

## 5-MAVZU

### **Avtomatlashtirishning MP larning programmalash tillari**

1. «Mashina tillari», «Assembler tili», «Yuqori darajadagi programmalash tillari». Mnemokodlar to'g'risida tushuncha, «Assembler» tilining maydonlari, tuzilishi (strukturasi).
2. MP uchun maxsus programmalashtiriladigan vositalar. Redaktorlovchi, yuklovchi, modellashtiruvchi, sozlovchi programmalar.
3. «Kompilyator» va «Assembler» programmali, operatsion sistemalar va monitor to'g'risida tushuncha.

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

Programmani yaratish jarayonidagi uning nomlari Beta versiya - bu versiya programmani ommaga havola qilinib, ularning fikri bo'yicha programmaga turli o'zgartirishlar kiritiluvchi versiyasi. Programmaning bu versiyasi, odatda, tekin bo'lib, ko'pchilik hukmiga havola etiladi. Programmaning bu versiyasi orqali sizga yetkazilgan ziyon qoplanmaydi(fayllaringizning o'chib ketishi, OS ning buzilishi va h.k.). Hozirda ko'pchilik firmalar o'z mahsulotlarini Beta versiyasini chiqarib, o'z mahsulotlarini takomillashtirib bormoqdalar. Bundan, programma ishlab chiqarish bo'yicha yetakchi bo'lgan Microsoft korporatsiyasi ham mustasno

emas(Windows Vista, Office 2007, Exchange Server 2007, Internet Explorer 7 va h.k.).

Release Candidate(versiya nomzodi) - bu versiyaning nomidan ma'lumki, u haqiqiy, sotuvga chiqariladigan versiyaga kandidat(nomzod)dir. Bu kabi versiyalar qisqacha RC deb ataladi. Shu turdagi versiyalar esa, RC1, RC2 kabi nomlanadi. Ko'pchilik RC versiyalar sotuvga chiqariladi, chunki bu versiya o'zida ma'lum bir imkoniyatlarni jamlagan bo'ladi va bu versiya orqali yetkazilgan zarar programmani yaratgan firma tomonidan qoplanadi. Final Release(so'nggi versiya) - programmani yaratishda qo'yilgan maqsadni "to'liq" amalga oshiruvchi versiya. Bu versiyani "alfa" versiya deb ham atashadi. Bu versiya o'z bahosida sotiladi, unga ko'rsatiluvchi xizmatlarning barchasi programma yaratuvchisi tomonidan ko'rsatiladi. Build XXXX(XXXX - qurish) - bu, odatda, versiya hisoblanmaydi. X ning o'rnida biror son keladi. Bu son programmaning kompilyatsiya(programmalash tilidan haqiqiy bajariluvchi kodga o'tkazish) sonini bildiradi. Programma yadrosi (qo'shimchalarsiz, asosiy qism)ga nisbatan yuritiladi. Masalan, Windows Vista Beta 2 Build 5308 - Windows Vistaning Beta 2 versiyasi turkumida 5308-kompilyatsiya(OSga nisbatan "kompilyatsiya"ni ishlatish noo'rinroq:)). Update(yangilash) - programmaning biror teshigini(biror kichik xato, kamchiligi) tuzatuvchi kichik yordamchi programma.

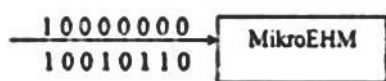
Bu yordamchi programmaning hajmi, odatda, kichik bo'lib, faqat o'sha kamchilikni tuzatishgagina xizmat qiladi. Ya'ni bu programmacha biror \*.dll faylni yoki programmaishlatuvchi funksiyalardan birortasini "tuzatib" qo'yadi, xolos. Bunday Update("qarz")lar har doim tekin bo'ladi. Service Pack(xizmat paketi) - o'zida bir qancha Update lar bajaruvchi amallarni saqlovchi paket. Uning qisqacha nomi SP. Programmaning haqiqiy versiyasi yoki avvalgi SP dan shu paytgacha bo'lgan xatoliklarni birdaniga tuzatuvchi programma. Ya'ni, u bir qancha Update lar ishini o'zi bajaradi.

