

9 – Mavzu: Turbo Paskal tilida protsedura konstruktsiyasi

Paskal tili protsedurasi

Dasturlash jarayonida murakkab dasturlarning bir necha joyida bir xil vazifani bajaruvchi operatorlar guruhini qo'llashga to'g'ri keladi va dasturda bir-biriga aynan o'xshash bir necha qismlar vujudga keladi. Kompyuter xotirasini va dastur tuzuvchining vaqtini tejash maqsadida ushbu qismdagi o'xshash operatorlar bir marta asosiy dasturdan ajratib yoziladi va unga asosiy dastur bajarilishi jarayonida murojaat qilinadi. Dasturning ixtiyoriy qismidan murojaat qilib, bir necha bor ishlatish mumkin bo'lgan bunday operatorlar guruhiga qism dastur deb ataladi va u asosiy dastur bilan bir butunni tashkil etadi. Qism dasturlarni ishlatish dasturning hajmini kichraytiradi va uning ko'rinishini, o'qilishini hamda xatolar sonining kamayishiga olib keladi.

Turbo Paskal tilida qism dasturlar protsedura va funktsiya qism dasturi ko'rinishida bo'ladi.

Protsedura qism dasturi

Qism dasturda bir necha natija hosil qilish kerak bo'lgan hollarda ishlatiladi va u asosiy dasturda struktura bo'yicha qism dastur bo'limida, ya'ni o'zgaruvchilar va operatorlar bloki bo'limlari orasida tasvirlanadi. Har bir protsedurani tasvirlash sarlavhadan boshlanadi. Bunda protseduraning nomi va parametrlarning ro'yxatlari ularning turlari bilan birga beriladi. Protsedura parametrsiz ham bo'lishi mumkin. Bu holda protsedura sarlavhasida faqat uning nomi ko'rsatiladi. Parametrlar yordamida belgilarni protseduraga berish hamda hosil bo'lgan natijani protsedurani chaqirayotgan dasturga berish global parametrlar yordamida amalga oshiriladi. Protsedura sarlavhasining umumiy ko'rinishi quyidagicha bo'ladi:

PROCEDURE < nom > (< soxta parametrlar ro'yxati >);

bu erda *PROCEDURE* - xizmatchi so'z; <nom>- protseduraning nomi; <formal parametrlar ro'yxati>- turlari ko'rsatilgan berilganlarni va natijalarni ifodalaydigan nomlar turlari bilan sanab o'giladigan qator. Sohta parametrlar ro'yxatida parametr-qiymat, parametr-o'zgaruvchilar (bularning oldida VAR xizmatchi so'zi turishi kerak), parametr-protseduralar va parametr-funktsiyalar (bularning oldida *FUNCTION* xizmatchi so'zi turishi kerak) bo'lishi mumkin. Protsedura sarlavhasidan keyin keladigan bo'limlar asosiy dasturda qanday tartibda kelsa, protsedurada ham shu tartibda keladi. Protsedura bloki nuqta vergul bilan tugaydi.

Protsedura va funktsiya qism dasturiga murojaat

Protsedurani bajarish uchun asosiy dasturning ixtiyoriy (kerakli) joyidan unga murojaat qilish kerak. Protseduraga murojaat va uni bajarish quyidagi operator yordamida amalga oshiriladi: <protsedura nomi> (<aniq parametrlar ro'yxati>); Bunda qism dastur tasvirida keltirilgan protseduraga murojaat qilishda sohta va aniq parametrlar orasida to'liq moslik bo'lishi shart, ya'ni sohta va aniq parametrlar soni bir xil bo'lishi, kelish tartiblari mos tushishlari va ularning turlari bir xil bo'lishi kerak. Protsedura chaqirilganda sohta parametrlar o'rniga aniq parametrlar mos

qo'yiladi. Bunda parametr-qiymatlar o'rniga kiritiladigan malumotlar beriladi. Natija esa parametr-o'zgaruvchilarda hosil bo'ladi. Protsedura chaqirilganda sarlavhada ko'rsatilgan sohta parametrlar aniq parametrlar bilan ularning kelishi tartibida almashtiriladi, ya'ni birinchi sohta parametrga birinchi aniq parametr, ikkinchi sohta parametrga ikkinchi aniq parametr va h.k. mos qo'yiladi. Funktsiya qism dasturiga murojaat uning nomi orqali amalga oshiriladi. Funktsiyaga murojaat qilinadigan joyda funktsiyaning nomi yoziladi va undan keyin qavs ichida aniq parametrlar yoziladi. Agar funktsiya qism dasturi parametrsiz ko'rinishda bo'lsa, u holda faqat uning nomi yoziladi.

