

4 - Mavzu: Ijrochi va uning ko'rsatmalari

Tilning asosiy konstruksiyasi

XVII asrda yashagan va dunyoda birinchi hisoblash mashinasini yaratgan frantsuz olimi Blez Paskal sharafiga shunday deb nomlangan PASKAL tili Shveysariyaning Syurix shahridagi Oliy texnika maktabining professori Niklaus Virt tomonidan yaratilgan bo'lib, eng birinchi hisoblash qurilmalaridan birini ixtiro qilgan Blez Paskal sharafiga Paskal deb nomlangan. O'zining soddaligi, mantiqiyli va samaraliligi tufayli bu til butun dunyoga tezda tarqaldi. Hozirgi paytda barcha hisoblash mashinalari, xususan, mikrokompyuterlar ham shu tilda ishlash imkoniyatiga ega. Dasturlar matnining to'g'riligini osonlik bilan tekshirish mumkinligi, ularning ma'nosi yaqqol ko'zga tashlanishi va oddiyligi bilan ajralib turadi. Paskal tili ancha murakkab va ko'p vaqt oladigan hisoblash ishlarini bajarishga mo'ljallangan tarkiblashtirilgan dasturlar tuzishga imkon beradi. Yana bir afzalligi shundan iboratki, foydalanuvchi xatolikka yo'l qo'ymasligi uchun yoki xato yozib qo'ygan bo'lsa, tez tuzatib olishi uchun dasturda ishlatilgan o'zgaruvchilar oldindan qaysi turga (toifaga) mansub ekanligi belgilab qo'yilgan bo'ladi. Shu bilan birga dasturning barcha elementlari haqida ma'lumot tavsiflash bo'limida mujassamlashgan bo'ladi. Operatorlar soni esa minimal darajada kamaytirilgandir.

Paskal tilining algoritmik alifbosi

Paskal algoritmik tilining alifbosi quyidagilardan iborat:

- 1) 26 ta lotin harflari;
- 2) 0 dan 9 gacha arab raqamlari;
- 3) 32 ta kirill harflari;
- 4) maxsus belgilar – (+ = * / < >)

Paskal tilida so'z deb bir nechta belgilar ketma-ketligi tushuniladi. Xizmatchi so'z deb Paskal tilidagi standart nom tushuniladi. Bu nom maxsus ma'noni anglatadi va uni ma'lumotlarga berib bo'lmaydi. Masalan, *PROGRAM*, *BEGIN*, *END* va h.k. Paskal tilidagi ma'lumotlarning elementlari bo'lib o'zgaruvchilar, o'zgarmaslar, izohlar xizmat qiladi. O'zgaruvchilar deb hisoblash jarayonida o'z qiymatini o'zgartiradigan kattaliklarga aytiladi. O'zgaruvchilarning nomlari (identifikatorlar) harflardan yoki harf va raqamlardan iborat buladi. Belgilar soni 8 tadan oshmasligi kerak. O'zgarmaslar (const) deb hisoblash jarayonida o'z qiymatini o'zgartirmaydigan kattaliklarga aytiladi. Bularga ham o'zgaruvchilar kabi nom beriladi.

Izohlar

Dasturning ma'lum qismini tavsiflash uchun ishlatiladi va bu qatorda hech qanday amal bajarilmaydi, ya'ni dasturning biror blokini yaxshiroq tushunishga xizmat qiladi. Izoh (*, *) yoki { } simvollar orasida beriladi. Katta qavslar {*, *} va [*, *] lar ham ishlatilishi mumkin.

Ma'lumotlar turlari

Paskal tilida ma'lumotlarning toifalari ikki xil bo'ladi:

Oddiy-skalyar va murakkab. Skalyar toifa, o'z navbatida, o'zgaruvchi va standart toifalarga bo'linadi. O'zgaruvchi toifaga qayd qilingan va chegaralangan toifalar kirsa, standart toifaga butun - **INTEGER**, haqiqiy - **REAL**, mantiqiy - **BOOLEAN** simvol - **CHAR** toifalar kiradi. Bundan tashqari, Turbo Paskalda qator - **STRING** turi kiritilgan.

Murakkab toifalarga esa ma'lumotlarning muntazam (massiv), to'plam, aralash (yozuvlar), murojaat va faylli toifalari kiradi. Ma'lumotlarning standart turlari. Standart turdagi ma'lumotlar dasturda maxsus xizmatchi so'zlar yordamida tavsiflanadi: **INTEGER - BUTUN** toifadagi ma'lumotlar faqat butun sonlarni qabul qiladi. Ular Paskal tilida quyidagicha ifodalanadi: -7, 10,89, -35, 0 va h.k.

