

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

O'ZBEKISTON MILLIY UNIVERSITETI

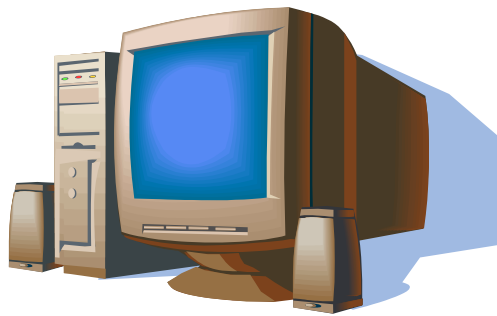
REFERAT

MAVZU: DASTURLASH TILLARI HAQIDA TUSHUNCHA

Bajardi: Olimova N.

Tekshirdi: Mamatqulova M.

TOSHKENT-2014



DASTURLASH TILLARI

Algoritmlarni yozish uchun qo'llaniladigan tillar algoritmik tillar deb ataladi. Algoritmik tilni EHM ham tushunsa, u holda bu til dasturlash tili deb ataladi. Demak, algoritmik til yoki dasturlash tili ham berilgan misol yoki masalani yechish algoritmining yozish usullaridan biri ekan. Shu yozish usullarining qanchalik mashinaga bog'liqligi yoki tabiiy tillarga yaqinligiga qarab ular turlicha nomlanadi.

Ma'lumki, har bir EHM o'zining buyruqlar (komandalar) sistemasi va mashina tiliga ega. Bu til ma'lum qonun-qoidalar asosida yozilgan 0 va 1 raqamlari ketma-ketligidan iboratdir. Bu qonun-qoidalar asosan, ma'lum bir EHM qurilmalarining tuzilishiga bog'liq bo'lib, aynan shu quruq mashinalari uchun o'rinlidir. Har bir EHM o'zining tilida yozilgan dasturnigina tushuna oladi va uni ijro etadi. Masalan, ikkinchi avlodga tegishli BESM-6 mashinasida $B \cdot C + A$ ifodani hisoblash dasturi quyidagicha bo'lishi mumkin:

Registor indeksi	Amal kodi	Makon (adres)
0 000	00 001 000	000 100 000 000
0 000	00 001 111	000 100 000 001
0 000	00 000 100	000 100 000 010

Bunda har bir buyruq 24 ta 0 va 1 lardan (24 ta ikkilik xonadan) iborat bo'lib, uning birinchi to'rttasi registrning indeksini ifodalasa, keyingi 8 ta xona (amal kodi) bajarilishi kerak bo'lgan amalni bildiradi. Qolgan xonalar xotira makonini (adreslarni) ifodalaydi. Yuqoridagi o'zgaruvchilar uchun xotira makonlari quyidagicha olingan:

V uchun 000 100 000 000 – makon;

S uchun 000 100 000 001 – makon;

A uchun 000 100 000 010 – makon.

Bu dasturda quyidagi amal kodlari ishlatilgan:

00 001 000 – xotiradagi sonni jamlagichga chaqirish;

00 001 111 – jamlagichdagi songa xotiradagi sonni ko'paytirish va natijani jamlagichda saqlash;

00 000 100 - jamlagichdagi songa xotiradagi sonni qo'shish va natijani jamlagichda saqlash.

Xuddi shu dasturni o'zi "ES" (edinaya sistema) EHM yoki boshqa turkum EHM lar uchun butunlay boshqa ko'rinishda bo'lishi mumkin.