Protsedura va funktsiya qism dasturlarining parametrlari

Protsedura va funktsiya qism dasturlari aniqlanayotganda ularning formal parametrlar ro'yxatida parametrlarning to'rt xil ko'rinishi ko'rsatilishi mumkin:

- 1) parametr- qiymatlar;
- 2) parametr- o'zgaruvchilar;
- 3) parametr- protseduralar;
- 4) parametr- funktsiyalar.

Formal va real parametrlar protsedura va funktsiya qism dasturlarida bir xil qayta ishlanadi. Har bir parametr turi ko'rsatilgan holda ro'yxatga kiritiladi.

Agar ikkita yoki undan ko'p sohta parametrlar ro'yxatda bir xil turda bo'lsa, u holda ularni bitta tasvirlashga kiritish mumkin. Sohta parametrlarni ro'yxatda ixtiyoriy tartibda ko'rsatish mumkin. Protsedura va funktsiya qism dasturlariga murojaat qilinayotganda sohta parametrlar qanday tartibda kelsa, aniq parametrlar ham shu tartibda kelishi kerak. Protsedura va funktsiya qism dasturlarini o'z ichiga olgan asosiy dasturda nom tanlanayotganda lokal va global parametrlarning bir xil bo'lmagani maqul. Parametr-qiymatlar parametrlari berilganlarni qism dasturlar (protsedura va funktsiya) ga uzatish uchun ishlatiladi. Bunda yuqorida aytilganidek, ular sohta parametrlar ro'yxatida turlari ko'rsatilgan holda, bir-birlaridan vergul bilan ajratilgan holda sanab o'tiladi.

Turbo Paskal algoritmik tilida fayl tushunchasi

Sizga ma'lumki, fayl deb xotiraning nomlangan sohasiga aytiladi. Faylda turli ma'lumotlar saqlanadi. Xar bir fayl bilan fayl ko'rsatkichi degan tushuncha birlashtirilgan. Fayl bir necha bi turdagi elementlardan iborat bo'lib, foydalanuvchi faqat faylning ko'rsatkichi ko'rsatayotgan ma'lumotga murojaat qilishi mumkin. Demak, fizik jixatdan biz faqat ketma-ket fayllarga egamiz. Ya'ni biz oldin birinchi, keyin ikkinchi, uchinchi va h.k. ma'lumotlarni o'qishimiz mumkin. Fayl o'z nomiga ega.

Turbo Paskal tili dasturiy vositalari yordamida, ya'ni dasturda ham fayllar tashkil qilish va undagi ma'lumotlarni qayta ishlash mumkin. Shu paytga qadar, Paskal dasturiy tilida bir necha o'zgaruvchilarning toifalari bilan ishlab keldik. Bular skalyar, oddiy va murakkab tarkiblashgan toifalardir. Bu toifadagi ma'lumotlar yordamida masalalarni echishda boshlang'ich ma'lumotlar klaviaturadan operativ xotiraga kiritiladi va natija ekranga chiqariladi. Ulardan boshqa dasturlarda foydalanib bo'lmaydi, chunki ular tizimidan chiqilgandan so'ng xech qaerda

saqlanmaydi. Bu ma'lumotlarni xotirada saqlash uchun Turbo Paskal tilida ma'lumotlarning faylli toifasi belgilangan. Fayl toifasi aloxida o`rin egallaydi. Fayl toifasi bilan ishlashda ma'lum tushunchalarni o`zlashtirish talab qilinadi.

1. Juda ko`p o`zgaruvchilardan foydalanganda ularning qiymatlarini xar doim klaviaturadan kiritishda ma'lum noqulayliklarga duch kelamiz. Bunga katta massivlar misol bo`la oladi.

2. Shunday masalalar uchraydiki, oldindan kattaliklarning qiymatlar soni noma'lum bo`ladi (masalan, natijalar), bu kattaliklarni faylga yozish maqsadga muvofiq.

3. Xech qanday toifalar tashqi qurilmalarga murojaat qilib, ular bilan ishlashga imkon yaratmaydi (dasturiy til muxitida) Va nixoyat, boshqa toifalardan fayl toifasi farqliligi shundaki, u boshqa toifalar tarkibiga kira olmaydi.

Fayllarning turlari

Fayllar uchun mo`ljallangan umumiy protsedura va funktsiyalar Faylda saqlanayotgan ma'lumotlar turiga ko`ra, Paskal algoritmik tilida fayllar quyidagi turlarga bo`linadi:

- 1) toifalashmagan;
- 2) toifalashgan;
- 3) matnli.

