) – xizmatchi suzlar; <indeks toifasi >-indeksli ifoda, bu toifa kiymatlari massiv elementlarining sonini belgilaydi, shuning uchun bu toifa sifatida REAL va cheklanmagan INTEGER toifasidan tashkari barcha oddiy toifalarni ishlatish mumkin, <element toifasi> - massiv elementlarining toifasi bulib, bu toifa sifatida fayl va tuplam toifasidan boshka barcha toifalarni ishlatish mumkin .

Yukorida kurib utilgan A vektorini massiv kurinishida dasturda quyidagicha tavsivlash mumkin;

VAR A :ARRAY [1..5] OF REAL;

Indeks sifatida fakat anik son emas, kiymati massiv elementining tartib rakamini belgilovchi ifoda xam ishlatish mumkin, masalan $A[iQ5]$, $B[i \text{ div } (iQ6)]$, $C[n1 \text{ or } n2]$, YEAR [1998] va x.k.

Indeksli ifoda kiymatining toifasi massiv elementi indeksining toifasi deyiladi va bu toifa kiymat-larining tuplami, massiv elementlarining sonini va ularning tartiblanganligini

aniklovchi tuplam bulishi kerak. Kuyida indeks toifasi sifatida Paskalda ishlatilishi mumkin bulgan toifalar bilan tanishamiz.

Ma'lumki Paskal tilida INTEGER va REAL toifasidagi ma'lumotlar tuplami cheklanmagan, REAL toifa esa shuningdek tartiblanmagandir xam. Shu sababli indeks toifa sifatida INTEGER va REAL toifasiga mansub ma'lumotlardan foydalanish mumkin emas, ya'ni
VAR

A:ARRAY [5] OF REAL;

A:ARRAY [INTEGER] OF REAL

B:ARRAY [1.20..1.45] OF INTEGER

Kup xollarda indeks toifasi, sifatida chkgaralangan toifa kupincha - chegaralangan butun toifa malumotlari ishlatiladi. Masalan, 100 ta xakikiy elementlardan iborat massiv kuyidagi kurinishda tavsiflanishi mumkin, VAR A:ARRAY[1..100] OF REAL. Bu erda 1, 100 chegaralangan butun toifa A massiv elementlari soni yuzta ekanligi va ularning tartiblangan- ligini [birinchidan yuzinchigacha] bildiradi.

Tugri tavsiflangan massivlarga misollar:

VAR

MASSIV:ARRAY[-745..-1] OF REAL;

VEAR:ARRAY[1477..1988] OF CHAR;

L:ARRAY[BOOLEAN] OF CHAR;

A, V, S: ARRAY[1..50] OF REAL;

Ma'lumki, kayd kilingan toifa kiymatlarining tuplami xam cheklangan tuplamni tashkil kiladi, bu esa navbatida kayd kilingan toifa kiymatlarini xam indeks toifasi sifatida kullash imkoniyatini beradi, masalan:

VAR

MONTH: ARRAY[mart, aprel, may] OF CHAR;

SOLOR: ARRAY [red, dlue, dlack] OF ITEGER; Indeks va ieks toifasi tushunchalari urtasida uzaro fark mavjud bulib, indeks toifasi massiv elementlari soni va ularning tartiblanganligini bildiradi va u massivni tavsiflash bulimida ishlatiladi, indeks esa massiv elementining tartib rakamini belgilaydi va operatorlar bulimi-dagina ishlatiladi. Agar biror bir massivga murojat kilish uchun uning tulik nomi, ya'ni muntazam toifaga mansub uzgaruvchining nomi ishlatilsa, massivning aloxida elementiga murojat kilish uchun indeksli uzgaruvchi ishlatiladi.

Masalan, A[5] massiv uchun A uzgaruvchi tulik uzgaruvchi (massiv nomi) A[1] indeksli uzgaruvchidir.

Ifodalarda indeksli uzgaruvchidir kiymat berish operatorining chap tarafida xam, ung tarafida xam ishtirok etishi mumkin va ular ustida solishtirish amallarini tartiblash, arifmetik amallar, eng kichik va eng katta kiymatni topish amallarini, ya'ni uning bazaviy toifa ustida bajarilishi

mumkin bulgan barcha amallarni bajarish mumkin. Masalan, agar bazaviy toifa INTEGER bulsa u xolda butun toifa ustida bajarilishi mumkin bulgan barcha amallar, xatto standart funktsiyalarni xam kullash mumkin.

