AZƏRBAYCAN RESPUBLİKASI TƏHSİL NAZİRLİYİ
SUMQAYIT DÖVLƏT UNİVERSİTETİNİN NƏZDİNDƏ
SUMQAYIT DÖVLƏT TEXNİKİ KOLLECİ

Alqoritmləşdirmənin əsasları və proqramlaşdırma fənnindən

Orta İxtisas Təhsil müəssələrində
Fənnin tədrisi üçün nəzərdə tutulub

Sumqayıt-2020

Mövzular:

1. Alqoritm nədir?.Alqoritmin xassələri.
2. Həll alqoritiminin yaradılmasının əsas prinsipləri.
3. Alqoritmin sxemlə təsviri.
4. Tipik alqoritmik strukturlar.
5. Proqramlaşdırmanın maiyyəti.
6. Proqram təminatı.
7. Proqramlaşdırma dillərinin təsnifatı.
8. Obyektyönlü,hadisəyönlü,vizual proqramlaşdırma.
9. Turbo Paskalla işləmə əmrləri.
10. Turbo Paskalla dilinin lüğəti.
11. Turbo Paskal dilinin əsas elementləri. İdentifikatorlar.
12. Standart tiplər.
13. Dəyişənlər(tam, məntiqi və bul).
14. Sətir və simvol tipli dəyişənlər. Paskal dilində istifadə olunan digər tiplər.
15. Sabitlər.Hesabi funksiyalar.
16. Sabitlər.Hesabi funksiyalar.
17. Proqramın strukturu.
18. Daxiletmə və xaricetmə operatorları.
19. Parametrli dövr operatorlar.
20. Ön şərtli dövr və son şərtli dövr operatoru.Bir-birinə daxil olan dövrlər.
21. Çoxluqlar.
22. Massivlər. Massivlər üzərində əməllər.
23. Prosedurlar.Funksiyalar.

1. Alqoritm nədir?

Alqoritm anlayışı da informasiya anlayışı kimi informatikanın əsasını təşkil edir. **Alqoritm-** qarşıya qoyulan məsələni həll etmək üçün yerinə yetirilməsi vacib olan əməliyyatlar ardıcıllığıdır.Latın dilində mənası qanun deməkdir.Alqoritm 783-850 ci illərdə Xarəzmdə (Özbəkistanda) yaşamış 9-cu əsrin məhşur özbək riyaziyyatçısı Məhəmməd İbn Musa əl- Xarəzminin (yəni Xarəzmli Musa oğlu Məhəmməd) adının latın hərflərilə olan "alqoritmi" yazılışı ilə bağlıdır. Əl-Xarəzminin yazdığı traktatın 12-ci əsrdə latın dilinə tərcümə olunması sayəsində avropalılar mövqeli say sistemi ilə tanış olmuş,onluq say sistemini və onun hesab qaydalarını alqoritm adlandırmışlar. Ümumiyyətlə,alqoritm verilmiş məsələnin həlli üçün lazım formal yazılışdır. İnsan hər gün bu və ya digər qaydalara uyğun hərəkət etmək,müxtəlif təlimatları və göstərişləri terinə yetirmək məcburiyyətində olur.Riyaziyyatın tipik məsələlərini həll etməkdikdə biz,hərəkətlərin ardıcıllığını təsvir edən müəyyən qaydalardan istifadə edirik. **Alqoritm –** sonlu sayda addımlar nəticəsində məsələnin həllini əldə etmək üçün icraçı tərəfindən yerinə yetirilən aydın və dəqiq müəyyən hərəkətlər ardıcıllığıdır. Bu riyazi mənada təyin edilmə demək olmayıb, alqoritm anlayışının intituv təsviridir və onun mahiyyətini açır. Alqoritm anlayışı nəinki riyaziyyatın ,həmçinin də müasir elmin əsas anlayışlarından biridir. Bundan başqa informatika əsrinin gəlməsi ilə alqoritm sivilizasiyasının da əsas amillərindən biri sayıla bilər.

Alqoritmin xassələri:

Alqoritm aşağıdakı xassələrə malik olmalıdır:

1. İcraçı üçün **anlamlı** olmalıdır- alqoritmin icraçısı həmin alqoritmin necə yerinə yetirilməsini başa düşməlidir. Başqa sözlə desək,alqoritm və verilənlərin ixtiyari variantına malik olan icraçı bu alqoritmin yerinə yetirilməsi üçün necə hərəkət etmək lazım olduğunu bilməlidir.

2. Diskretlilik -(kəsilmələr,ayrılmalar) –alqoritm,məsələnin həllolunma prosesini ardıcıl olaraq,sadə addımların yerinə yetirilməsi kimi dərk etməlidir.