Toifalashgan fayllar

Toifalashgan fayllar bir xil toifali elementlardan tashkil topadi. Ularni faqat ma'lum qurilmalarda uzatish mumkin, lekin ekranda o`qish mumkin emas. Faylning elementlari mashina kodlarida yoziladi va saqlanadi. **Toifalashmagan fayllarda** turli toifadagi ma'lumotlarni saqlash mumkin. Ular xam mashina kodlari bilan yozilgan bo`lib baytlar to`plamini tashkil qiladi. **Matnli (tekst) fayllar** ASCII kodlardan tashkil topgan va qatorlarga ajratilgan bo`ladi. Matnli fayllarda nafaqat faylning yakunida fayl oxiri belgisi, balki xar qatorning yakunida maxsus qator oxiri belgisi qo`yiladi.

Fayl turidagi o`zgaruvchi fayl o`zgaruvchisi deyiladi, u faylning mantiqiy nomini belgilaydi va u mantiqiy fayl bilan tashqi (fizik) fayl o`rtasida «vositachi» vazifasini o`ynaydi.

Fayl turi uchun arifmetik amallar belgilanmagan. Xatto fayllarni solishtirish va bir faylning qiymatini ikkinchi faylga o`zlashtirish amallari xam aniqlanmagan. Xar bir turdagi fayllar ustida, umuman olganda, quyidagi amallarni bajarish mumkin va bu amallar uchun maxsus protsedura va funktsiyalar ishlatiladi:

1. Turbo Paskal muxitida fayl bilan ishlashdan oldin faylning fizikaviy va mantiqiy nomlarini bog`lash lozim. Bu aloxida protsedura yordamida amalga oshiriladi. Bu protseduraning ma'nosi shundaki, u fayl uchun yo`l ochib dasturdan tashqi qurilmaga murojaat qilishi va axborot almashtirish imkonini yaratib beradi.

2. Faylga ma'lumot yozish uchun faylni ochish. Buning uchun quyidagi protsedura ishlatiladi: Rewrite (<fayl o`zgaruvchisi>);

Bu protsedura bajarilganda xotirada Assign protsedurasida ko`rsatilgan nom

bilan yangi fayl unga ma'lumot yozish uchun ochiladi va fayl ko`rsatkishi faylning boshiga o`rnatiladi. Lekin bu protsedurani extiyotkorlik bilan ishlatish kerak, chunki ko`rsatilgan fayl oldindan xotirada bo`lsa, undagi ma'lumotlar butunlay o`chirib tashlanadi.

3. Faylni undan ma'lumotlarni tezkor xotiraga o`qish uchun ochish. Buning uchun quyidagi protsedura ishlatiladi: *Reset* (<fayl o`zgaruvchisi>);

Reset protsedurasi bajarilganda Assign protsedurasida ko`rsatilgan fayl undagi ma'lumotlarni o`qish uchun tayyorlanadi, ya'ny fayl ko`rsatkishi faylning birinchi elementiga keltirib qo`yiladi.

4. Faylga ma'lumotlarni yozish, kiritish. Buning uchun bizga tanish bo`lgan Write operatori quyidagi ko`rinishda ishlatiladi:

Write (<fayl o`zgaruvchisi>, <kattalik>);

Bu erda kattalik o`rnida o`zgaruvchi yoki ifoda ishlatilishi mumkin. Protsedura bajarilganda kattalikning qiymati fayl o`zgaruvchisi bilan bog`langan faylda fayl ko`rsatkishi o`rnatilgan joyga yoziladi. So`ngra fayl ko`rsatkishi bitta keyingi pozitsiyaga suriladi. *Write* protsedurasini *Rewrite* protsedurasi bajarilgandan keyingina ishlatish mumkin.

5. Fayldan ma'lumotlarni o`qish. Bu maqsadda quyidagi protseduradan foydalaniladi:

Read (<fayl o`zgaruvchisi>, <o`zgaruvchi>);

Bu protsedura bajarilganda Reset protsedurasi bilan ochilgan faylda fayl ko`rsatkichi o`rnatilgan joydagi elementning qiymati protseduradagi o`zgaruvchiga o`zlashtiriladi. So`ngra fayl ko`rsatkishi yana bitta pozitsiyaga suriladi.

6. Turli maqsadda ochilgan barcha mantiqiy fayllar albatta yopilishi kerak. Buning uchun quyidagi protsedura mo`ljallangan:

Close (<fayl o`zgaruvchisi>);

Bu protsedura bajarilganda axborot uzatishning barcha kanallari yopiladi.

7. Fayl oxirini aniqlash funktsiyasi:

Eof (<fayl o`zgaruvchisi>);

Bu funktsiyaning qiymati Boolean toifasida bo`lib, u fayl ko`rsatkishi faylning oxiriga o`rnatilganda True qiymatiga ega bo`ladi, aks xolda uning qiymati False ga teng. Fayllarning ixtiyoriy ko`rinishi bilan ishlashda shuni nazarga olish kerakki, bir vaqtning o`zida bitta fayldan unga ma'lumot yozish uchun va undan ma'lumotlarni o`qish uchun foydalanib bo`lmaydi. O`qish uchun yoki yozish uchun ochilgan fayl albatta Close protsedurasi yordamida yopilgan bo`lishi shart. Endi turli ko`rinishdagi fayllar bilan yaqindan tanishib chiqamiz.