Dasturlash tillari, ularning turlari va asoslari (dasturlash tili) Kompyuterda dasturlash bu – kompyuter mikroprotsessori uchun turli buyruqlar berish, qachon, qayerda nimani o'zgartirish va nimalarni kiritish yoki chiqarish haqida buyruqlar

berishdir. Ushbu maqolada, qanday dasturlash tillari borligi, eng keng tarqalgan dasturlash tillari va ularning farqi. Hamda, Dasturlashni o'rganish yo'llari haqida suhbatlashamiz Kompyuter dunyosida ko'plab dasturlash tillari mavjud bo'lib, dasturlash va unga qiziquvchilar soni ortib bormoqda.

Bir xil turdagi ishni bajaradigan dasturlarni Basic, Pascal, Ci va boshqa tillarda yozish mumkin. Pascal, Fortran va Kobol tillari universal tillar hisoblanadi, Ci va Assembler tillari mashin tiliga ancha yaqin tillar bo'lib, quyi yoki o'rta darajali tillardir. Algoritmik til inson tillariga qanchalik yaqin bo'lsa, u tilga yuqori darajali til deyiladi. Mashina tili esa eng pastki darajali tildir. Mashina tili bu sonlardan iboratdir, Masalan: 010110100010101 Dasturlash tillari 2 ta katta guruhlarga bo'linadi, Quyi va Yuqori darajali dasturlash tili. Quyi darajali dasturlash tili ancha murakkab bo'lib ular juda maxsus sohalarda ishlatiladi va ularning mutaxassisleri ham juda kam. Chunki quyi dasturlash tillari (masalan: assembler) ko'pincha mikroprosessorlar bilan ishlashda kerak bo'lishi mumkin. Odatda turli dasturlash ishlari uchun yuqori darajali dasturlash tilidan keng foydalaniladi. EHM (Elektron Hisoblash Mashinasi) 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 darajali dasturlashda, 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.

#### MASHINA TILIDA PROGRAMMALASHTIRISH.

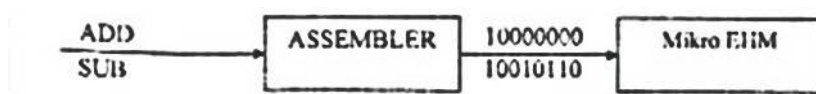


5.1-rasm. Mashina tilida mikroEHM ga ta'lumot kiritish.

Hamma programmalashtirish tillarining ichida mashina tili ko'proq umumiydir. Bu yerda programmist mashina funktsiyasidagi terminda fikr qilishi kerak. Mashina, bajarayotgan har bir boshqarishni programmistga ta'minlab beradi. Bunday boshqarish program man ing vaqt bo'yicha bajarilishini va xotiraga uni talab bo'yicha joylashtirishni optimaliashtirib beradi.

### ASSEMBLER TILIDA PROGRAMMALASHTIRISH.

Assembler tilini ishlatganda programmistni MP dan assembler ajratib turadi. Assembler - bu assembler tilidan mashina tiliga o'tkazuvchi (translyatsiya qiluvchi) vositadir



5.2-rasm. Assembler tili.

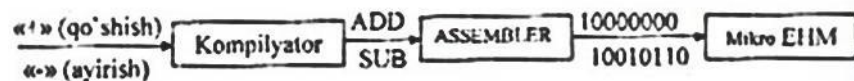
Assembler tilining mashina tilidan afzalligi shundaki, ya'ni bu tilda programmalash mashina kodlariga nisbatan oddiy bo'lgan ramziy belgilarda bajariladi.

Bu yerda MP ga qo'shimcha funktsiya yukianadi ya'ni assembler tilida programma tuzish hamda programmani saqlash uchun EHM xotirasining hajrainsi kattalashtirish.