3. **Müəyyənlik-** alqoritmin hər bir qaydası dəqiq,bir mənalı olmalı və ixtiyari hərəkətlərə yol verməməlidir. Alqoritmin tərtibi məsələnin həllini ardıcıl yerinə yetirilən mərhələlərə bölmək deməkdir. Bu zaman əvvəlki mərhələlərin nəticələri sonrakı mərhələlərdə istifadə oluna bilər və hər bir mərhələnin məzmunu və mərhələlərin yerinə yetirilmə ardıcıllığı müəyyən olmalıdır.Bu xassə sayəsində alqoritmin yerinə yetirilməsi mexaniki xarakter daşıyır və həll ounan məsələ barəsində əlavə göstərişlər və məlumatlar tələb olunmur.
4. **Nəticəvilik-** (və ya sonluluq) ondan ibarət olur ki,sonlu sayda addımlar sayəsində alqoritm ya məsələnin həllinə gətirib çıxarmalıdır ,ya sonlu sayda addımlardan sonra həllin əldə edilə bilməməsi səbəbindən dayanmalı və bu barədə uyğun məlumat verməlidir,ya da ki,alqoritmin icrası üçün ayrılmış vaxt ərzində aralıq nəticələri verməklə, icranı davam etdirməlidir.
5. **Kütləvilik-** o deməkdir ki,məsələnin həll alqoritmi ümumi hal üçün işlənir, yəni,o yalnız ilkin verilənlərlə fərqlənən müəyyən sinif məsələləri üçün tətbiq edilə bilər. Bu halda ilkin verilənlər,alqoritmin tətbiq olunma oblastı adlanan müəyyən bir oblastdan seçilə bilərlər.

2. Həll alqoritminin yaradılmasının əsas prinsipləri

Komputerdə istənilən məsələnin həlli aşağıdakı mərhələlər ardıcıllığı ilə aparılır:

1. Məsələnin qoyuluşu
2. Həll alqoritminin yaradılması
3. Hər hansı proqramlaşdırma dilində ilkin proqramın tərtibi
4. İlkin proqramın maşın dilində tərcüməsi və redaktə edilməsi
5. İşçi proqramın sazlanması
6. İşçi proqramın icraçısı və nəticələrin alınması

Məsələnin qoyuluşu- ilkin verilənlərin siyahısını,onların tipini,dəqiqliyini və ölçülərini, dəyişənlərin dəyişmə hədlərini, başlanğıc və sərhəd şərtlərini,nəticələrin siyahısını,onların tipini,dəqiqliyini və ölçülərini ,məsələnin

həllini təmin edən hesabat düsturları və tənlikləri nəzərdə tutur. Alqoritm məsələnin həllini təmin edən formal qaydalar sistemidir. Məsələnin komputerlə həlli baxımından alqoritm axtarılan cavabların alınması üçün məsələnin verilənləri üzərində icra olunan hesabi və məntiqi əməliyyatlar ardıcıllığıdır.

Alqoritmin yaradılması prosesində onun təsvirinin müxtəlif üsullarından istifadə olunur:

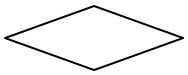
1. Təbii dil üsulu
2. Sxem üsulu
3. Psevdokod üsulu
4. Stukturoqram üsulu
5. Alqoritmik dillə təsvir üsulu

Təbii dil (alqoritmin sözlə təsviri) alqoritmin icrasının insan tərəfindən aparıldığı halda əlverişli ola bilər. Alqoritmin sözlə yazılış üsulu verilənlərin emalının ardıcıl mərhələlərinin təsvirindən ibarətdir.Alqoritm təbii dildə ixtiyari şəkildə yazılır.