Massiv elementlarini operatorlar bulimida ishlatilishi ga misollar:

```
V[5]:=B[3]Q1
```

```
SUM:=SUM ROUND(C[K]);
```

```
P1:=SORT(A[ 2*IQ1]);
```

Paskal' tilida massivlarni uzgaruvchilar bulimida tavsif-lashdan tashkari TYPE toifalar bulimida xam tavsiflash mumkin. Buning uchun TYPE bulimida massiv toifasi nomi va massiv toifasi beriladi, VAR bulimida ushbu toifaga mansub uzgaruvchilar sanab utiladi.

TYPE bulimida toifalarni tavsiflash dasturlashda yaxshi uslub sanaladi va dasturni mantiqiy mukammalligini oshirish imkoniyatini beradi. Massivni TYPE bulimida tavsiflash quyidagi kurinishga ega buladi:

```
TYPE <toifa nomi>q ARRAY[<indeks toifasi>] OF<element toifasi>;
```

```
VAR <uzgaruvchi yoki massiv nomi>:<toifa nomi>;
```

Masalan:

```
TYPE
```

```
MAS= ARRAY [1...5] OF REAL;
```

```
VAR A: MAS;
```

Agar dasturda bir emas, bir necha massiv MAS toifasiga mansub bulsa, u xolda xar bir massivni aloxida tavsivlamas- dan, uzgaruvchilar bulimida ularni vergul orkali keltirish mumkin, ya'ni

```
VAR A,B,C,D: MAS;
```

Shu vaktgacha biz elementlari fakat bitta indeksli massivlarni, ya'ni bir ulchamli massivlarni kurib chikayotgan edik. Paskal dasturlash tilining massiv elementlari toifaga ularning xammasi bir toifaga mansub bulishi kerakligidan boshka xech kandy cheklanishlar kuyilmasligi massiv elementlari sifatida massivlar xam ishtirok etish imkoniyatini beradi. Bunday massivlar kup ulchamli massivlarni tashkil kiladi. Agar massiv elementlari uz navbatida massivdan iborat bulsa, u xolda ikki ulchamli massiv (matritsa), agar matritsa elementlari massiv bulsa, u xolda biz uch ulchovli massivni olamiz va x.k.

Massivlarning ulchami ularni EXM da kayta ishlashga xech kandy tuskinlik kilmaydi, chunki massiv elementlari ularni ulchamidan kat'iy nazar EXM xotirasida chizikli ketma-ketlik sifatida saklanadi, ya'ni.

Bir va kup ulchamli massivlar va ularni tavsiflash.

TYPE bulimida toifalarni tavsiflash dasturlashda yaxshi uslub sanaladi va dasturni mantiqiy mukammalligini oshirish imkoniyatini beradi. Massivni TYPE bulimida tavsiflash quyidagi kurinishga ega buladi:

TYPE <toifa nomi>= ARRAY[<indeks toifasi>] OF<element toifasi>;

VAR <uzgaruvchi yoki massiv nomi>:<toifa nomi>;

Masalan:

TYPE

MAS= ARRAY [1...5] OF REAL;

VAR A: MAS;

Agar dasturda bir emas, bir necha massiv MAS toifasiga mansub bulsa, u xolda xar bir massivni aloxida tavsivlamas- dan, uzgaruvchilar bulimida ularni vergul orkali keltirish mumkin, ya'ni

VAR A,B,C,D: MAS;

Kup ulchamli massivlar dasturda quyidagicha tavsiflanadi:

ARRAY[<indeks toifasi>] OF ARRAY[<indeks toifa si>] OF<element toifasi>;

Yoki

ARRAY[<indeks toifasi, indeks toifasi>] OF <ele- ment toifasi>;

Masalan, ikki ulchovli massivlarga misol kilib matritsalarini keltirish muin. Ikki ulchovli massivni e'lon kilishning bir necha usullari mavjud, masalan massivni elementlari xam massivdan iborat massiv sifatida e'lon kilish mumkin,

TYPE

MAS= ARRAY [1...5] OF REAL;

MATR=ARRAY [1..3] OF MAS

VAR A: MAS;

B: MATR;

Bu erda A uzgaruvchi 5 ta xakikiy toifaga mansub elementdan iborat bir ulchovli massiv, V uzgaruvchi 3 ta satr va 5ta ustundan iborat (3*5)ikki ulchamli massiv sifatida tavsiflangan.