Bu usulda programmaiashda umumiy xotiradan samarali foydalanish yomonlashadi, chunki haqiqiy programma orqali mashina programmaning hammasini boshqarish mumkinchiligi imkoni bo'lmaydi.

### YUQORI DARAJADAGI TIL.

Bu tilga quyidagilar kiradi: ALGOL, FORTRAN, K080L, BEYSIK, RL/1. RL/L2, PA SKA L., KARAT, KLIPPER, DELFI va boshqalar. Bu yuqori tillarning har biri alohida xususiyatlarga ega va har xil belgilar, funktsiyalarga tayanadi. Bulaming o'xshashligi shundaki, ya'ni uining hammasi tajribasi yo'q dasturchiga o'zica yaqin bo'lgan professional tilda programmalashga imkon beradi.



5. 3-rasm. Yuqori darajadagi iida mikroEHM ga ta'lumot kiritish.

Programma tuzayotcanda mashina tiliga ko‘proq irrtiyoz beriladi. Chunki mashina tilida xotira registrlarini yacheykalari juda samarali ishlatiladi.

Mashina tilining kamchiliklaridan biri shundaki, ya’ni bu tilda murakkab masalaiami yechish uchun programma tuzish juda qiyin. Chunki programma bevosita ikkilik kodida tuzilishi va EMM kintilishi kerak. Ikkilik kodida bajariidigan amallami eslab qolish mumkin emas, xatouiiklarni tuzatish xam qiyin.

Assembler tilida mashina tiliga nisbatan programma tuzish oson hamda programma aisqaroq hajmgga ega, uni uizishga ko‘p vaqt ketmaydi. Lekin bu til qo‘shimcha xotirani talab etadi, xotira katakchalari dcyarli samarali ishlatiimaydi.

Yuqori darajadagi til esa programmani tez tuxishi bilan ajralib turadi. Bu til qo‘shimcha xotirani talab etadi. Shu bilan birgalikda eng murakkab masalalarni yecha oladi. Bu tilda ishlovchi EHM ancha qimmat turadi.

Programmash oddiy tilda va shunga o‘xshash belgilar asosida tuziladi. Programmaning xatolarini xam tuzatish oson.

Assembler tilida tuzilgan programma ketma-ket so‘zlardan yoki operatorlardan tashkil topgan.

Assembler tilidagi operator quyidagi to‘rtta maydonni o‘z ichiga oladi:

belgilar maydoni;

amallar, buyruqlar maydoni;

operandlar maydoni;

sharxlash maydoni.

Shu maydonlardan faqatgina amallar maydoni (buyruqlar maydoni) zarur, qolgan maydoniar esa bo‘lmasligi ham mumkin.

### 5.1-jadval

Belgilar maydoni	Buyruqlar maydoni	Operandlar maydoni	Operandlar maydoni	Sharxlash maydoni
@, A, B, ..., F	MOVA	B		
M1:	MOVA	B		
M2:	LDAXH	00	OE	

Belgilar maydoni berilgan operandning simvolik ismini yozish uchun kerak. Simvolik ism programmaning boshqa bir operatorida shu operatorga murojaat qilish kerak bo‘lsagina kerak bo‘ladi.

Simvolik ism sifatida alfavit, rakam ketma-ketligidan tuzilgan ishoralarni (belgilarni) ishlatish mumkin. Eng yuqori uzunlikdagi ism har xil assembler tllarida har xildir va odat bo'yicha 8 ta ishoradan oshmasligi kerak.