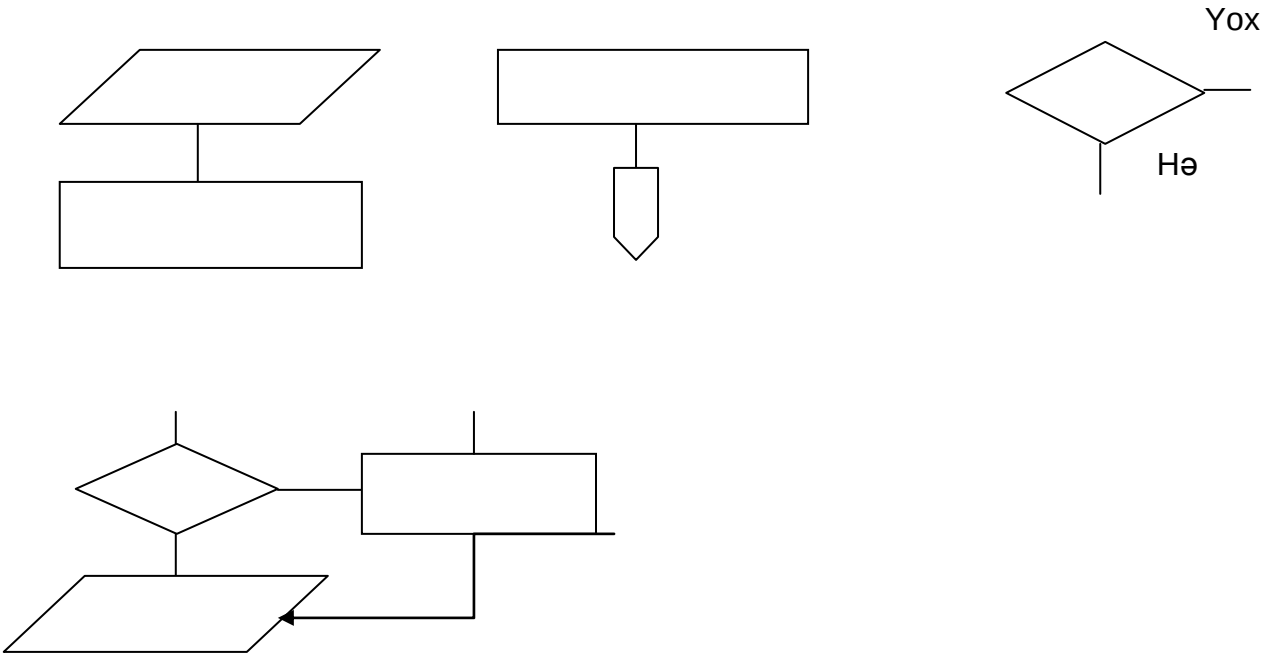
Sxem- alqoritmin qrafiki təsviri(əyani təsvir) üsuludur. Alqoritmi qrafik şəkildə təsvir etmək üçün funksional bloklardan istifadə olunur ki, o bloklardan hər biri bir və ya bir neçə hərəkətləri yerinə yetirə bilir.

3. Alqoritmin sxemlə təsviri

Sxem- alqoritmin qrafik təsviri üsuludur. Alqoritmi qrafiki şəkildə təsvir etmək üçün funksional bloklardan istifadə olunur ki,o bloklardan hər biri bir və ya bir neçə hərəkətləri yerinə yetirə bilər. Bu cür qrafik şəkildə təsvir alqoritmin sxemi və ya blok-sxem adlanır. Blok-sxemdə hər hərəkət tipinə (ilkin verilənlərin daxil edilməsi,ifadənin qiymətinin hesablanması,şərtin yoxlanılması,hərəkətlərin təkrar olunmasının idarəsi,emalın sonu və.s) **blok simvolu** şəklində təsvir olunan həndəsi fiqura uyğun gəlir. Blok simvolları keçid xətləri ilə birləşdirilir ki, o da hərəkətlərin yerinə yetirilmə ardıcıllığını təyin edir. Blokların həndəsi fiqurla ifadəsi Proqram Sənədlərinin Vahid Sistemli (PSVS) əsasında qəbul olunmuşdur.

ab bloku (proses)	
ntiqi blok (seçmə)	
iletmə-xaricetmə bloku	
etmə bloku	

Bloklar həndəsi fiqur şəklində ifadə olunur və bir-birinə şaquli və yaxud üfüqi xətlərlə birləşdirilir.