Matnli fayllar uchun mo`ljallangan funktsiyalar

Yuqorida ta'kidlab o`tilganidek, matnli fayl qatorlardan tashkil topgan fayldir. Matnli fayllarda:

- a) ma'lumot matn shaklida ASCII kodlar jadvalining simvollarida tasvirlanadi;
- b) ma'lumotlar qatorlarga bo`linishi mumkin;
- c) sonlar, mantiqiy qiymatlar, qatorlar Char toifasidagi ma'lumotlarga aylantiriladi va mashina kodlariga yoziladi. Dasturda matnli fayllar Text xizmatchi so`zi yordamida beriladi.

Turbo-Paskal muxitida ikkita matnli fayl o`zgaruvchilari standart ravishda aniqlangan. Bu *INPUT* va *OUTPUT* o`zgaruvchilaridir. Dasturda ma'lumotlarni kiritish va chiqarish aynan shu fayllar orqali amalga oshiriladi, lekin dastur sarlavhasida va tavsiflash bo`limida ularni e'lon qilish shart emas. Bu fayllar Turbo - Paskal tizimi yuklanishi bilan avtomatik tarzda tezkor xotiraga yuklanadi. Umumiy fayl o`zgaruvchisiga mansub protseduralardan tashqari, matnli fayllarga quyidagi funktsiya va protseduralar mo`ljallangan:

1. *Readln* (*Fayl o`zgaruvchisi*), *<o`zgaruvchi>*) - fayldan simvollar qatorini o`qish protsedurasi. Bu protsedura bajarilganda fayl ko`rsatkichi o`rnatilgan qator o`zgaruvchiga o`zlashtiriladi, faylda qator «qator oxiri belgisi» yordamida ajratib olinadi.

2. *Writeln* (*<fayl o`zgaruvchisi>*, *<qator>*); - faylga simvollar qatorini yozish protsedurasi. Protседura bajarilganda fayl ko`rsatkichi o`rnatilgan joyga *<qator>* yoziladi.

3. *Append* (*<fayl o`zgaruvchisi>*); protsedurasi. Bu protsedura faylni unga qo`shimcha elementlarni (ma'lumotlarni) yozish uchun ochadi. Bu protsedura xotirada saqlangan fayllar uchun ishlatilib, Rewrite protsedurasi o`rnida keladi.

4. *Eoln* (*<fayl o`zgaruvchisi>*); funktsiyasi fayldagi joriy qatorning oxirini aniqlaydi. Funktsiyaning toifasi Boolean (mantiqiy) bo`lib, fayl ko`rsatkichi qator oxiriga o`rnatilganda funktsiyaning qiymati True ga, aks xolda False ga teng bo`ladi.

5. *SeekEoln* (*<fayl o`zgaruvchisi>*); funktsiyasi qator yakuniga erishilganini aniqlaydi.

6. *SeekEof* (*< fayl o`zgaruvchisi>*); funktsiyasi faylni yakuniga erishilgan yoki erishilmaganligini aniqlaydi.

Eof va *SeekEof*, *Eoln* va *SeekEoln* funktsiyalarining farqi shundaki, *Eof* va *Eoln* fizikaviy faylga nisbatan, *SeekEof* va *SeekEoln* esa mantiqiy faylga nisbatan qo`llaniladi.

Toifalashmagan fayllar

Turbo-Paskal dasturiy tilida aloxida axamiyatga ega bo`lgan fayllardan, ya'ni toifalashmagan fayllardan foydalanish mumkin. Bu fayllarni umumlashgan toifa deb atasak yanglishmaymiz. Faylning toifalashmagan deb atalishidan maqsad, fayl turli toifadagi ma'lumotlardan tashkil topadi. Toifalashmagan fayllarni tavsiflashda elementlar toifasi ko`rsatilmaydi, faqat File xizmatchi so`zidan foydalaniladi: *Var <fayl o`zgaruvchisi>: File;* Toifalashmagan fayllarga nisbatan ma'lumotlarni kiritish, ma'lumotlar faylini yaratish, ma'lumotlarni fayldan o`qish, tezkor xotirada fayl elementlarini qayta ishlash kabi amallarni bajarish mumkin. Fayldagi elementlar toifasi oldindan ma'lum bo`lmagani uchun undagi ma'lumotlar bir xil uzunlikdagi bloklarga (yozuvlarga) ajratilib o`qiladi va shu tarzda faylga yoziladi. Blok uzunligi baytlarda olinadi. Shundan kelib chiqqan xolda bu ko`rinishdagi fayllarga nisbatan quyidagi protsedura va funktsiyalarni ishlatish mumkin:

1. *Reset* (*F*, *S*); protsedurasi faylni o`qish uchun ochadi (yuqorida berilgan Reset protsedurasi o`rnida ishlatiladi), bu erda *F* - fayl o`zgaruvchisi, *S*-xar bir blok uchun belgilangan xotira xajmi (baytlarda olinadi).