O'z-o'zidan ko'rinib turibdiki, hatto sodda dasturlarni mashina tilida yozish ancha mushkul ish ekan. Chunki dastur tuzish jarayonida hamma amal kodlari jarayonini bilish, kerakli registr indekslarini tanlash va har bir o'zgaruvchiga ajratilgan xotira joyning makonini bilish va hokazo talab etiladi. Shu va shunga o'xshash qiyinchiliklarni bartaraf etish maqsadida, belgilash (simvolik) tillar (avtokodlar, assemblerlar) yaratildi. Avtokod va assemblerlar dasturlash jarayonini avtomatlashtirishda birinchi qadam hisoblanadi. Bu tillarda dastur yozish mashina tilida dastur yozishdan oson va qulay. Lekin qar bir mashina turi o'zining avtokodi yoki assembleriga ega va uni shu mashinaning o'zigina tushunadi. Yuqorida keltirilgan dasturni BESM-6 mashinasining avtokodida (BEEHM) quyidagicha yozish mumkin:

<i>CT</i>	<i>V</i>
<i>VMH</i>	<i>S</i>
<i>CT</i>	<i>A</i>

Bunda xotiradagi sonni jamlagichga chaqirishda ChT (ChTeniya), ko'paytirishda UMN (UMNojenie) va qo'shishda SL (SLojenie) kabi mazmunli belgilashlar (mnemonikalar) ishlatiladi.

Xuddi shu dasturni "ES" (edinaya sistema) EHM larining assembler tilida ham har bir o'zgaruvchining mashina xotirasida qanday qolipda (formatda) joylashganligiga qarab, quyidagicha ko'rinishlarda yozish mumkin:

So'z qolipi	O'nlik qolip	Suzuvchi vergullik qolip
<i>Л - 3, B</i>	<i>3AЛ-3,B</i>	<i>ЛE-2,B</i>
<i>M - 2,C</i>	<i>MЛ-3,C</i>	<i>ME-2,C</i>
<i>A - 3, A</i>	<i>AЛ-3,A</i>	<i>AE-2,A</i>

Bu dastur lavhalaridagi belgilashlar ham biror mazmun ifodalaydi, ya'ni L (Load)–yuklash, M (Multiply)–ko'paytirish, A (Add)–qo'shish, Zap (Zero And Add)–tozalash va qo'shish.

Yuqorida keltirilgan dastur lavhalarini mashina to'g'ridan-to'g'ri bajara olmaydi, chunki mashina tilida CHT, UMN, SL A, V, S, L, EAP kabi belgilashlar yo'q. Ana shu belgilashlarni mashina tiliga o'tkazish uchun tarjimon dasturlar (translyatorlar) yaratilgan. Translyatorlarning o'zi ham mashina tilida tuzilgan dasturdan iboratdir. Bu dasturning ijrochisi mashinaning o'zidir, ya'ni EHM o'zi belgilashlar tilida tuzilgan dasturni o'z tiliga o'tkazib oladi. Demak, biz tuzgan dastur translyatorlar uchun boshlang'ich ma'lumot rolini o'ynar ekan. Ana shunday

tarjima qiluvchi dasturlar (algoritmilar) ni yaratish dasturlar sohasida muhim rol o'ynaydi.

Ammo mashinalar soni va xillarini keskin ko'payishi, bir mashina tilida yoki avtokodi (assembleri)da tuzilgan dasturni boshqa mashinaning tushunmasligi (chunki assemblerlar qam mashinaga boqliq til), bu soha bo'yicha mutaxassis bo'lmagan EHM dan foydalanuvchilar uchun dastur yaratishning juda qiyinligi katta-katta muammolarni keltirib chiqaradi va EHM dan foydalanish samaradorligini keskin kamayishiga sabab bo'ladi. Bu va shunga o'xshash muammolarni hal qilish maqsadida yangi tillar, ya'ni mashinaga bog'liq bo'lmagan tillar – algoritmik tillar yaratildi. Bu algoritmik tilni mashina tushinishi uchun mashina tilida yaratilgan translyatorlar yaratildi. Bunday translyatorlar qar bir mashina uchun bir marta tuziladi. Bu turdagi translyatorlar algoritmik tildan to'g'ridan-to'g'ri mashina tiliga yoki avval avtokod yoki assemblerga, so'ngra esa belgilashlar tilidan mashina tiliga tarjima qiladi. Tarjima qilish jarayonini ham mashinaning o'zi bajaradi. Umuman algoritmik til yaratish qiyin emas, lekin shu yaratilgan algoritmik tildan mashina tiliga tarjima qiluvchi tilmoch dasturlarni yaratish og'ir masaladir.