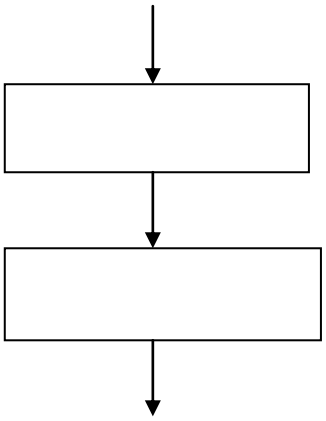


4. Tipik alqoritmik strukturlar

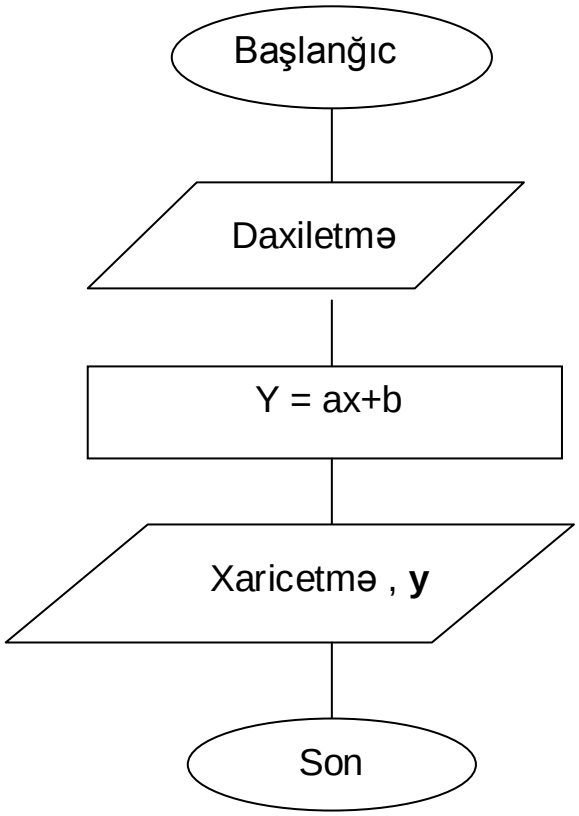
Tipik alqoritmik strukturlar alqoritmlərin və proqramların struktur yolla yaradılmasıda tətbiq olunur. Bu strukturların kombinasiyası ilə alqoritmlər və proqramlar tətbiq edilir. İstənilən hesablama prosesi aşağıdakı tipik (elementar)

alqoritmik strukturların kombinasiyasından təşkil olunur : xətti,budaqlanan, dövrü.

Xətti alqoritmik struktur. Tərkibində məntiqi və dövri bloklar olmayan iki və daha çox mərhələlər ardıcıllığından təşkil olunur. Bu strukturu sxematik olaraq aşağıdakı kimi göstərmək olar :



Misal : y= ax+ b xətti tənliyinin alqoritmini tərtib etməli:



5. Proqramlaşdırmanın mahiyyəti

Proqram modullarından ibarət olub, məsələnin həll mərhələsinə hazırlığını təmin edir. Proqramlaşdırma dili kompüterin alqoritmi qəbul etməsi üçün istifadə edilir.

Proqramlaşdırma dilləri adi dillərdən "sözlərin" (ancaq translyatorun başa düşdüyü) sayına və əmrlərin ciddi yazılış qaydasına görə fərqlənir. EHM-də proqram yazmaq üçün istifadə olunan formallaşmış dillərə proqramlaşdırma dilləri deyilir. Proqramlaşdırma dili süni dil olub, təbii dillərdən məhdud sayda sözlərin olması ilə fərqlənirlər. Bu dillərlə hazır proqramlar deyil, yalnız proqramın mətni yaradılır. Proqram dilini kompüterin başa düşdüyü maşın dilinə çevirmək üçün translyatorlardan (translator – tərcüməçi) və kompilyatorlardan (compiler – tərtibatçı) istifadə edilir. Hər bir proqramlaşdırma dilinin öz translyatoru (kompilyatoru) olur. Məsələ həll edərkən əvvəlcə yerinə yetiriləcək əməliyyatların alqoritmi tərtib edilir, daha sonra bu əməliyyatlar hər-hansı alqoritm (proqramlaşdırma) dilində əmrlər şəklində yazılır. Tərtib olunmuş proqram xüsusi əlavələr (translyator proqramlar) vasitəsilə yerinə yetirilir və ya maşın koduna çevrilir.

İstənilən proqramlaşdırma dilinin əsas elementləri bunlardır: dilin əlifbası, sintaksisi və semantikası.

- Dilin əlifbası dedikdə, həmin dildə işlənən bütün simvollar nəzərdə tutulur.
- Sintaksis – əlifbada olan simvollardan dilin ayrı-ayrı konstruksiyalarının (komandaların, operatorların) düzəldilməsinin formal qaydalarıdır. Bu qaydalar müxtəlif həll alqoritmlərini proqramlaşdırmağa imkan verir.
- Semantika – dilin bu və ya digər sintaksis konstruksiyalarının təsviridir. Məsələn, əgər proqramın bu yerində y =a*(b+c) ifadəsinin hesablanması yazılıbsa, onda semantika qaydaları maşına "göstərir" ki, əvvəlcə (b+c) cəmini tapsın, sonra həmin cəmi a-ya vursun.

Beləliklə, hər hansı verilənlərin emalı prosesini birbaşa həyata keçirməyə imkan verən proqramlar, dili təyin edən sintaksis qaydalara uyğun olaraq əlifbadakı simvolların birləşməsi nəticəsində və semantika qaydalarını nəzərə almaqla işlənib hazırlanır.

Proqramlaşdırma dili vasitəsilə hazır proqram yox, ancaq qurulmuş alqoritmi təsvir edən mətn yaradılır. Proqramçının başa düşdüyü dildə olan bu proqram maşının başa düşdüyü dilə çevrilməlidir. Bunun üçün kompüterdə translyatorlar və kompilyatorlar olur.