Belgiga bittadan to'rttagacha yoki 8 tagacha bo'lgan ishoralarni ishlatish

-----  
 @, ? A, V .... 2, 0 ... 9.  
 Belgining birinchi ishorasi harf yoki @ (?) lar bo'lishi kerak. Misol: A V S  
 @:  
 @ 001 :  
 AV:  
 ? A I I S:  
 Amallar maydoni. (buyruqlar maydoni) - operatorlar bajaradigan amallarni mnemonik harflar bilan yozilishini o'z ichiga oladi.  
 Mnemonik yozishda belgilar soni amallarning turiga qarab bitta harfdan boshlab bir nechtagacha (tilga bog'lik holda) bo'lishi mumkin.  
 Misol: LAV: MOV A,V  
           belgi  amal  operand  
           buyruq kodi  
 LAB: RAR  
 ?B11C: LDA

mumkin 1111H

Operandlar maydoni - bitta yoki bir nechta operandlarga ajratilgan. Operandlar bir nechta bo'lsa, ular bir-biridan vergul bilan ajratiladi. Operandlar sifatida sonlar, simvolik ism va ifoda bo'lishi mumkin. Ifodada arifmetik ishoralar ishlatilishi mumkin. O'zgarmas sonlar o'n oltilik, o'nlik, sakkizlik, yoki ikkilikda bo'lishi mumkin.

Masalan, o'nlik 27 sonini har xil ko'rinishda keltirishga misollar:

o'n oltilikda 1 VN ( $1 \text{ VN} = 1 \cdot 16^1 + V \cdot 16^0 - 1 \cdot 16 + 11 \cdot 1 = 27$ )

o'nlikda 27 D yoki 27

sakkizlikda 33 O yoki 33 Q

ikkilikda 11011B yoki MVIB:22N ma'nosi V registriga o'n oltilikda 22 sonini ko'chirilsin.

STA 32841D - Akkumulyatordagi qiymat o'nlik sonidagi 32841 adresda eslab qolinsin.

IN 62Q - Sakkizlikdagi 62 kanalni qiymati kiritilsin. Tekstli belgilar (simvollar). Tekstli belgilar bittalik opostroflarning orasiga olingan bitta yoki bir qancha ishoralardan tuzilgan. Misol:

A'-bitta tekstli belgi.

'AV' "S+D` - tekstii belgilar ketma-ketligi.

Ifoda, Ifoda quyidagi operatorlar bilan bog'liq bo'lgan I - ILI, o'zgarmas simvollardan tuzilgan. Ifodalarni hisoblaganda amallar quyidagi ketma-ketlik bo'yicha bajariladi.

- 1) qavsdagi ifodalar:
- 2) \*, F, MOD, SHL. SNR;
- 3) +, -;
- 4) NOT;
- 5) AND:
- 6) OR; XOR.

Ifodalarga misollar:

(A+V)F2 AND 11V+17;

`A` AND 77Q.

Bu yerda:

MOD - bo'lingandan keyin qolgan qoldiq.

NOT - razryad bo'yicha inkor etish.

AND - razryad bo'yicha I.

OR - razryad bo'yicha ILI

XOR - razryad bo'yicha chiqarib tashlash.

SHL- chapga surish.

SHR - o'nga surish.

Sharxlash maydoni. Bu maydonga operatorning yoki boshlang'ich programmaning fragmentini tushuntirib beradigan va programmani bajarishga halaqit bermaydigan ixtiyoriy tekstni joylashtirish mumkin.

Hamma sanalgan (keltirilgan) maydonlar bir-biridan eng kami bitta probel yoki qo'shtirnoqli belgilar orqali ajratiladi.

Assembler tilida quyidagi gurux operatorlari bor:

mashina buyruqlarining operatori;

pseudobuyruqlar operatori;

—makrokomanda;

- sharhlash maydoni;
- assemblerni boshqarish buyrug‘i.

Mashina buyruqlarining operatori.

MikroEHM dagi mashina buyruqlarini simvolik ko‘rinishda yozishga to‘g‘ri keladi. Har bir bunday operator translyatsiya yoki assemblerlash natijasida tegishli mashina buyrug‘iga o‘zgartiriladi.