2. *BlockRead* (F, V, N); protsedurasi, bu erda F - fayl o`zgaruvchisi, N - o`qilishi lozim bo`lgan bloklar soni (Integer), V - o`qilgan bloklar joylashtiriladigan xotiradagi birinchi adres raqami (Integer, Word). Bu protsedura bajarilganda F va V da joylashgan S uzunlikdagi N bloklar o`zlashtiriladi.

3. *Rewrite* (F, S); protsedurasi F fayliga S uzunlikdagi yozuvlarni yozish uchun faylni ochadi.

4. *BlockWrite* (F, V, N); protsedurasi F fayliga tezkor xotiraning V adresli joydagi N ta yozuvni joylashtiradi.

5. *FilePos*(F) funktsiyasi joriy blokning tartib raqamini aniqlaydi.

6. *FileSize*(F) funktsiyasi fayldagi bloklar uzunligini aniqlab beradi.

Ko`rsatkichlarni tavsiflash

Biz shu vaqtgacha ishlatib kelgan o`zgaruvchilar, ya'ni tavsiflash bo`limining Var bo`limida tavsiflangan barcha o`zgaruvchilar statik o`zgaruvchilar bo`lib, bu o`zgaruvchilarga dastur bajarilishiga qadar, kompilyatsiya vaqtida ularning turiga qarab ma'lum miqdorda xotiradan aniq joy ajratiladi. Kompyuterda xotiraning bu maydoni (katta) ma'lumotlar segmenti deb ataladi. Segment data - 65536 bayt. Katta miqdordagi ma'lumotlar (katta o`lchamdagi massivlar) ishlatiladigan masalalarni echishda, xamda kompyuterning tovush va grafik imkoniyatlarini ishlatishda xotira xajmi etmasligi mumkin. Bunday xollarda dinamik xotiradan foydalanish maqsadga muvofiqdir.

Dinamik o`zgaruvchilar

Dinamik xotira bu shaxsiy kompyuterning dasturga ma'lumotlar segmentidan tashqari yuklatilgan tezkor xotirasidir. Bu xotira taxminan 200-300 kbaytni tashkil qiladi. Shuning uchun katta xajmdagi ma'lumotlarni dinamik xotiraga joylashtirish yaxshi samara beradi. Dinamik xotiradan foydalanish uchun ma'lumotlarning maxsus turi- murojaat turi aniqlangan. Bu turning kattaligi sifatida maxsus dinamik o`zgaruvchi, ya'ni ko`rsatkichlar aniqlangan. Bu o`zgaruvchilarni xotirada joylashtirishni esa kompilyator amalga oshiradi. Ko`rsatkich shunday o`zgaruvchiki, uning qiymati o`zgaruvchi qiymatiga emas, balki shu o`zgaruvchi qiymati joylashgan xotira adresiga tengdir. Ko`rsatkichlarni ishlatishning o`ziga xos xususiyati shundaki, ularga statik o`zgaruvchilar singari biror nom bilan murojaat qilib bo`lmaydi. Bundan tashqari, bu turdagi o`zgaruvchilar dasturning bajarilishi davomida yaratilishi va yo`qotilishi mumkin. Ko`rsatkichlar bilan ishlashdan avval tezkor xotira tarkibi bilan qisqacha tanishib chiqamiz. Tezkor xotira maydoni 9 ta qismga bo`linadi, bu qismlarning xar biri aniq turdagi axborotni saqlash uchun xizmat qiladi:

1. Amaliyot tizimi va qobiqdar.
2. Paskal kutubxonalari.
3. Interfeys, taxrirlagich, kompilyatorlar.
4. Xatolar xaqida ma'lumot beruvchi fayl.
5. Boshlang`ich modul.
6. Ob'ekt kodi.
7. Dinamik xotira.

8. Rekursiv stek.

9. Markaziy protsessor steki.

Kompyuter xotirasining 1-6 qismlari avtomatik ravishda shakllandi va amaliyot tizimini, kutubxona, funktsiya va modullarini, taxrirlovchi, qayta ishlovchi dasturlardan tashkil topadi. Foydalanuvchi dasturi boshlang'ich modul sifatida yozilib, dasturning ob'ekt kodiga kompilyatsiya qilinadi. Dinamik xotira faqat dasturchi qo'llashi mumkin bo'lgan maxsus ko'rsatkichlar bilan to'ldiriladi. Rekursiv stek rekursiv protseduralar bilan ishlash jarayonida avtomatik ravishda shakllanadi. Markaziy protsessor steki maksimal qo'llash mumkin bo'lgan adres bilan boshlanadi. Ko'rsatkich doimo adres kamayib borish tomoniga siljib boradi. Dinamik xotiraning boshlang'ich nuqtasi «adres» Heap org deb, oxiri - Heap end deb ataladi.