Proqram ancaq onların translyatorları olan halda icra oluna bilərlər.

Translyatordan fərqli olaraq kompilyatorlar exe-faylların yaradılması üçün istifadə olunur ki, onlar da sərbəst icra oluna bilərlər (yəni, proqpamin yazildiği mühitdən (sistemdən) asılı olmadan). Proqramlaşdırma dillərinin səviyyələri.

Müxtəlif tip prosessorlar müxtəlif tip əmrlər sisteminə malikdir. Əgər proqramlaşdırma dili konkret prosessor tipinə yönəlibsə və onun xüsusiyyətlərini nəzərə alırsa, onda ona aşağı səviyyəli proqramlaşdırma dili deyirlər.

Assembler aşağı səviyyəli proqramlaşdırma dilidir. Çünki o, bir əmri mnemonika adlanan simvol işarəmələrinin köməyilə ədədlər şəklində yox, maşın kodları şəklində verir. Assemblerin köməyilə çox səmərəli və kompakt proqramlar yaratmaq mümkündür. Assemblerdən adətən, sistem əlavələrin, drayver-proqramların, kompüterin aparat resurslarına müraciət edən proqram modullarının hazırlanması üçün istifadə olunur. Aşağı səviyyəli proqramlaşdırma dillərindən, adətən yüksək səviyyəli peşəkar proqramçılar istifadə edir. Bu dillərdə tutulan proqramlar yaddaşda az yer tutmaqla yanaşı, daha sürətlə icra olunurlar. Yüksək səviyyəli proqramlaşdırma dilləri isə adi dilə daha yaxın və insan üçün daha aydın başa düşüləndir. Çox yayılmış, bəzi proqramlaşdırma dilləri haqqında məlumat verək

Proqramlaşdırma dilləri iki hissəyə bölünür:

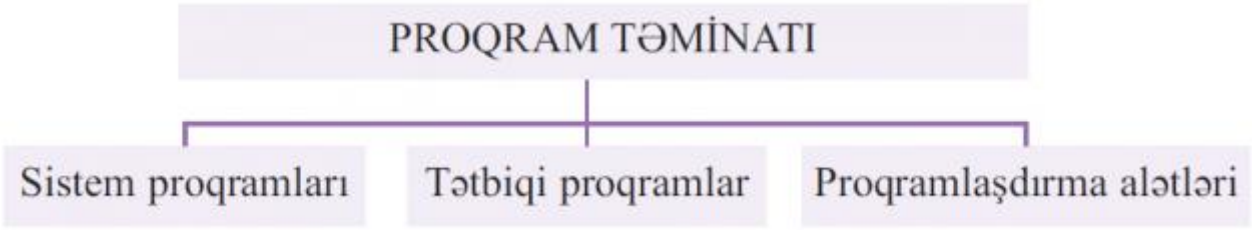
- Aşağı səviyyəli dillər (Assembler, Avtokod və s.),
- Yüksək səviyyəli dillər (Fortran, Alqol, Kobol, Basic, Pascal, Ci və s.).

Aşağı səviyyəli proqramlaşdırma dillərində hər operatora bir maşın əmri uyğun gəlir. Bu dildə yazılan proqram az yer tutur və tez yerinə yetirilir. Aşağı səviyyəli dillərdən sistem proqramçılar istifadə edir. Yuxarı səviyyəli proqramlaşdırma dillərində hər operator bir neçə maşın əmri ilə əvəz edilə bilər, bu isə yaddaşda çox yer tutur. Yüksək səviyyəli dillərdən isə tətbiqi proqramçılar istifadə edir.

6 . Proqram təminatı

Kompüterin hər hansı bir işi yerinə yetirməsi üçün aparat təminatı ilə yanaşı,ona göstərişlər toplusu, yəni proqramlar lazımdır. Klaviaturada klavişin basılmasına,siçanın hərəkətinə, başqa kompüterdən informasiyanın alınmasına və digər hərəkətlərə kompüterin necə reaksiya verməsini məhz proqramlar müəyyən edir. Ekrana görüntünün çıxarılmasını, sənədin printerdə çap olunması üçün hazırlanmasını, kompüterdə musiqinin səsləndirilməsini proqramlar həyata keçirir.