Psevdo buyruqlar operatorlari (aniqlovchi buyruqlar) quyidagi vazifalarni bajarish uchun mo‘ljallangan:

xotirani rezervlash (zaxiralash) uchun;

buyruqlar sanagichini boshqarish uchun;

o‘zgaruvchan va o‘zgarmas qiymatlarni ifoda etish (yozish) uchun;

ekvivalent qiymatlarni ko‘rsatish uchun;

programma modulining boshlanishi va oxirini va sh.o‘xsh. Berish uchun. Har xil assembler tillaridagi psevdobuyruqlarda amallar mnemonikasi har xildir.

Masalan, DS - xotirani aniqlash, yozilishi: <belgi> :

DS «ifoda».

DB baytni aniqlash: <belgi>: DB<ro‘yxat>. DW so‘zni aniqlash: <belgi>:DW <ro‘yxat>. ORQ - boshlanish;

<belgi>: ORQ <ifoda>. YeQV - teng; <belgi> EQV <ifoda>.

DS psevdobuyrug‘i uchun misol: (quyida keltirilgan har qanday psevdobuyruq M1 indeksi bo‘yicha 11ta baytni rezervlaydi.

M1: DS 11;

M1: DS OV N;

M1:DSK-3:

M1: DSK\*K-5

Buyerda V=11, K=8.

DS psevdobuyrug‘i uchun

M2: DB 0 F1H;

DBK+ 1 0, 1 FN-ZF2, ‘A’.

Makrobuyruqlar. Makrobuyruqlar, assembler tilidagi shunday operatorki, ular translyatsiya davomida tilni boshqa operatorlar ketma-ketligi bilan almashtiradi. Bunday ketma-ketlikni makrobuyruqni makrokengaytirish deyiladi. Bundan tashqari har bir makrobuyruqqa makroaniqlash to‘g‘ri keladi.

Makroaniqlash makrobuyruq yoki makroaniqlash bibliotekasi joylashgan programma modulida bo‘lishi mumkin.

Sharxlash. Berilgan (boshlang‘ich) programmaning tekstiga tushuntirish berish uchun kerak. Sharxlash operatori ayrim ajratilgan ishora bilan boshlanadi. Masalan, «.» yoki «.» belgilari orqali. Bu belgilardan keyin erkin tekst kelishi mumkin.

Assemblerni boshqaradigan buyruqlar quyidagi ishlarni bajarishi mumkin:

- 1) listingni chiqarish va uni shaklini o‘zgartirish, mashinani ishlash rejimini boshqarish;
- 2) boshlang‘ich programmani kiritish va translyatsiyalar natijasini chiqarish maqsadida tashqi qurilmanni tanlashni ta’minlash uchun;
- 3) translyatsiyani davom etkazish, to‘xtatish va boshqa amallarni bajarishni amalga oshirish uchun.

Assemblerni boshqaradigan buyruqlar assembler tilining biron-bir qismi emasdir. Bu buyruqlar terminaldan boshlang‘ich programmani tuzishni diaolog rejimida dasturchi tomonidan har qanday bosqichda berilishi mumkin.

Odatda dasturlash yuqori saviyali dasturlash tillari (Delphi, Java, C++, Python) vositasida amalga oshiriladi. Bu dasturlash tillarining semantikasi odam tiliga yaqinligi tufayli dastur tuzish jarayoni ancha oson kechadi. Ko‘p ishlatiladigan dasturlash tillari. Biz hozir biladigan va ishlatadigan tillarning barchasi shu guruhga mansub. Ular insonga "tushunarli" tilda yoziladi. Ingliz tilini yaxshi biluvchilar programma kodini qiynalmasdan tushunishlari mumkin. Bu guruhga Fortran, Algol, C, Pascal, Cobol va h.k. tillar kiradi(ko‘pchiligi hozirda deyarli qo‘llanilmaydi). Eng birinchi paydo bo‘lgan tillardan to hozirgi zamonaviy tillargacha ishlatish mumkin. Lekin, hozirgi web texnologiya orqali ishlaydigan tillarda(PHP, ASP.NET, JSP) bunday dasturlar tuzilmaydi. Chunki bunday