REAL - haqiqiy toifadagi ma'lumotlar haqiqiy sonlarni qabul qiladi va ikki xil ko'rinishda bo'ladi:

- a) qo'zg'almas nuqtali haqiqiy sonlar (3.56, 0.88, -150.45);
- b) qo'zg'aluvchan nuqtali haqiqiy sonlar (0.546=546E-03, 96.78*100=96.78E02 va h.k.

BOOLEAN - MANTIQIY

BOOLEAN - MANTIQIY toifadagi ma'lumotlar. Bu nom ingliz matematigi Jorj Bul sharafiga shunday deb ataladi va shu turdagi o'zgaruvchilar faqat ikkita qiymat qabul qiladi, *TRUE -ROST* yoki *FALSE -YOLG'ON*. Bu ma'lumotlar ustida mantiqiy ko'paytirish **AND**, mantiqiy qo'shish **or** va mantiqiy inkor qilish **NOT** amallarini bajarish mumkin. Quyidagi jadvalda **BOOLEAN** turidagi A va B o'zgaruvchilarda mantiqiy amallarning bajarilishi natijasi keltirilgan. **CHAR - BELGI** (simvol) toifadagi ma'lumotlar qiymat sifatida kodlar jadvalidagi belgilarning bittasini qabul qiladi. Bu ma'lumotlar apostrof ('belgisi) ichiga olib yoziladi.

Ifodalar

Paskal algoritmik tilida arifmetik, mantiqiy va belgili ifodalar aniqlangan. Arifmetik ifodalar o'zgaruvchilar, o'zgarmlar, arifmetik amal belgilari qavs va standart funktsiyalardan tashkil topadi. Arifmetik ifodada qatnashayotgan amallar ustivorligiga qarab bajariladi. Amallar quyidagi ustivorlikka ega:

1. Qavs ichidagi amallar va standart funktsiyalarni hisoblash.
2. Ko'paytirish, bo'lish amallari.
3. Qo'shish va ayirish amallari.

Turbo Paskal algoritmik tilida 10^n ko'rinishidagi sonlarni $E+n$, 10^{-n} ko'rinishidagi sonlarni esa $E-n$ kabi tasvirlash mumkin. 10 ning musbat darajasini ifodalashda «+» ishorasini tashlab ketish ham mumkin. Masalan: $10^6 = E6$; $3000 = 3E3$.

Turbo Paskal tilidagi dasturning tuzilishi

Dastur deb algoritmik tilda kompyuterda masala echish uchun qadamma-qadam yozilgan aniq va mukammal algoritimga aytiladi. Turbo Paskal tilidagi dastur

tarkibi 2 asosiy qismdan iborat bo'ladi. Bular dastur sarlavhasi va dastur tanasi. Dastur tanasidan keyin nuqta qo'yiladi, bu nuqta dastur oxirini bildiradi. Dastur sarlavhasi dastur tanasidan nuqta vergul bilan ajratiladi: <dastur >:: =<dastur sarlavhasi>;<dastur tanasi>

Dastur sarlavhasi PROGRAM xizmatchi so'z bilan boshlanadi, bu so'zdan keyin shu dasturga berilgan nom yoziladi (bu nom dastur ichida hech qanday maqsadda ishlatilishi mumkin emas). Dastur nomidan keyin yumaloq qavs ichida dastur parametrlari ro'yxati beriladi. Bu parametrlar fayllar nomi bo'lib, ular orqali dastur tashqi muhit bilan bog'lanadi. Mavjud bo'lgan ko'p fayllar orasidan 2 ta standart matnli fayllar ajratilgan. Bular dasturni boshlang'ich ma'lumotlar va ma'lumotlarni qayta ishlash natijalari bilan bog'laydigan *INPUT* va *OUTPUT*lardir.

Belgilar bo'limi

Dasturning ixtiyoriy operatori oldiga belgi va ikki nuqta qo'yib qolgan operatorlardan ajratish mumkin. Operatorning belgisi bittadan ortiq bo'lishi mumkin emas. Operatorlarning belgilari bir xil bo'lishi mumkin emas. Operatorning belgisiga uning nomi sifatida xaraladi. Dasturda ishlatilayotgan har bir belgi avval belgilar bo'limida tavsiflash yo'li bilan e'lon qilinishi kerak. Belgilar bo'limi label (belgi) xizmatchi so'z bilan ochiladi, undan keyin belgilar, ya'ni ketma-ket vergul bilan ajratilgan konstanta shaklida yoziladi. Belgilar bo'limida e'lon qilinayotgan belgilar ixtiyoriy ketma-ketlikda yozilishi mumkin. Belgilar (0-9999) orasidagi sonlardan olinadi. Turbo Paskalda belgi sifatida alfavitning boshqa elementlarini ham ishlatish mumkin.