Yuqorida ta'kidlab o'tganimizdek, ko'rsatkich ma'lum turdagi o'zgaruvchining qiymati yozilgan xotira adresini saqlaydigan o'zgaruvchidir, ya'ni u o'zgaruvchining qiymati joylashgan adresga murojaat qiladi.

Ko'rsatkichlarning turlari

Turbo-Paskalda ko'rsatkichlar aniq bir turdagi ma'lumotlar yoki turli turdagi ma'lumotlar yozilgan adreslarga murojaat qilishi mumkin. Shunga ko'ra ular turli va tursiz ko'rsatkichlarga bo'linadi. Turli ko'rsatkichlar quyidagicha tavsiflanadi:

<Ko'rsatkich>: [^]*<tur>*;

bu erda: [^] - ko'rsatkich belgisi, *<tur>*- ko'rsatkich murojaat qilayotgan tur nomi yoki turning berilishi. Ko'rsatkich biror qiymatni qabul qilgandan so'ng, ya'ni berilgan adres bo'yicha aniq fizik baytlarni ko'rsatgandan so'ng u erga dasturning bajarilishi davomida mos turning ixtiyoriy qiymatini joylashtirish mumkin. Ko'rsatkichlar tursiz bo'lishi xam mumkin. Bu xolda ular yordamida strukturasi oldindan noma'lum bo'lgan va dastur bajarilishi davomida o'zgaradigan kattaliklarni xotiraga dinamik joylashtirish mumkin. Shunday qilib, Turbo-Paskalda ko'rsatkichni aniq bir turga bog'lamasdan e'lon qilish mumkin. Bu esa maxsus - *Pointer* so'zi yordamida amalga oshiriladi. Dasturda ko'rsatkichlar bilan ishlashdan avval ularni yaratish, ular xajmini aniq belgilash, ko'rsatkich murojaat qiladigan dinamik xotira soxasini aniqlash lozim. Buning uchun quyidagi protsedura va funktsiyalar mo'ljallangan. Toifali ko'rsatkichlar ustida quyidagi amallarni bajarish mumkin:

1) *NEW(<Ko'rsatkich>)* protsedurasi - yangi turlashgan dinamik o'zgaruvchi (ko'rsatkich) uchun joy ajratadi va uning manzilini ko'rsatkichga joylaydi. Har qanday ko'rsatkich bilan ishlashdan avval, uni dinamik xotirada yaratish kerak. So'ngra u murojaat qilayotgan adresga ma'lumot kiritish mumkin. Parametrlar (ko'rsatkich) bir necha bo'lishi mumkin. Kompyuter dinamik o'zgaruvchining qiymatini saqlash uchun maxsus yacheyka ajratadi. Agar bu qiymat yozuv ko'rinishida bo'lsa, u xolda kompyuter uning ixtiyoriy varianti (elementi) uchun zarur bo'lgan xotira ajratadi.

Shunday qilib, bu protsedura tanlash konstantasiga mos variantli dinamik o'zgaruvchisini yaratadi. Tanlash konstantalari yozuvda qanday tavsiflangan bo'lsa, shunday tartibda ko'rsatilishi kerak

Tursiz ko`rsatkichlar uchun qo`llaniladigan funktsiyalar

Longint funktsiyasi eng katta bo`sh maydonni tanlab, uning xajmini ko`rsatadi. Ikki turdagi turlar uchun NIL konstantasi belgilangan. Const NIL tursizdir, uning qiymatini va so`zini ixtiyoriy ko`rsatkichga berish mumkin.

1. Ko`rsatkichga NIL onstantasining qiymatini berish mumkin. Boshqa turdagi ko`rsatkichlar ishlatilganda amalning chap tomonida joylashgan o`zgaruvchining turi o`ng tomonga mos kelishi kerak.

2. Ko`rsatkichlar ustida taqqoslash amallarini qo`llash mumkin. Amalning ikki chetidagi operandlar bir xil turda bo`lishi kerak.