dasturlarning ishlashi uchun yana bir amaliy dastur ishlab turishi kerak. Hozirda, amaliy dasturlar, asosan, Visual C++, C#, Borland Delphi, Borland C++, Java, Python kabi tillarda tuziladi. O'zbekistonda ko'pchilik Delphi dan foydalanadi. Buning asosiy sababi: soddaligi, komponentlarning ko'pligi, interfeysining tushunariligi va h.k. Delphida birinchi ishlagan odam ham qanaqadir dastur tuzishi oson kechadi. Lekin, Windows da dasturning asosiy ishlash mohiyatini ancha keyin biladi(komponentlarning ko'pligi va API funksiyalari dasturda ko'rsatilmaligi uchun). Yana bir tarafi, Delphi(Pascal) operativ xotirani tejashga kelganda ancha oqsaydi. Unda o'zgaruvchilarni oldindan e'lon qilib qo'yish evaziga ishlatilmaydigan o'zgaruvchilar va massivlar ham joy olib turadi. Eng keng tarqalgan dasturlash tili(Windows OS ida) Microsoft Visual C++ tilidir CODE VISION bitta kristalli mikroprotessorlarga dasturlar tuziladi va uni PROTEUS dasturida simulyatsiya qilinadi. Ko'pchilik dasturlar hozirda shu tilda tuziladi. Umuman olganda, C ga o'xshash(C-подобный) tillar hozirda dasturlashda yetakchi. Deyarli hamma zamonaviy tillarning asosida C yotadi. Bundan tashqari, Turli komputer o'yinlari tuzishda yoki kichik hajmdagi dasturlar tayyorlashda LUA script yoki JavaScript tillari ham keng ishlatilmoqda. Biz sizga hozirgi kunda keng tarqalgan desktop dasturlashda ishlatiladigan dasturlash tillaridan bazilari haqida aytib o'tamiz: Delphi (talaff. délfı) — dasturlash tillaridan biri. Borland firmasi tomonidan ishlab chiqarilgan. Delphi dasturlash tili ishlatiladi va avvaldan Borland Delphi paketi tarkibiga kiritilgan. Shu bilan bir qatorda 2003-yildan hozirgacha qo'llanilayotgan shu nomga ega bulgan. Object Pascal — Pascal tilidan bir qancha kengaytirishlar va to'ldirishlar orqali kelib chiqqan bo'lib, u ob'yektga yo'naltirilgan dasturlash tili hisoblanadi. Avvaldan ushbu dasturlash muhiti faqatgina Microsoft Windows amaliyot tizimi uchun dasturlar yaratishga mo'ljallangan, keyinchalik esa GNU/Linux hamda Kylix tizimlari uchun moslashtirildi, lekin 2002-yilgi Kylix 3 sonidan so'ng ishlab chiqarish to'xtatildi, ko'p o'tmay esa Microsoft.NET tizimini qo'llab quvvatlashi to'g'risida e'lon qilindi. Lazarus proekti amaliyotidagi (Free Pascal) dasturlash tili Delphi dasturlash muhitida GNU/Linux, Mac OS X va Windows CE platformalari uchun