Konstantalar-o'zgarmlar bo'limi

Konstanta deganda aniq bir u yoki bu turdagi qiymatni tushunamiz, dastur ishlash jarayonida bu qiymat o'zgarmaydi shu sababli o'zgarmlar deb ham yuritiladi. Paskal tilida konstantalarga nom beriladi. Konstantalar bo'limi **const** xizmatchi so'z bilan boshlanadi va nuqta vergul (;) bilan tugaydi.

Misol: const C=34; D=6.3;

Toifalar bo'limi

Turbo Paskal tilida to'rtta standart turdagi qiymatlar bor, bular:

integer (butun)

real (haqiqiy)

char (matnli)

boolean (mantiqiy)

Bu turdagi qiymatlar bilan bir qatorda, Paskal tilida boshqa turdagi qiymatlarni ham ishlatish mumkin. Lekin standart turlardan farqli ravishda boshqa turlar dasturda aniq tavsiflanishi kerak.

Turlar bo'limi type (tur) xizmatchi so'zi bilan boshlanadi, keyin turlar tavsifi yoziladi. Turlar tavsifi bir-biridan nuqta vergul (;) orqali ajratiladi.

Protseduralar va funktsiyalar bo'limi

Boshqa bo'limlardan farqli ravishda, bu bo'lim maxsus xizmatchi so'z bilan belgilanmaydi. Bu bo'limning boshi Procedure yoki Function so'zi bilan boshlanadi. Protsedura tavsifi har doim funktsiya tavsifidan oldin turadi. Bu bo'lim dasturda standart protsedura va funktsiyalardan tashqari, protsedura va funktsiyalar ishlatilayotgan bo'lsa va agar ularga dasturdan murojaat qilinayotgan bo'lsa ishlatiladi.

Operatorlar bo'limi

Bu bo'lim dasturning asosiy bo'limi hisoblanadi. Dasturni bajarish bo'limi operatorlik xizmatchi so'zlari begin va end ichiga joylashgan operatorlar ketma-ketligini bajarishga keltiriladi. Operatorlar bir-biridan nuqtali vergul (;) yordamida ajratiladi.

Shuni aytish kerakki, protsedura va funktsiya bo'limidagi algoritmlar faqat operatorlar bo'limidan murojaat qilinganda bajariladi. Operatorlar yordamida masalani echish algoritmini bajarishga kerak bo'lgan amallar tavsiflanadi. Funktsional vazifalari bo'yicha Paskal tilidagi operatorlar quyidagi guruhlarga bo'linadi: o'zlashtirish, kiritish-chiqarish, boshqarish. O'zlashtirish operatorlari-dasturda o'zgaruvchilarga ma'lum qiymatlarni o'zlashtirish uchun ishlatiladi. Kiritish-chiqarish operatorlari - tezkor xotiraga boshlang'ich ma'lumotlarni kiritadilar va yozuvga chiqaruvchi qurilmalarga hisoblashlar natijasini chiqaradilar. Boshqarish operatorlari - dastur operatorlarini ketma-ket bajarilishini tashkil etadi.

Chiziqli hisoblash jarayonini algoritmlash va dasturlash

Shuni eslatib o'tish lozimki, dastur bu masalaning echish jarayonini ifodalovchi, ma'lum bir tilda ifodalangan algoritmlar bo'lib, kompyuter bajaruvchi vosita hisoblanadi. Paskal tilida yozilgan dastur to'g'ridan-to'g'ri kompyuterga emas, balki biror bajaruvchiga mo'ljallanganiki, unga qanday toifadagi qiymatlar bilan ishlay olishi va qo'shimcha ko'rsatmalarsiz qanday amallarni bajara olishi belgilab berilgan bo'lishi kerak. U yoki bu masalani echish uchun kattaliklar ustida bajarilishi kerak bo'lgan amallarni ko'rsatish uchun algoritmik tilda operator tushunchasi xizmat qiladi. Dasturdagi har bir operator ma'lumotlarga ishlov berishning mustaqil, mantiqan tugallangan bosqichini ifodalaydi. Paskalda operatorlarning 8 ta turi ko'zda tutilgan. Har qanday masalani echish aniq qoidalar bo'yicha, berilgan kattaliklardan boshqa kattaliklarni keltirib chiqarish jarayonidan iboratdir.