Kompüterin proqram təminatı [software] kompüter sisteminin ayrılmaz bir hissəsi olub, kompüterin texniki təminatının məntiqi davamını təşkil edir. Kompüterin konkret tətbiq sahəsi onun proqram təminatı ilə müəyyən olunur. Kompüterin özlüyündə heç bir “bacarığı” yoxdur. Bütün “bacarıqlar” kompüterdə icra olunan proqramlarda cəmləşdirilib. Müasir kompüterlərin proqram təminatı oyun proqramlarından tutmuş elmi proqramlara kimi milyonlarla proqramdan ibarətdir. Kompüterdə olan bütün proqramları şərti olaraq üç sinfə ayırmaq olar: sistem proqramları, tətbiqi proqramlar, proqramlaşdırma alətləri.



Sistem proqramları.

Sistem proqramları kompüterin resurslarını – mərkəzi prosessoru, yaddaşı,

giriş-çıxış qurğularını idarə etmək üçündür. Onlar bütün istifadəçilər üçün nəzərdə tutulmuş proqramlardır. Kompüterin sistem proqramları elə hazırlanır ki, tətbiqi proqramlar səmərəli işləyə bilsin. Sistem proqramları arasında əməliyyat sistemləri xüsusi yer tutur, sistem proq ram təminatının əsasını əməliyyat sistemi təşkil edir. O, fərdi kompüterlərin vacib elementlərindən biridir. Əməliyyat sistemi kompüter yandırıldıqda işə düşən, kompüterin bütün hissələrinin tam bir vəhdət halında işləməsini təmin edən və informasiyanı idarə edə bilən proqramlar sistemidir.

Əməliyyat sisteminin köməyilə: • kompüterlə istifadəçi arasında dialoq yaranır;

- operativ və daimi yaddaş qurğuları işə salınır;
- kompüter idarə olunur;
- istənilən proqram yerinə yetirilməyə başlayır və s.

Vaxtilə IBM PC tipli kompüterlərdə əsasən Microsoft firmasının hazırladığı MS-DOS əməliyyat sistemindən istifadə olunurdu. Bu əməliyyat sistemində işləyən istifadəçi yalnız konkret bir məsələni həll edə bilərdi. Hazırda fərdi kompüterlərdə çoxtapşırıqlı əməliyyat sistemlərindən istifadə olunur – fərdi kompüterlərin yaddaşında eyni zamanda bir neçə proqram və məsələlər olur ki, mikroprosessor kompüterin resurslarını onların arasında bölüşdürür. Belə əməliyyat sistemlərinə misal olaraq OS/2, MacOS, UNIX, Linux, Windows XP, Windows Vista və digər əməliyyat sistemlərini misal gös tər mək olar.

Sistem proqramlarının digər vacib hissəsini **xidməti proqramlar – utilitlər** (lat. “utilitas” – xeyir, fayda) təşkil edir. Onlar əməliyyat sistemini tamamlayır və onun imkanlarını artırır, həmçinin, müstəqil olaraq bir çox vacib məsələləri də həll edir. Utilitlərin bəzi növləri bunlardır:

- interfeys proqramları;
- antivirus proqramları;
- arxivləşdirmə proqramları;
- proqram örtükləri;
- kompüter qurğularının iş qabiliyyətini yoxlayan proqramlar;

• qurğuların işini idarə edən proqramlar (drayverlər) və s.

Tətbiqi proqramlar.

İnsan fəaliyyətinin müxtəlif sahələrinə aid məsələləri həll etmək üçün nəzərdə tutulan proqram təminatına tətbiqi proqramlar deyilir. İndiki zamanda fərdi kompüterlər üçün yüz minlərlə tətbiqi proqramlar işlənib hazırlanmışdır. Onlardan ən çox istifadə olunanlar bunlardır:

• mətn redaktorları (prosessorları);

• cədvəl verilənlərinin emalı proqramları – elektron cədvəllər;

• nəşriyyat sistemləri;

• verilənlər bazasının idarə olunması sistemləri;

• təqdimatların hazırlanması proqramları;

• qrafik redaktorlar;

• verilənlərin statistik təhlili proqramları;

• kompüter oyunları, öyrədici proqramlar və s.

Proqramlaşdırma alətləri.

Bu sinfə aid olan proqramlar sistem və tətbiqi proqram təminatını yaratmaq üçün nəzərdə tutulmuşdur. Proqram təminatının hazırlanması üçün Basic, C++, Pascal və b. proqramlaşdırma dillərindən istifadə olunur. Dünyanın bir çox təhsil müəssisələrində uşaqlara proqramlaşdırmanın əsaslarını öyrətmək üçün LOGO dilindən istifadə olunur.