V massiv tavsifini MAS toifasini kursatmasdan birmuncha soddalashtirish mumkin,

TYPE

MATR= ARRAY [1...3] OF ARRAY [1...5] OF REAL;

VAR

B: MATR ; yoki

TYPE

MATR= ARRAY [1..3,1..5] OF REAL; VAR

B: MATR;

Agar kursatilgan toifa dasturda bitta massivni aniklash uchun ishlatilayotgan bulsa, massivni uzgaruvchilar bulimida e'lon kilish maksadga muvofik buladi.

VAR

V: ARRAY [1..3,1..5] OF REAL;

V matritsaning I-nchi satr va J-nchi ustuni kesishmasida turgan elementiga murojaat V[I, J] yoki B[I],[J] kurinishga ega buladi, Yukoridagi tavsifga asosan kuyidagi operatorlarni ishlatish mumkin:

A[I]:=2.5E03; B[I,JQ1]:=A[K]*B[K,J]; READLN(B[I,J]);

Dasturda massivlardan foydalanish uchun massiv elementlariga kiematlari xotiraga kiritilgan bulishi zarur. Massiv elementlariga kiymat berishda ma'lumotlarni kiritish va uzlashtirish operatorlaridan xam foydalanish mumkin.

Berilgan massiv elementlarini EXM xotirasiga kiritish va chikarish

Kuyida keltirilgan dastur lavxasida bir ulchovli A va ikki ulchovli V massivlar elementlarini kiritish va chikarish amalga oshirilgan.

BEGIN

A massiv elementlarini kiritish

READ (V[1],V[2],V[3]);

B massiv elementlarini kiritish

READLN;

FOR I:=1 TO 2 DO

FOR J:=1 TO 3 DO

READ (A[I,J]);

*Amassiv elementlarini chikarish *

WRITELN ('MASSIV V');

WRITELN ;

FOR I:=1 TO 3 DO

WRITELN (A[I]:5);

WRITELN;

B massiv elementlarini chikarish

WRITELN ('MASSIV V');

FOR I:=1 TO 2 DO

BEGIN

WRITELN ;

FOR J:=1 TO 3 DO

WRITE (B[I,J]:4)

END

END.

V massiv 3 ta elementdan iborat, shuning uchun kiritish operatorida uning barcha tashkil etuvchilari sanab utilgan. Bu massiv elementlarini chikarish uchun I indeks buyicha sikil tashkil kilinib bosmaga «MASSIV A» sarlovxasini chikarish yuli bilan amalga oshirilgan. Ikki ulchamli V massivi elementlarini kiritish (va chikarish) uchun matritsani satrlab kirituvchi ichma-ich joylashgan sikllar ishlatilgan. Kiritish jarayonida massiv elemenlari kiymati bir-biridan probel bilan ajratilgan bulishi mumkin. Berilganlarni barchasi kiritilgandan sung <Enter> klavishasini bosish zarur. Kurib utilgan misolda <Enter> klavishasini A massiv elementlarini xammasi kiritilgandan sung bosish kerak, chunki keyingi berilganlar READLN operatoridan sung kiritiladi. Dasturda bu amallar kuyidagi kurinishga ega buladi.

1 2 3 <Enter>

4, 5, 6, 7, 8, 9 <Enter>

Chikariladigan ma'lumotlar mos ravishda kuyidagi kurinishga ega buladi:

MASSIV V

____ 1 ____ 2 ____ 3

MASSIV MAS

___ 4 ___ 5 ___ 6

___ 7 ___ 8 ___ 9

Kuyida massiv elementlari ustida kupincha ishlatiladigan amallar ishtirokida misollarni kurib chikamiz.

Misol 1. Xakikiy N ta elementlardan iborat A massiv berilgan. Massiv elementlarini usib borish tartibida joylashtiring.

Ushbu masalani echish uchun massiv elementlarini IF operatori yordamida ketma-ket solishtirish va uzlashtirish operatori yordamida elementlarni urnini uzgartirib, tartiblangan massiv ruyxatini tuzish kerak buladi. IF operatorida shart tekshirilib, uning bajarilishi X kushimcha uzgaruvchi yordamida elementlarni urnini uzgarishiga olib kelsa, shartning bajarilmasligi keyingi kadamga utish imkoniyatini beradi. Paskal tilidagi dastur kuyidagi kurinishga ega:

PROGRAM MASSIVNI SARALASH;

VAR

Aq ARRAY [1..100] OF REAL;

I, K, N : INTEGER;

X : REAL;

BEGIN

READLN (N);

FOR I:=1 TO N DO

```

READ (A[I]);
FOR I:=1 TO N-1 DO
FOR K:=N DOWNT0 I DO
IF A[K] >A[KQ1]
THEN BEGIN
X:=A[K];
A[K]:=A[KQ1];
A[KQ1]:=X
END;
FOR I:=1 TO N DO
WRITELN (A[I]);
END.