Ma'lumotlarni kiritish va chiqarish

Ko'p oddiy algoritmlarni dasturlashtirishda asosan o'zlashtirish, kiritish-chiqarish operatorlaridan foydalaniladi. Masalani echish jarayoni qator bajariluvchi bosqichlarga bo'linib ketadi. Bu bosqichlarning har birida ma'lum qiymatlar bo'yicha yangi qiymatlar hisoblanadi. Yangi qiymatlarni hisoblash

uchun ifoda tushunchasi xizmat qiladi, har bir ifoda bitta qiymatni hisoblash qoidasini belgilaydi. Hisoblangan qiymatni hisoblash jarayonining keyingi bosqichida foydalanish uchun eslab qolish zarur, bunday eslab qolish hisoblangan qiymatni ma'lum o'zgaruvchiga o'zlashtirish yo'li bilan amalga oshiriladi. Bunday amalni bajarish, asosiy operatorlardan biri hisoblanuvchi qiymat berish, ya'ni o'zlashtirish operatori bilan bajariladi. O'zlashtirish operatorining bajarilishida «: =>» belgisining o'ng tomonidagi ifodaning qiymati hisoblanib, belgining chap tomonidagi o'zgaruvchi tomonidan o'zlashtiriladi. Shunday qilib, o'zlashtirish operatori mustaqil mantiqiy tugallangan hisoblash jarayonini ifodalaydi: o'zlashtirish operatori bajarilishi natijasida ma'lum o'zgaruvchilar keyingi bosqichlarda ishlatilishi mumkin bo'lgan yangi joriy qiymatni qabul qiladi.

Ma'lumotlarni kiritish va chiqarish

Dasturda ma'lumotlarning qiymatlarini xotiraga kiritishni bir necha usullarda bajarish mumkin. Sonli o'zgaruvchilarga ularning qiymatini erishda o'zlashtirish operatoridan foydalaniladi. Dasturni o'zgaruvchilarning turli qiymatlarida bajarish uchun *READ* - kiritish operatori mo'ljallangan. O'zgaruvchilarga qiymatlar turiga mos ravishda klaviaturadan kiritiladi. Aytaylik, *A*, *B*, *C* o'zgaruvchilarga dastur bajarilishi davomida quyidagi qiymatlarni berish kerak bo'lsin: $A=15$, $B=11$, $S=2.3$. Operator *READ* (*A*, *B*, *C*) ko'rinishiga ega bo'lib, sonlar qiymatlarini dastur bajarilishi davomida quyidagicha kiritish mumkin: 5 17 6.2 [Enter]. Agar o'zgaruvchi *REAL* toifada aniqlangan bo'lsa, uning qiymatini butun son yoki haqiqiy son ko'rinishida kiritiladi. Kompyuterning o'zi butun sonni haqiqiy songa o'tkazib oladi. *READLN* - bu operator kiritish jarayonida bo'sh qator qoldiradi; Bu operator oldingi ikki operatorga teng kuchlidir. Kompyuter xotirasidagi ma'lumotlarni displey ekraniga chiqarish operatori - *WRITE* dir. Operator quyidagi bir nechta ko'rinishlarda ishlatilishi mumkin: *WRITE* oddiy o'zgaruvchilar, o'zgaruvchilar yoki ifodalar bo'lishi mumkin va ular standart *OUTPUT* protsedura fayliga chiqariladi.

“*B* ning qiymati =” va undan so'ng *B* o'zgaruvchining qiymatini chiqaradi. *WRITE* operatorida butun va haqiqiy sonlarni ma'lum formatda chiqarish mumkin. Bu format ikki nuqta orqali o'zgaruvchidan so'ng ko'rsatiladi. Masalan: *WRITE(Y:4:2)*; operatori bilan *Y* ning qiymatini chiqarishda, *Y* ning hamma qiymatini chiqarish uchun 4 ta xona ajratilishi, ulardan ikkitasi kasr qismi uzunligini anglatadi (bunda sonning butun va kasr qismini ajratuvchi vergul (nuqta) ham hisobga olinishi zarur).

Tarmoqlanuvchi hisoblash jarayonini algoritmlash va dasturlash.

Boshqaruvni uzatish operatorlari

Ko'pgina masalalarni echishda ba'zi bir jarayonlar ma'lum shart yoki shartlarning qo'yilishiga nisbatan bajariladi. Bunday jarayonlar tarmoqlanuvchi jarayonlar deb yuritiladi. Tarmoqlanuvchi hisoblash jarayonlari oddiy va murakkab bo'lishi mumkin. Bu esa jarayondagi tarmoqlar soniga bog'liq. Ma'lum bir tarmoqlanuvchi jarayon tarkibida yana tarmoqlanishlar bo'lishi mumkin. Bunday tarmoqlanishlari bor bo'lgan hisoblash jarayonlari murakkab tarmoqlanuvchi

hisoblash jarayonlari deb ataladi. Paskal tilida tarmoqlanuvchi jarayonlarni dasturlash uchun shartsiz, shartli o'tish va tanlash operatorlaridan foydalaniladi.