Hozirgi kunda turli-tuman algoritmik va dasturlash tillari mavjud. Algoritmik tilga misol qilib A. P. Ershov boshchiligida yaratilgan algoritmik tilni misol qilib keltirish mumkin. Dasturlash tillarining dastlabkisi FORTRAN tilidir. U hisoblash xarakteridagi masalalarning algoritmini yozish uchun qulaydir. Bu til 1954 yilda prof. J.V.Bekus boshchiligidagi bir guruh amerikalik mutaxassislar tomonidan yaratildi. Fortran inglizcha FORMula TRANslation (FORTRAN) so'zlaridan olingan bo'lib, formulani tarjima qilish degan ma'noni bildiradi. Hozir FORTRAN tilining bir qancha ko'rinishlari (oiladoshlari) mavjud.

Murakkab tarmoqlangan hisoblash jarayonlarining algoritmini yozishga mo'ljallangan tillardan biri Algol-60 (1960 yilda yaratilgan) tilidir. Algol so'zi inglizcha ALGOritmic Language (ALGOL) so'zlaridan olingan bo'lib, algoritmik til demakdir. Algol tilining Algol-68 oiladoshi ham mavjud.

Dasturlashni o'rganishni boshlovchilarga mo'ljallangan dialog sistemasida ishlaydigan turli-tuman jarayonlar algoritmini tuzishga qulay bo'lgan tillardan biri BEYSIK (BASIC) tilidir.

KOBOL va ALGEK tillari iqtisodiy masalalarning algoritmini yozishga, SNOBOL va LISP tillari esa satrlarni qayta ishlashga mo'ljallangan.

Ko'p imkoniyatlarga ega bo'lgan tillardan biri PL 1 (PL 1 – Programming Language 1) dasturlash tilidir. Bu tilning birinchi ko'rinishi 1964 yilda yaratilib, NPL deb nomlangan va FORTRAN, ALGOL, KOBOL kabi tillarning imkoniyatlarini o'z ichiga oladi.

1970 yilda N.Virt yangi tilni yaratdi, bu til mashhur olim Blez Paskal nomi bilan, ya'ni Paskal deb ataldi.

Bu tillardan tashqari maktabda o'quv jarayonlarini ta'minlash uchun “ShKOLNISA” sistemasi ham mavjud. Bu sistemadan AGAT turidagi kompyuterlar yordamida foydalanish mumkin. Sistema sobiq ittifoq FAsi Sibir

bo'limining hisoblash markazi, Novosibirsk Davlat universiteti va Novosibirsk shahridagi maktablararo o'quv ishlab chiqarish kombinati xodimlari hamkorligida yaratilgan bo'lib, uning tarkibida "ROBIK", "RAPIRA", "ShPAGA" kabi til va sistemalar mavjud. "ROBIK" tilining birinchi ko'rinishi 1975 yilda yaratilgan bo'lib, kichik va o'rta yoshdagi (8-11 yosh) maktab o'quvchilarini dasturlashga o'rgatishga mo'ljallangan. "RAPIRA" tilining birinchi ko'rinishi 1978 – 1979 yillarda yaratilgan bo'lib, o'rta va yuqori yoshdagi (11 – 17 yosh) maktab o'quvchilariga mo'ljallangan. Bu ikki tilda dasturlar rus tilida yoziladi. "ShPAGA" sistemasi grafik chizish uchun mo'ljallangan sistemadir.

BEYSIK PROGRAMMALASH TILI

Beysik tilining nomi inglizcha BASIC (Beginners All-purpose Symbolic Instruction Code) ning o'qilishiga mos kelib, boshlovchilar uchun belgili ko'rsatmalarning universal kodi (tili) degan ma'noni anglatadi.

1963 yil may oyida Beysik tili "GENERAL ZLECTRIC" firmasining DATANET-30, GE-225, GE-235 kabi kompyuterlarida qo'llana boshlandi.