dasturlar yaratishga imkoniyat beradi. Visual Basic (talaffuzi: "Vijual Beysik") – Microsoft korporatsiyadan dasturlash tili va uning uchun dasturlash muhitidir. U BASICdan ko'p tushunchalar oldi va tez rasmi interfeys bilan dasturlar taraqqiyot ta'minlaydi. Oxirgi versiya 6.0 1998 yilda reliz kelishdi. Microsoftdan voris Visual Basic .NET 2002 yilda paydo bo'ldi. Java dasturlash tili - eng yaxshi dasturlash tillaridan biri bo'lib unda korporativ darajadagi mahsulotlarni(dasturlarni) yaratish mumkin.Bu dasturlash tili Oak dasturlash tili asosida paydo bo'ldi. Oak dasturlash tili 90-yillarning boshida Sun Microsystems tomonidan platformaga(Operatsion tizimga) bog'liq bo'lmagan holda ishlovchi yangi avlod aqlli qurilmalarini yaratishni maqsad qilib harakat boshlagan edi. Bunga erishish uchun Sun hodimlari C++ ni ishlatishni rejalashtirdilar, lekin ba'zi sabablarga ko'ra bu fikridan voz kechishdi.Oak muvofaqiyatsiz chiqdi va 1995-yilda Sun uning nomini Java ga almashtirdi, va uni WWW rivojlanishiga hizmat qilishi uchun ma'lum o'zgarishlar qilishdi. Java Obyektga Yo'naltirilgan Dasturlash (OOP-object oriented programming) tili va u C++ ga ancha o'xshash. Eng ko'p yo'l qo'yildigan xatolarga sabab bo'luvchi qismlari olib tashlanib, Java dasturlash tili ancha soddalashtirildi. Java kod yozilgan fayllar (\*.java bilan nihoyalanuvchi) kompilatsiyadan keyin bayt kod(bytecode) ga o'tadi va bu bayt kod interpretator tomonidan o'qib yurgizdiriladi. C++ (talaffuzi: si plyus plyus) — turli maqsadlar uchun mo'ljallangan dasturlash tili. 1979-yili Bell Labsda Biyarne Stroustrup tomonidan C dasturlash tilining imkoniyatlarini kengaytirish va OOP(object Oriented Programming) xususiyatini kiritish maqsadida ishlab chiqarilgan. Boshida „C with Classes“ deb atalgan, 1983-yili hozirgi nom bilan ya'ni C++ deb o'zgartirilgan. C++ C da yozilgan dasturlarni kompilyatsiya qila oladi, ammo C kompilyatori bu xususiyatga ega emas. C++ tili operatsiyon tizimlarga aloqador qisimlarni, klient-server dasturlarni, EHM o'yinlarini, kundalik ehtiyojda qo'llaniladigan dasturlarni va shu kabi turli maqsadlarda ishlatiladigan dasturlarni ishlab chiqarishda qo'llaniladi. Quyidagi jadvalda programmalash tillari haqida ma'lumotlar keltirilgan.

Til	Yaratilgan yili	Mualliflar	Tashkilot, firma
Ada	1979-80	Jean Ichbian	<u>Cii-Honeywell (Fransiya)</u>
Algol	1960		International Committee
ARL	1961-1962	Kenneth Iverson, Adin Falkoff	IBM
DELPHI	1995		Borland
VASIS, Beysik	1964-1965	JohnKemeny, Thomas Kurtz	Dartmouth Colleje
C	1972-1973	Dennis Ritchie	Bell Laboratories
C++	1980	Bjarne Strostrup	Bell Laboratories
Kobol	1959-1961	Grace Murray Hopper	
Fort	1971	Charles H Moore	
FORTRAN	1950-1958	John Backus	IBM
HTML	1989	Tim Berners-LI	CERN, Jeneva
LISP, LISP	1956-1960	John MCCarthy	
LOGO	1968-70	Seymour Papert	Massachusetts Institute of Techn.
Pascal	1967-1971	Niklaus Wirth	Federal Institute of Technology (SHveysariya)
PL 1	1964, 1966		

Savollar:

1. «Mashina tillari», «Assembler tili» xaqida tushuntiring.
2. «Yuqori darajadagi programmalash tillari». Mnemokodlar to'g'risida tushuncha, «Assembler» tilining maydonlari, tuzilishi (strukturasi)xaqida gapirib bering.
3. MP uchun maxsus programmalashtiriladigan vositalarini aytib bering.
4. Redaktorlovchi, yuklovchi, modellashtiruvchi, sozlovchi programmalar xaqida nimani bilasiz.
5. «Kompilyator» va «Assembler» programmali, operatsion sistemalar va monitor to'g'risida tushuncha bering.