Shartsiz o'tish operatori

Shartsiz o'tish operatori. Dasturda ba'zi bir hollarda boshqaruvni to'g'ridan-to'g'ri biron-bir operatorga uzatishga, ya'ni dasturning bajarilish ketma-ketligini buzishga to'g'ri keladi. Bu jarayon shartsiz o'tish operatori yordamida bajariladi. Shartsiz o'tish operatorining umumiy ko'rinishi quyidagicha: *GOTO < operator belgisi>*; Bu erda operator belgisi boshqaruv uzatiladigan operator belgisidir. Belgi sifatida 0-9999 oraliqdagi natural sonlar va *CHAR* turidagi belgilar ishlatiladi. Belgi tavsiflash bo'limining *LABEL* bo'limida albatta tavsiflangan bo'lishi shart.

Shartli o'tish operatori

Dasturda boshqaruvni ma'lum shart asosida u yoki bu tarmoqqa uzatish shartli o'tish operatori yordamida amalga oshiriladi. Shartli o'tish operatori ikki xil o'rinishda ishlatilishi mumkin: to'liq va qisqa. Shartli o'tish operatorining to'liq ko'rinishi:

IF <mantiqiy ifoda> THEN S1 ELSE S2;

Bu erda *IF* (agar), *THEN* (u holda) va *ELSE* (aks holda) degan xizmatchi so'zlar, *S1* va *S2* ixtiyoriy operatorlar. Operatoridagi mantiqiy ifoda boshqaruvni uzatish shartini belgilaydi. Operatorning ishlash tartibi quyidagicha: Agar keltirilgan mantiqiy ifoda *TRUE* (rost) qiymatni qabul qilsa, ya'ni qo'yilgan shart bajarilsa, *THEN* - xizmatchi so'zidan keyingi operator bajariladi, aks holda *ELSE* xizmatchi so'zidan keyingi operator bajariladi. Mantiqiy ifodalarda munosabat amallari, mantiqiy amallar ishlatilishi mumkin.

Shartlar oddiy va murakkab bo'lishi mumkin. Agar mantiqiy ifodada bitta munosabat amali berilgan bo'lsa, «oddiy shart» ni ifodalaydi. Kattaliklar orasidagi shartlar *HAM*, *YOKI*, *EMAS* (Paskal tilida *AND*, *OR*, *NOT*) mantiq amallari belgilari orqali bog'lanuvchi bir necha munosabatlardan iborat bo'lsa „murakkab shartlar" deb ataladi. *OR* amalining natijasi rost bo'lishi uchun argumentlardan birining rost bo'lishi etarli. *NOT* amalining natijasi argumentning inkor qiymatiga teng, ya'ni argument rost bo'lsa - natija yolg'on, argument yolg'on bo'lsa - natija rost bo'ladi.

Shuni ta'kidlab o'tish kerakki, agar mantiqiy ifodalar, biz yuqorida aytganimizdek, mantiqiy amallar yordamida (*AND*, *OR*, *NOT*) murakkab ko'rinishga ega bo'lsa, ular qavslarga olib yoziladi. Ayrim algoritmlarda ba'zan shunday xol uchrashi mumkinki, bunda hisoblash jarayonida ayrim amallar ba'zi bir shartlar bajarilgandagina hisoblanadi, aks holda, hech qanday amal bajarilmaydi.

Tanlash operatori

Juda ko'p tarmoqlanish jarayonlarida tarmoqlanish ikki yoki undan ortiq tarmoqqa ajraladi. Umuman olganda, buni bizga tanish shartli o'tish operatori yordamida amalga oshirish mumkin:

IF B1 THEN A1 ELSE

IF B2 THEN A2 ELSE

IF BK THEN AK;

Lekin bu hollarda shartli o'tish operatorlarining yozilishi noqulay. Ko'p hollarda dasturchi uchun shartli operatorning umumiy qilishgan ko'rinishi - tanlash (variant) operatorini ishlatish qulay.

Tanlash operatorining metaformulasi quyidagicha yoziladi:

< tanlash operatori> := CASE <operator selektori> OF < tanlash ruyxati elementi>; END

Bunda tanlash operatorining umumiy ko'rinishi:

CASE S OF

M1: A1;

M2: A2;

Mp: An

END;

Bu erda *CASE* (tanlash) -xizmatchi so'z, *OF* (dan), *S* - selektor, *Mi* – operatorlar belgilari, *Ai* –operatorlar. *CASE* operatori tarmoqlanish jarayonini berilgan bir necha operatorlardan birini tanlash yo'li bilan amalga oshiradi. Tanlash operatorida barcha operatorlar, shu jumladan bajarilishi uchun tanlangan operator ham aniq ravishda keltiriladi (berilgan operatorlar ketma-ketligi chegaralangan). Bajarilishi kerak bo'lgan operator yoki operatorlar ketma-ketligi operator selektorining qiymatiga ko'ra aniqlanadi. Operator selektori sifatida haqiqiy bo'lmagan, skalyar ko'rinishdagi har qanday ifoda yoki o'zgaruvchi ishlatilishi mumkin.