7.Proqramlaşdırmanın dillərinin təsnifatı

Proqramlaşdırma dillərinin təsnifatının əsas əlamətlərindən biri dilin hasnı proqramlaşdırma üslubuna mənsub olmasıdır.Proqram texnologiyasında əsasən aşağıdakı üslublardan istifadə olunur:

1. prosedur proqlaşdırma
2. məntiqi proqlaşdırma

3. obyektyönlü proqramlaşdırma
4. hadisəyönlü proqramlaşdırma
5. vizual proqramlaşdırm

Proqramlaşdırmada müxtəlif səviyyəli dillərdən istifadə edilir: maşın dilləri, Assembler, yüksək səviyyəli (alqoritmik) dillər.Maşın dili konkret kompüterin əmrlər sistemindən ibarət olub, bilavəsitə həmin maşın tərəfindən həyata keçirilir. Maşın dilində proqram tərtib etdikdə hər şeydən əvvəl dəyişənlər və konstantlar üçün maşının yaddaşında yer ayrılır. Maşın dilində proqram maşın əmrləri ardıcıllığından və dəyişənlər, konstantlar üçün yaddaşda təyin edilmiş müəyyən sahələrdən ibarətdir.

Assemberlər səviyyəli dillər konkret kompüterlərin əmrlər sisteminə uyğun gələn maşınyönümlü dillərdir.Buna baxmayaraq, onlar proqramı istifadəçi üçün daha rahat olan formada tərtib etməyə imkan verirlər.Assembler dilinin üstün cəhəti ondadır ki, dildə əmrlərə, konstantlara və dəyişənlərə müəyyən adlar mənsub edilir və bu adlar vasitəsilə onların özlərinə müraciət etmək imkanı yaranır. Assembler dilində kompüterin bütün imkanlarından tam istifadə etməyə imkan verən effektiv proqramlar yazılır.

Çatışmayan cəhət proqramın həddindən artıq təfərrüfatı ilə yazılmasıdır.

Yüksək səviyyəli dillər iki sinfə bölünürlər:

problemyönümlü dillər;

proseduryönümlü dillər;

Problemyönümlü dillər çox kiçik sinif təşkil edən məsələləri həll etmək üçün təyin edilmişdir Proqramlaşdırma prosesində həll alqoritmi deyil, məsələnin özü təsvir edilir.

Proseduryönümlü dillər məsələnin həll alqoritmi təsvir etmək üçün təyin edilmişdir.Onlar öz növbəsində maşından asılı olan və maşından asılı olmayan alqoritmik dillərə bölünürlər .

Maşından asılı olan yüksək səviyyəli dillər maşının bütün imkanlarından tam istifadə etməyə, aydın və asanlıqla oxunan proqramlar yazmağa imkan verirlər.Bu dillərdə operatorlar və ifadələr istifadəçi üçün daha rahat şəkildə yazırlar.Onlara misal olaraq PL/M dilinin göstərmək olar.Bununla belə bu dillər konkret quruluşdan çox asılıdır və bu səbəbə görə praktikada geniş yayılmışdır.

Maşından asılı olmayan yüksək səviyyəli dillərin yaxud alqoritmik dillərin tərkibində maşından asılı olan operatorlar iştirak etmir .Bu dillərə Alqol, Fortran, Beysik, Fokal,PL/1, Paskal və s. daxildir.Alqoritmik dillərin əsas üstünlüyü proqramçının yüksək əmək məhsuldarlığı, proqramların asanlıqla bir maşından digərinə keçirilməsi, proqramlardan asanlıqla istifadə etmək imkanının olmasıdır.Maşın dilində yazılmış xüsusi proqram-transiyator alqoritmik dildə təsvir edilən alqoritmin simvolik təsvirini emal edir və proqramı avtomatik olaraq maşın dilinə çevirilir.