Shuni ta'kidlash lozimki, beysik tili shu paytlarda mavjud bo'lgan Fortran, Algol, IOSS, CORC kabi dasturlash tillari asosida yaratilgan bo'lib, bu tillarning ayrim elementlarini o'z ichiga olgan.

Kemeni va Kurts tomonidan yaratilgan 1964 yilda chop etilgan Beysik tilidagi dastur qo'yidagi ko'rinishga ega bo'lgan:

```
10 LET X + (7 + 8) / 3
```

```
20 PRINT X
```

```
30 END
```

1965 yildan 1971 yilgacha Beysik tilining bir qancha ko'rinishlari matbuotda chop etildi. 1975 yilda MITS firmasining xodimlari P. Allen va B. Geyts tomonidan ALTAIR-8800 mikro-EHM lari uchun Beysik tili interpretatori yaratildi.

Keyinchalik bu olimlar tashkil qilgan "Microsoft" firmasi tilning rivojlanishi va mikro-EHM va PEHM larda joriy etilishiga salmoqli hissa qo'shdilar. "Microsoft" firmasi yaratgan Beysik tilining birinchi ko'rinishi: "Apple" va "Tandy" firmalarining PEHM larida joriy etildi. 1979 yilda "Microsoft" firmasi MBASIC (BEYSIK-80) nomli Beysik tilining yangi ko'rinishini yaratdi. Bu til IBM firmasining kompyuterlari uchun yaratilgan interpretator va MS-DOS operasion sistemalari ko'magida keng tarqaldi. 1981 yilda "Microsoft" firmasi IBM PC kompyuterlari uchun Beysik-80 tilining kengaytirilgan ko'rinishi BASIC-A ni yaratishdiki, unda matnlar va grafik holatda ishlash imkoniyatlari mavjud edi. BASIC-A tilining kengaytirilishi va rivojlanishi, yangi imkoniyatlarning qo'shilishi bilan uning yangi ko'rinishi quick BASIC tili vujudga keldi.

Mutaxassislar uchun yaratilgan mini-EqMlar "YES"-1841, "YES"-42, Iskra-1030.11, Neyron I9.66, IBM PC bilan dastur moslikdagi EHM lar uchun Beysik tilining MBASIC ko'rinishi OS SR/M-86 va MS-DOS sistemalari uchun joriy etildi. Bundan tashqari MSX-BASIC tilida ishlovchi Yaponiyaning "Yamasa MSX" va

“Yamaxa MSX-2” xo'jalik kompyuterlari maktab o'quv-hisoblash texnikasi majmuasida (KUVT) keng qo'llanilmoqda.

BEYSIK TILINING ASOSIY BELGILARI

Ma'lumki, ixtiyoriy tilni o'rganishdan maqsad xoh u til tabiiy bo'lsin, xoh u til sun'iy bo'lsin, fikrlarni shu til vositalari yordamida yozma ifodalashdan iboratdir. O'zbek tili o'zbek harflaridan tashkil topganidek, dasturlash tillarining ham o'z harf va belgilari mavjud.

Tilning asosiy belgilari quyidagilardir:

1) 26 lotin harfi:

Aa, Bb, Cc, Dd, Ee, Ff, Gg, Hh, Ii, Jj, Kk, Ll, Mm, Nn, Oo, Pp, Qq, Rr, Ss, Tt, Uu, Vv, Ww, Xx, Yy, Zz.

2) Arab raqamlari:

1, 2, 3, 4, 5, 6, 7, 8, 9, 0.

3) Mahsus belgilar:

, , : q Q / * % < q > () [] { } # \$ & “ Ъ ^ ! q _

4) Rus harflari:

Aa, Бб, Вв, Гг, Дд, ее, Жж, Зз, Ии, Йй, Кк, Лл, Мм, Нн, Оо, Пп, Рр, Сс, Тт, Уу, Фф, Хх, сс, Чч, Шш, Xx, ъъ, Ыы, Эе, Юю, Яя.