O`z tuzilishiga ko`ra ko`rsatkich yozuvni tashkil etadi. Uning birinchi maydoni qiymatni, ikkinchi maydoni unga mos adresni ko`rsatishi kerak. Bu esa juda ko`p masalalarni echish imkonini beradi. Ko`rsatkichlar xali tavsiflanmagan kattaliklar turiga murojaat qilishi mumkin:

Kompyuterda barcha turdagi ko`rsatkichlar bir xil ko`rinishga ega, ya'ni bular kompyuter xotirasining adresidir. Shuning uchun kompyuterda fizik nuqtai nazardan ko`rsatkichlar tursiz deb xisoblanadi. Turli ko`rsatkichlarni qo`llash ko`rsatkichlarni noto`gri ishlatish natijasida yuzaga keladigan xatolarni aniqlashda yordam beradi.

Dinamik o`zgaruvchilardan foydalanish

Ko`rsatkichlardan dinamik xotirada saqlanadigan ro`yxatlar tuzishda samarali foydalanish mumkin. Ro`yxatlar, o`z navbatida, dinamik strukturaning quyidagi turlariga bo`linadi: chiziqli ro`yxat, stek, navbat.

Ro`yxat - bu tartiblangan shunday strukturaki, uning xar bir elementi keyingi element bilan bog`lanuvchi murojaatdan tashkil topadi. Ro`yxat elementi yozuv singari tashkil etilib, asosiy va qo`shimcha deb ataladigan ikkita qismdan iborat bo`ladi. Bular aniq qiymatli xar hil turli o`zgaruvchilar bo`lishi mumkin. Qo`shimcha qismda ro`yxatning keyingi yozuviga murojaat qiluvchi ko`rsatkich joylashadi. Ro`yxatning boshi doimo dasturda ishtirok etuvchi o`zgaruvchi orqali ko`rsatiladi. Agar ro`yxat bo`sh deb atalsa, qo`shimcha qiymat beriladi. Ro`yxatni tugatish uchun xam ro`yxat oxiridagi elementning qo`shimcha qismiga NIL qiymat berish kerak.

Stek bu shunday ro`yxatki, uning elementiga faqat bir nuqtadan murojaat qilish mumkin. Stek elementini olib tashlash yoki qo`shish faqat uning cho`qqisi orqali amalga oshiriladi. Bunday struktura LIFO - «oxirida kirib birinchi chiqish» deb ataladi. Ro`yxatli strukturalar ustida, asosan, formatlash, qo`shish, olib tashlash, ko`rish kabi amallarni bajarish mumkin. Stekni ishlatganda quyidagi xolatlar yuzaga kelishi mumkin: stekning to`lib ketishi, ya'ni stek xotirasida joy qolmaslik; to`lmaslik xolati - stekdan u bo`sh bo`lganda o`qishga xarakat qilish. ma'lumotlarning shunday strukturasi, uning bir tomoniga element qo`shib boriladi.

Navbat ikkinchi tomonidan olib tashlaydi. Bunday strukturani tashkil qilish uchun *LEFT* va *RIGHT* o`zgaruvchilari ishlatiladi. Navbatga element qo`shilayotganda, elementlar *RIGHT* o`zgaruvchisining qiymatiga mos xotiraga joylashadi. Shunday qilib, *RIGHT* xotiraning bo`sh joyini ko`rsatadi. Navbatdan elementlarni tanlash navbatning keyingi elementini ko`rsatuvchi qiymat orqali

amalgga oshadi. Agar $LEFT = RIGHT$ bo'lsa, u xolda navbat bo'sh xisoblanadi. Navbat ustida xam quyidagi amallarni bajarish mumkin: navbatni tashkil qilish; navbatga qo'shish; navbatdan olib tashlash; navbat elementlarini ko'rish.

Turbo - Paskal standart modullari

Turbo-Paskalning shunday imkoniyatlari mavjudki, foydalanuvchilar, ya'ni biz o'zimizga yoqqan protsedura va funktsiyalar to'plamini ixtiyoriy dasturlarda ishlatishimiz mumkin. Bu to'plamni esa maxsus fayl ko'rinishida xotirada saqlab qo'yib, unga ixtiyoriy dasturda murojaat qilish mumkin. Turbo-Paskalda bu to'plam modul deb yuritiladi. Modulga yuqorida ta'kidlab o'tilgan protseduralar, funktsiyalar, kattaliklar ma'lum qonun-qoida asosida kiritiladi. Modul kompilyatsiyadan o'tkazilgandan keyin, undan dasturlarda foydalanish mumkin. Modullar **standart va foydalanuvchi** modullarga bo'linadi. Bu modullarga murojat dasturning e'lonlar bo'limidagi *USES* maxsus so'zidan keyin amalga oshiriladi. Bu erda *USES* – paskal algoritmik tilining maxsus so'zi, crt, graph, ... - dasturda murojat etiladigan modullar ketma-ketligi. Quyida paskal algoritmik tili modullarining ayrimlari bilan tanishib o'tamiz.