Operatorning ishlashida uning tarkibidagi har bir operator tanlash belgisi deb ataluvchi belgi bilan ta'minlanadi. Bu belgi operatorning bajarilishi uchun zarur bo'lgan selektorning maxsus qiymatini qabul qiladigan selektorning tavsifiga mos konstantadir. Operator bir necha mavjud qiymatlar bilan ishlashi uchun, unda tanlash belgilari ro'yxati keltirilishi kerak. Tanlash operatoridagi belgili operatorlar oddiy belgiga ham ega bo'lishlari mumkin. Bu holda oldin tanlash belgilari, so'ngra oddiy belgilar yoziladi. Shuni ham inobatga olish lozimki, tanlash operatoriga faqat *CASE* xizmatchi so'z orqali kirish mumkin, ya'ni tanlash operatoridan tashqaridagi o'tish operatori orqali bu operatorga murojaat qilish mumkin emas. Tanlash operatorining bajarilishi uning tarkibidagi operatorlar ketma-ketligidagi bitta operatorning bajarilishiga olib keladi. Shuning uchun ularning biridan biriga *GOTO* operatori yordamida o'tish xato demakdir.

Takrorlanuvchi hisoblash jarayonlarini algoritmlash va dasturlash

Amaliyotda murakkab jarayonlarni dasturlashda ma'lum buyruqlar ketma-ketligini ma'lum shartlar asosida qayta-qayta bajarish zaruriyati tug'iladi. Ma'lum bir o'zgaruvchining turli qiymatlarida ma'lum buyruqlar tizimining biron-bir qonuniyatga asosan qayta-qayta bajarilishi takrorlanuvchi hisoblash jarayoni - sikl deb ataladi. Takrorlanuvchi hisoblash jarayonining takror-takror hisoblanadigan qismini takrorlanishning tanasi (jismi) deb ataladi. Takrorlanish ichida qiymatlari o'zgarib boradigan o'zgaruvchini takrorlanish o'zgaruvchisi yoki takrorlanishni boshqaruvchi o'zgaruvchisi (sikl parametri) deb yuritiladi. Takrorlanuvchi jarayonning algoritmi umumiy holda quyidagilarni o'z ichiga olishi kerak:

1. Takrorlanishni tayyorlash - takrorlanishni boshlashdan oddin takrorlanishda qatnashadigan o'zgaruvchilarning boshlang'ich qiymatlari yoki takrorlanish o'zgaruvchisining boshlang'ich qiymati o'rnatiladi, takrorlanish o'zgaruvchisining o'zgarish qadami belgilanadi.

2. Takrorlanish tanasi - takrorlanish o'zgaruvchilarining turli qiymatlari uchun takror bajariladigan amallar ketma-ketligi ko'rsatiladi.

3. Takrorlanish o'zgaruvchisiga yangi qiymat berish - har bir takrorlanishdan avval o'zgaruvchiga o'zgarish qadamiga mos ravishda yangi qiymat beriladi.

4. Takrorlanishni boshqarish - takrorlanishni davom ettirish sharti tekshiriladi, takrorlanishning boshiga o'tish ko'rsatiladi.

Paskal algoritmik tilida uch xil ko'rinishda takrorlanuvchi hisoblash jarayonini tashkil qilish mumkin va bu jarayonlarni dasturlash uchun maxsus operatorlar belgilangan:

- sharti avval tekshiriladigan takrorlanish jarayoni, bu maxsus WHILE operatori orqali amalga oshiriladi;

- sharti keyin tekshiriladigan takrorlanish jarayoni, bu jarayonni maxsus REPEAT operatori orqali amalga oshiriladi;

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

Sharti avval tekshiriladigan takrorlanish jarayoni

Takrorlanuvchi jarayonning bu ko'rinishi takrorlanish soni oldindan noma'lum bo'lgan hollarda, ya'ni takrorlanishdan chiqish ma'lum shartga bog'liq bo'lgan hollarda ishlatiladi. Takrorlanishning bu jarayonida takrorlanishdan chiqish sharti takrorlanish tanasini bajarishdan oldin tekshiriladi. Ushbu operatorning umumiy ko'rinishi quyidagicha: *WHILE L DO M*; bu erda, *WHILE* - toki, *do* - bajarish ma'nosini anglatuvchi xizmatchi so'zlar, *L* - mantiqiy ifoda, *M* - operatorlar yoki operatorlar guruhi, u takrorlanish tanasini belgilaydi. Takrorlanish tanasida bitta yoki bir necha operatorlar guruhi bo'lishi mumkin. Bunda operatorlar guruhi, albatta, **begin va End** orasida yozilishi kerak. Operatorning bajarilishi quyidagicha: *L* mantiqiy ifodaning qiymati hisoblanadi. Agar *L* mantiqiy ifoda rost qiymatga ega bo'lsa, *M* operatori bajariladi va bu operator *L* mantiqiy ifodaning qiymati yolg'on bo'lgungacha qayta-qayta bajariladi. Agar *L* mantiqiy ifodaning qiymati birinchi tekshirishdayoq yolg'on bo'lsa, *M* operatori biron marta ham bajarilmaydi va boshqaruv *WHILE* operatoridan keyingi operatorga uzatiladi.