Bu belgilarning ichida ayrimlari masalan, matematikada ko'p ishlatiladigan rim raqamlari, grek qarflari va hokazolar yo'q. Bularni tasvirlash uchun yuqoridagi belgilardan foydalanish mumkin.

STANDART FUNKSIYALAR

Juda ko'p qollarda qisoblash xarakteridagi misollarni echishda ayrim funksiyalarning qiymatlarini qisoblashga to'qri keladi. Masalan, va q.k. Bu funksiyalar standart funksiyalar deb ataladi. Beysik tilida bu funksiyalarni qisoblash uchun translyatorning kutubxonasiga kiritilgan maxsus qism dasturlardan foydalaniladi. Standart funksiyalarning nomlari Beysik tilida 3 ta lotin qarfidan iborat bo'ladi. Funksiyalarning argumenti ixtiyoriy arifmetik ifoda bo'lib, ular funksiya nomidan keyin kichik qavslar ichida turishi shart.

ARIFMETIK IFODALAR VA ULARNING BEYSIK DASTURLASH TILIDA YOZILISHI

Arifmetik ifodalar qar doim sonli qiymatga ega bo'lib, ular o'zgarmas va o'zgaruvchi kattaliklar, funksiyalar, oddiy kasrlar, arifmetik belgilar yordamida hosil qilinadi.

Ma'lumki matematikada darajaga oshirish belgisi yo'q. Ammo dasturlash tillarida qar qanaqa yozuv bir satrda yozilganligi sababli darajaga oshirish (), ko'paytirish (*) va ikki sonni bo'lib butun qismini ajratish (MOD) belgilari kiritilgan.

Misol:

Matematikada yozilishi	Dasturlash tilida yozilishi
a^5	A^5
$a \times b$	$A * B$
a / b yoki $\frac{a}{b}$	A / V
A ni B ga bo'lganda butun qismini ajratish	$A \text{ MOD } B$

Arifmetik ifodani qisoblash uchun unda ishtirok etgan barcha o'zgaruvchilarning qiymati oldindan ma'lum bo'lishi shart. Arifmetik ifodalarni hisoblashda quyidagilarni qisobga olish kerak:

1. Oldin qavslar ichidagi amallar bajariladi. Agar qavslar juftligi bir qancha bo'lsa, u holda hisoblash eng kichik qavslardan boshlanadi.
2. qavslar ichida amallar quyidagi tartibda bajariladi:
 - funksiyaning qiymati hisoblanadi;
 - darajaga oshiriladi;
 - ko'paytirish, bo'lish va butun natijali bo'lish;
 - qo'shish va ayirish;
3. Bir xil amallar ketma-ket kelsa, hisoblash chapdan o'ngga qarab bajariladi. quyidagi misolda amallarning bajarilish ketma-ketligi keltirilgan:

-	X * B	*	C / D	*	I^E	+	A*	SIN(X + 3)
9	4	5	6	7	3	10	8	2 1

Bunga quyidagi ifoda mos keladi: $\frac{x \cdot b \cdot c}{d} i^e + a \sin(x + 3)$.

$\sqrt{2 - \sin^2 2x}$ ifodani Beysik tilida yozilishi quyidagicha bo'ladi:

$$\text{SQR}(2 - \text{SIN}(2 * X)^2)$$

$\sqrt[p]{x}$ ixtiyoriy darajali ildiz ko'rinishidagi ifodani hisoblashda

$x^{1/p}$ ko'rinishidagi ekvivalent formuladan foydalaniladi.

Masalan, $\sqrt[5]{(x-3)^3 + (y+2)^2}$ ifoda quyidagicha yoziladi:

$$((X-3)^3 + (Y+2)^2)^{(1/5)}$$

Shuni nazarda tutish lozimki, ikkita arifmetik amalni ketma-ket yozish mumkin emas. Masalan, $\frac{a}{-b}$ ko'rinishidagi misolni $A / -B$ ko'rinishida yozish xato bo'ladi. Bu ifodani $A / (-B)$ yoki $-A / B$ ko'rinishida yozish mumkin. Qavslarni turli joylarda

ishlatish turli natijalarga olib kelishi mumkin. Masalan, $(A + B) / C$ yozuv $\frac{a + b}{c}$ ifodaning, $A + B / C$ yozuv esa $a + \frac{b}{c}$ ifodaning Beysikdagi yozilishidir.