```

Kushimcha kiritilgan I, J, N butun uzgaruvchilarni aloxida takidlab, I- massivning joriy uzgaruvchisining indekisi, K- ichki sikildagi uzgaruvchi indekisi, N – massiv elementlarining umumiy soni.

Misol 2. $A(2 \times 3)$ va $V(3 \times 3)$ matritsa elementlarini uzaro kupaytmasini topish dasturini tuzing.

Natijaviy S matritsa elementlari kuyidagi formula yordamida topiladi,
 $S_{qab} = \sum_{j=1}^n a_{jq1} v_{j2}$. Bu erda n – A matritsaning satrlar soni, m – A matritsaning ustunlar soni va V matritsaning satrlar soni ; r –V matritsaning satrlar soni. Umumiy xolda natijaviy matritsa S n –ta satr va r – ta ustundan iborat buladi.

Boshlangich kiymatlarni kiritish va chikarish ichma-ich joylashgan sikl yordamida amalga ohiriladi. Natijaviy matritsaning elementlarini xisolash xam ichma-ich joylashgan siklning ichki k parametrli siklida xisoblab topiladi.

Kurib utgan misolni echish dasturi kuyidagi kurinishga ega:

```

PROGRAM KUPAYTMA;
*MATRITsALARNI KUPAYTIRISH *
VAR
A, C: ARRAY [1..2,1..3] OF REAL;
B : ARRAY [1..3,1..3] OF REAL;
I, J, K, : INTEGER;
BEGIN
*MASSIVNI KIRITISH *
FOR I:=1 TO 2 DO
FOR J:=1 TO 3 DO
READ (A[I,J])
*B MASSIVNI KIRITISH *
FOR J:=1 TO 3 DO
BEGIN
C[I,J]:=0
FOR K:=1 TO 3 DO
C[I,J]:=C[I,J]+A[I,K]*B[K,J]
END;
*C MASSIVINI BOSSMAGA ChIKARISH *
WRITELN (MASSIV S);
FOR I:=1 TO 2 DO
BEGIN
WRITELN;
FOR J:=1 TO 3 DO
WRITE (C[I,J]:12:5);
END
END.

```

```

READLN;
FOR I:=1 TO 3 DO
FOR J:=1 TO 3 DO
READ (B[I,J]);
*MATRITsALARNI KUPAYTIRISH *
FOR I:=1 TO 2 DO

```

Nazorat savollari:

1. Pifagor jadvalini shakillantiring va bosmaga chikaring.
2. Pifagor jadvali – bu 10x10 ulchamli kvadrat matritsa bulib elementlar kuyidagi formula orkali topiladi:

PIFAGOR [I,J] := I*J.

Foydalanilgan adabiyotlar

.Foydalanilgan adabiyotlar ruyxati.

1. .N.Kultin Samouchitel Programmirovaniye v Turbo Pascal 7.0
2. i Delphi. Moskva Sank-Peterburg. 1999 g.
3. V.G.Abramov i dr. Vvedenie v yazk Paskal.-M.:Nauka,1988.
4. Sagatov M.V.,Yakubov O.X. Informatika (maruzalar matni)
5. Toshkent- 2000 y.
6. D.B.Polyakov, I.Yu.Kruglov. Programmirovaniye v srede Turbo
7. Paskal (versiya 5.5)-M.:MAI AG`O «ROSVUZNAUKA», 1992.
- G'ulomov C.C. Iqtisodiy informatika. Toshkent. 1999 y*
- Gulomov C.C. Axborot tizimlari va texnologiyalari. Toshkent. 2000 y*
- M.Aripov, A.Xaydarov, N. Muhitdinova. Algoritm asoslari va algoritmik*
- 8. tillar. Toshkent. 2000 y.*
- B.G. Abramov , N.P. Trifanov. Vvedenie v yaz'k Paskal. Moskva. 2002 y*
- 9. 5.WWW. diyalektika. Com*
- 10. 6. http:Ffiit metodist.ru 7. http:// www. Form. Norod.ru*