Sharti keyin tekshiriladigan takrorlanish jarayoni

Takrorlanish jarayonining bu ko'rinishi ham takrorlanish soni oldindan ma'lum bo'lmagan hollarda ishlatiladi. Bunday jarayonda biror buyruq yoki buyruqlar tizimi berilgan shart bajarilgunga qadar takror va takror bajariladi. Bu jarayonni avvalgisidan farqi shundaki, bunda takrorlanishning tanasi hech bo'lmaganda bir marta bo'lsa ham bajariladi, chunki takrorlanishdan chiqish sharti takrorlanishning tanasi bajarilgandan keyin tekshiriladi. Bunday takrorlanish

jarayonini dasturlashda maxsus REPEAT operatoridan foydalaniladi. Ushbu operatorning umumiy ko'rinishi quyidagicha: *REPEAT M UNTIL L*; bu erda *REPEAT* - takrorlamoq, *UNTIL* - «gacha» ma'nosini anglatuvchi xizmatchi so'zlar; *M* - takrorlanish tanasi, uning tarkibida bitta operator yoki operatorlar guruhi bo'lishi mumkin. Operatorlar guruhini Begin va End orasiga olish shart emas. Operator quyidagi tartibda bajariladi: takrorlanish tanasiga kirgan operatorlar birin-ketin bajariladi. So'ng *L* mantiqiy ifodaning qiymati topiladi, ya'ni shart tekshiriladi. Agar bu shart bajarilsa (*L* ning qiymati TRUE (rost) bo'lsa) boshqaruv takrorlashdan tashqariga, *UNTIL* so'zidan keyingi operatorga uzatiladi. Aks holda takrorlanish davom etadi. Takrorlanish jarayonining bunday ko'rinishi yuqorida keltirilgan takrorlanishdan shu bilan farq qiladiki, bunda takrorlanish tanasi hech bo'lmaganda bir marta bajariladi.

Parametrli takrorlanish jarayoni

Yuqorida keltirilgan takrorlanish operatorlarini, odatda, takrorlanish soni noma'lum bo'lgan hollarda ishlatish maqsadga muvofiqdir. Takrorlanish jarayonining takrorlanish soni uni bajarilishidan oldin ma'lum bo'lsa hamda takrorlanish o'zgaruvchisining qiymatlari orasidagi farq (o'zgarish qadami) 1 ga yoki -1 ga teng bo'lsa, parametrli takrorlanish operatoridan foydalanish qulay. Bu buyruq quyidagi ko'rinishga egadir:

FOR V: = L1 TO (DOWNTO) L2 DO M

Bu erda: *FOR* operatorning nomi; *V* - takrorlanish o'zgaruvchisining nomi; *L1*, *L2* - mos ravishda *V* ning boshlang'ich va oxirgi qiymati; *TO* - "gacha" ma'nosini anglatuvchi xizmatchi so'z; *DOWNTO* - *V* ning qiymati -1 ga o'zgarganda *TO* ning o'rnida ishlatiladi; *DO* - bajarilsin degan ma'noni anglatadi; *M* - takrorlanish tanasi. Tanada bir necha operator bo'lsa, ular Begin va End xizmatchi so'zlari orasida yoziladi. Takrorlanish o'zgaruvchisining toifasi REAL va cheklanmagan INTEGER toifasidan tashqari, barcha oddiy toifalarda berilishi mumkin va uning qiymatlari orasidagi farq bitta kattalikka teng. *L1* va *L2* *V* ning toifasidagi kattaliklar bo'lib, uning o'rnida o'zgaruvchi nomlarini, ifodalarni va o'zgarmlarni ishlatish mumkin.

Murakkab takrorlanish jarayonlari

Yuqorida keltirilgan operatorlar asosida murakkab takrorlanishlar tashkil qilish mumkin. Agar takrorlanish jarayonlarining tanasi takrorlanish strukturasi tashkil topgan bo'lsa, u holda bunday takrorlanish ichma-ich joylashgan yoki murakkab deb ataladi, ya'ni boshqacha qilib aytganda, bitta takrorlanish ichiga bir yoki bir necha boshqa takrorlanishlar kirsa murakkab tarkibli takrorlanishli dasturlar hosil bo'ladi. Boshqa takrorlanish jarayonlarini o'z ichiga olgan takrorlanish tashqi takrorlanish deb ataladi. Takrorlanish jarayonlarini ichida joylashgan takrorlanish ichki takrorlanish deb ataladi.