Manfiy qiymatni faqat butun darajaga oshirish mumkin.

SHARTLI IFODALAR VA ULARNI BEYSIK DASTURLASH TILIDA YOZILISHI

Shartli ifodalar kattaliklarni taqqoslashdan hosil bo'ladi va ular munosabatni tashkil qiladi.

Beysik tilida munosabatlarda quyidagi taqqoslash belgilari qo'llaniladi:

Munosabat	Taqqoslash iborasi	Misollar
Kichik ($<$)	$<$	$X < Y$
Katta emas ($\leq$)	$\leq$	$X \leq Y$
Teng ($=$)	$=$	$X = Y$
Teng emas ($\neq$)	$\neq$	$X \neq Y$
Kichik emas ($\geq$)	$\geq$	$X \geq Y$
Katta ($>$)	$>$	$X > Y$

Munosabatlarning o'rinli yoki o'rinsizligiga qarab, uning qiymati rost yoki yolg'on bo'lishi mumkin. Masalan, $1 > 0$ munosabatning qiymati har doim rost bo'lsa, $4 > 5$ munosabatning qiymati har doim yolg'ondir. Munosabatlarning qiymati odatda mantiqiy kattaliklar deb ataladi va ular yuqoridagi shartlarning qiymatidan iboratdir.

Berilishi	Beysikda yozilishi
$d < 0$	$D < 0$
$(a + b)^2 \leq (c + d)^2$	$(A + B)^2 \leq (C + D)^2$
$ \sin x \geq 1$	$ABS(SIN(X)) \geq 1$

Yuqorida keltirilgan munosabatlar oddiy yoki sodda munosabatlar deb ataladi. Oddiy munosabatlardan mantiqiy amallar yordamida murakkab munosabatlar yoki mantiqiy ifodalar hosil qilinadi. Beysik tilida mantiqiy amal belgilari sifatida AND (mantiqiy ko'paytirish), OR (mantiqiy qo'shish) va NOT (inkor) so'zlari ishlatiladi.

AND (va), OR (yoki) va NOT (inkor) amallarining natijasini quyida keltirilgan jadvaldan bilish mumkin. Jadvalda X va Y oddiy munosabatlar, R –rost va YO–yolg'on X va Y munosabatlarning mumkin bo'lgan qiymatlaridir.

X	Y	X AND Y	X OR Y	NOT X
R	R	R	R	YO
R	YO	YO	R	YO
YO	R	YO	R	R
YO	YO	YO	YO	R

Jadvaldan ko'rinib turibdiki, AND amalining natijasi rost bo'lishi uchun X va Y lar rost bo'lishi, OR amalining natijasi rost bo'lishi uchun X va Y ning birortasi rost bo'lishi yetarli. NOT amalida argumentning yolg'on bo'lishi, uning rostligini ta'minlaydi.

Murakkab munosabatlarga misollar:

$$4 < 5 \text{ AND } X \leq Y$$

$$\sin(X) = 1 \text{ OR } X = X$$

$$\text{NOT } (X <> (Y+1)^2)$$

Mantiqiy ifodalarni hisoblash tartibi quyidagichadir:

- mantiqiy ifoda tarkibiga kiruvchi arifmetik ifodalar hisoblanadi;
- oddiy munosabatlarning rost yoki yolg'onligi aniqlanadi;
- mantiqiy amallar bajariladi.

Mantiqiy amallar ketma-ket kelganda oldin NOT, so'ngra AND va oxirida OR amali bajariladi. Lozim bo'lganda qavslar yordamida bu amallarning ketma-ketligini o'zgartirish mumkin.

Matematikada yuqoridagi amallar (AND), (OR) va (NOT) belgilar yordamida ham ifodalanadi.