System moduli

Bu modul asosan matnlar (string turi) ustida turli xil amallarni bajaruvchi funktsiyalardan tashkil topgan bo'lib, ular quyidagilardan iborat:

Concat – matnlar ketma-ketligini qo'shish amalini bajaradi. Dasturda u quyidagicha qo'llaniladi:

Bu erda $s1, s2, \dots, sN$ – string turidagi o'zgaruvchilar nomi.

Copy – matnlarning kerakli qismining qo'shimcha nusxasini oladi.

Bu erda s – boshlang'ich matn, n – matnda ajratib olinayotgan qismdagi birinchi belgining tartib nomeri, m – ajratib olinayotgan qismdagi belgilar soni.

Delete – matnlarning ayrim qismlarini o'chirish uchun ishlatiladi.

Qo'llanilishi:

Bu erda s – boshlang'ich matn, n – matnda ajratib olinayotgan qismdagi birinchi belgining tartib nomeri, m – ajratib olinayotgan qismdagi belgilar soni.

CRT moduli

Bu modul asosan ekran, klaviatura va dinamikaning ishlashini muvofiqlashtirish uchun ishlatiladi. Bu modulga murojat qilish bilan Turbo Paskal kutubxonasining quyidagi funktsiyalaridan foydalanish mumkin:

Clrscr – ekranni tozalaydi va kursorni ekranning yuqori chap burchagiga joylashtiradi.

Textcolor (color) – matnning rangini aniqlaydi;

Textbackground (color) – ekranda xosil bo'luvchi rangni aniqlaydi;

Gotoxy (i, j) – ekrandagi i qator va j ustundagi nuqtaga kursorni joylashtiradi;

Inline – kursor turgan qatorni belgilardan bo'shatadi;

Clreol – kursor turgan qatorni o'chiradi;

Delay (n) – dastur bajarilishini n millisekundga to`xtatib turadi;
Sound (m) – dinamikani oldindan berilgan m chastotada ishlashini ta'minlaydi;

GRAPH moduli

Bu modul o`z ichiga kompyuterda turli xil chizmalar va tasvirlar xosil qiluvchi funktsiyalarni olgan. Ekranda tasvirlarni xosil qilish uchun grafik rejimiga o`tish zarur bo`ladi. Bu rejimda matn rejimidan farqli ekran rang-barang nuqtalar – pixcellar to`plamidan tashkil topgan to`rtburchakdan iborat bo`ladi. Ekranning xolatini adapterlar belgilaydi va ular bir-biridan ekranga mumkin qadar ko`p nuqtalar joylashtira olish va rang-baranglikni oshirish imkoniyati bilan farqlanadi. Kompyuter odatda matn xolatida ishlaydi.

Putpixel (x, y, Color) – ekranda x, y koordinatali nuqtani Color rangida xosil qiladi;

Getpixel (x, y) – ekrandagi x, y koordinatali nuqtaning rangini aniqlaydi;

SetColor (Color) – chiziqlarning rangini belgilaydi. Color ga 0 dan 15 gacha qiymat berilib, 16 xil rangni xosil qilish mumkin. Bu erda 0 – black (qora), 15 – white (oq), ya'ni raqamlarning o`rniga ranglarning inglizcha nomlari yozilishi mumkin;

Drawpoly - ko`pburchak xosil qilish. Bu erda numpointes – ko`pburchak uchlari soni, polypointes - ko`pburchak uchlari koordinatalari berilgan massiv.

Graph modulining protseduralari

Graph modulining yana shunday protseduralari mavjudki, ular yordamida chizmalarni ekran bo`ylab harakatga keltirish mumkin. Statik chizmalarni harakatga keltirishning bir necha usullari bor. Ulardan biri harakatlantirishni takrorlanish buyrug`i orqali tashkil qilishdir. Ikkinchi usul ekranda chizilgan tasvir joylashgan sohani massiv ko`rinishida eslab qolib, uni maxsus protseduralar yordamida ekranning kerakli nuqtasiga ko`chirishdir

GetImage - xotiraning berilgan Area maydonida to`g`ri to`rtburchakli tasvirni saqlaydi. Bu protsedurada Area-tasvir saqlanadigan joyning adresini saqlaydigan o`zgaruvchi.

PutImage (X, Y, Area, mode) - ekranning berilgan joyiga tasvirni chiqaradi. Bu protsedurada (X, Y) - xotiraning Area maydonidagi tasvirdan nusxa ko`chiriladigan ekran maydonining chap yuqori burchagi.

Mode - tasvirni ekranga chiqarish rejimi;

Quyidagi dastur *GetImage*, *PutImage* protseduralari va *ImageSize* funktsiyasini ishlatgan holda tasvir chizilgan to`g`ri to`rtburchakli sohani eslab qoladi va ekranning berilgan joyiga berilgan rejimda chiqaradi.

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.