Ma'lumotlarning qayd qilingan toifasi

Biz shu vaqtgacha dasturlarda ma'lumotlarning standart toifalari bilan ishlab keldik. Bu toifalardagi qiymatlar diapazoni Paskal tili tomonidan belgilanadi. Ma'lumotlarning toifasi shu toifadagi kattaliklarning qabul qilishi mumkin bo'lgan

qiymatlar to'plamini belgilaydi. Ma'lumotlarning toifasini belgilash, o'z navbatida, bu toifadagi kattaliklar ustida bajariladigan amallarni ham belgilaydi. Masalan, haqiqiy toifadagi kattaliklar ustida qo'shish, ayirish, ko'paytirish va bo'lish amallarini bajarish mumkin, lekin *DIV*, *MOD* amallarini bajarish mumkin emas. Lekin Paskal algoritmik tili dastur tuzuvchiga yangi, nostandart toifalarni kiritish, ya'ni ma'lumotlarning o'ziga qulay bo'lgan qiymatlarini ishlatish imkoniyatini beradi. Bularga oddiy o'zgaruvchi toifaga kiruvchi qayd qilingan, chegaralangan va tarkibiy toifalar kiradi. Ma'lumotlarning qayd qilingan toifasidagi qiymatlar cheklangan sonda va qat'iy tartibda qayd etish, keltirish yo'li bilan beriladi. Bu toifadagi o'zgaruvchilar faqat shu toifada keltirilgan qiymatlarning birini qabul qilishi mumkin. Odatda, qayd qilingan toifa qiymatlari echilayotgan masalaning mazmuniga ko'ra belgilanadi. Qayd qilingan toifalarni har doimgidek, *TYPE* (toifalarni tavsiflash) va *VAR* (o'zgaruvchilarni tavsiflash) bo'limida tavsiflash mumkin.

Ma'lumotlarning chegaralangan toifasi

Ba'zi bir hollarda o'zgaruvchi o'zi aniqlangan toifaning hamma qiymatlarini emas, balki ma'lum chegara (diapazon) dagi qiymatlarinigina qabul qilishi mumkin. Bu hollarda o'zgaruvchi uchun chegaralangan toifa belgilash mumkin. Chegaralangan toifa oldindan aniqlangan toifaga cheklanish qo'yish bilan beriladi. Oldindan aniqlangan toifa chegaralangan toifaga nisbatan baza toifasi vazifasini bajaradi. Baza toifasi sifatida *REAL* toifasidan tashqari barcha standart toifalarni ishlatish mumkin. Chegaralangan toifa quyidagi ko'rinishda tavsiflanadi: Chegaralangan turdagi o'zgaruvchi baza turining birinchi qiymati va oxirgi qiymatlari orasidagi ixtiyoriy qiymatlarning birini qabul qilishi mumkin.

Ma'lumotlarning murakkab toifasi. Muntazam toifa

Indeks va indeks toifasi tushunchalari o'rtasida o'zaro farq mavjud bo'lib, indeks toifasi massiv elementlari soni va ular o'zaro tartiblanganligini bildiradi va u massivni tavsiflash bo'limida ishlatiladi, indeks esa massiv elementining tartib raqamini belgilaydi va operatorlar bo'limidagina ishlatiladi. Agar biror-bir massivga murojaat qilish uchun uning to'liq nomi, ya'ni muntazam toifaga mansub o'zgaruvchining nomi ishlatilsa, massivning alohida elementiga murojaat qilish uchun indeksli o'zgaruvchi ishlatiladi. Masalan, *A* (4) massiv uchun *A* o'zgaruvchi to'liq o'zgaruvchi (massiv nomi), *A* [1] - indeksli o'zgaruvchi bo'lib, u *A* massivining 1 elementini ifodalaydi.

Massivlarning o'lchami ularni kompyuterda qayta ishlashga hech qanday to'sqinlik ko'rsatmaydi, chunki massiv elementlari ularning o'lchamidan qat'iy nazar kompyuter xotirasida chiziqli ketma-ketlik sifatida saqlanadi. Ko'p o'lchamli massivlar dasturda quyidagicha tavsiflanadi. Masalan *B* o'zgaruvchini to'rtta satr va beshta ustundan iborat (4,5) ikki o'lchamli massiv sifatida quyidagicha tavsiflash mumkin

Foydalanilgan adabiyotlar

1. Informatika va informatsion texnologiyalar, M.Aripov va boshqalar. Oliy o'quv yurti talabalari uchun darslik. Toshkent-2019 y.
2. Axborot texnologiyalari, M.Aripov va boshqalar. Oliy o'quv yurti talabalari uchun o'quv qo'llanma. Toshkent-2019 y.
3. Delphi tilida dasturlash asoslari, Sh.Nazirov. Toshkent-2018 y.