Perl 80-ci illərdə Larri Uoll tərəfindən işlənmişdir. Bu proqram dilinin böyük həcmli mətn fayllarının effektiv işlənməsində, hesabatların generasiyasında və məsələlərin idarəsində istifadəsi nəzərdə tutulmuşdur.
Perl-dən sətirlərlə, massivlərlə, ayrı-ayrı verilənlərlə, proseslərin idarəsində, system informasiyaları ilə işlərdə istifadə edilir. HTML web səhifələrin hazırlanmasında istifadə edilən populyar dildir.
Assembler-dən başqa, qalan proqramlaşdırma dillərinin hər biri yüksək səviyyəli dil adlandırılır, ancaq bu hələ onların eyni səviyyəli olması demək deyildir. Bir dilin səviyyəsi başqasının səviyyəsindən yuxarı, aşağı ola bilər.
"Yüksək səviyyəli dil" dedikdə, onun insan dilinə nə qədər yaxın olması başa düşülür. Beləliklə, insan üçün daha anlaşıqlı olan və proqramlaşdırma prosesini asanlaşdıran yeni dillər yaradılmağa başladı.
Proqramlaşdırma dillərinin müxtəlif səviyyələri var. Əsasən 5 qrupa ayrılırlar:
1. Çox yüksək səviyyəli dillər və ya vizual dillər: Access, FoxPro, Paradox, XBase, Visual Basic.
2. Yüksək səviyyəli dillər (bunlara bəzən "alqoritmik dillər" də deyilir): Pascal, Basic, Fortran
3. Orta səviyyəli proqramlaşdırma dilləri: C, C++
4. Aşağı səviyyəli proqramlaşdırma dilləri: Assembly language
5. Maşın dili: Ən aşağı səviyyəli dil olub, 0 və 1-lərdən ibarətdir.
Bundan başqa, proqramlaşdırma dillərini 2 ayrı qrupa da bölmək olar:
1. Prosedur proqramlaşdırma dilləri (Pascal, Basic)
2. Obyekt yönümlü proqramlaşdırma dilləri (C++, Java, Smaltalk)
Hər hansı proqramlaşdırma dilini istifadə etmək üçün isə, bizə ilk növbədə kompilyator lazımdır. Kompilyator olduqdan sonra, biz öz proqramımızı mətn redaktorunda (məsələn, Notepad) yaza bilərik.
Lakin bizə daha çox IDE, yəni proqramlaşdırmanın inteqrallaşmış mühitindən istifadə edirik. Sadə dillə desək, bizə bir mühit verilir, orda komponentlər olur və bunlardan istifadə edərək, biz öz proqramımızı yazırıq.
Və ən əsası, proqramlaşdırmanı öyrənmənin yolu proqram yazmaqdır.

8.Obyekt-yönlü proqramlaşdırma

Obyekt-yönlü proqramlaşdırmanın bir çox vasitələri simulə-67 dilindən götürülmüşdür. Proqramlaşdırmanın obyekt-yönlü üslubu obyekt anlayışına əsaslanır,mənası isə " obyekt= verlənlər+presedurlar " düsturu ilə ifadə olunur. Hər bir obyekt özündə verilənlərin stukturunu birləşdirir vəonlara müraciət bu verilənlərin emalı proseduru ilə mümkündür ki,bu da metod adlanır. Verilənlər və prosedurun bir obyektdə birləşməsi inkapsulyasiya adlanır. Obyektlərin təsvirinə siniflər xidmət edir. Sinif bu sinifə aid olan obyektlərin xassə və metodlarını təyin edir. Uyğun olaraq,istənilən obyekt sinfin nüsxəsini təyin edə bilir. Yeni vərəsə obyektlərin yaradılmasında valideyn obyektlərin xassəsi əlavə oluna bilər. Buna xassələrin irsən verilməsi və ya vərəsəlik deyilir. Obyektlərlə iş prosesində polimorfizmə icazə verilir, yəni müxtəlif tiplin verilənlərin emalı üçün eyni adlı metodlarından istifadə etmək imkanı var. Müasir obyekt-yönlü proqramlaşdırma dillərinə Smalltalk, C++, Object Paskal, Java vəs. aiddir. 1983-cü ildə C- nin siniflərlə variantı, bir az sonra isə C++ yarandı. 1990-cı ildə Sun kornorasiyasının əməkdaşı D.Qoslinq C++ genişlənməsi əsasında obyekt-yönlü OaK dilini yaratdı. Dilin əsas dəyəri müxtəlif tip qurğuların qarşılıqlı şəbəkə əlaqəsinin təminidir.İnternet-də yayılmaq üçün bu dildə yazılan böyük olmayan proqramlar appelet adlandırıldı. Bu dilin internetdəki yeni versiyası **Jave** adlandırdı. İlk brauzer Sun korporasiyanın 25 yaşlı proqramçı P.Noton tərəfindən yaradılaraq, əvvəl WebRunner, sonra isə HotJava adlandırıldı. Obyekt-yönlü vasitə imkanı nöqteyi nəzərdən Java dilinin C++ ə nəzərən bir sıra üstünlükləri var.Belə ki,Java dili informasiyanın inkapsulyasiyası üçün olduqca çevik və güclü sistemə malikdir. Obyekt-yönlü proqramlaşdırma ideyası bir çox universal prosedur dillərdədə istifadə olunur.

Hadisə-yönlü proqramlaçdırma- obyekt-yönlü proqramlaşdırmaya əsaslanır və sistemdə baş verən hadisələrə reaksiya verən obyektlərdən istifadə edilməsi nəzərdə tutur. Hadisə-yönlü proqramlaşdırmadan həm müstəqil proqramların, həm də əməliyyat sisteminin qurulmasında istifadə olunur. Hadisə-yönlü və obyekt yönlü yanaşmadan istifadə etməklə hazırda çoxlu

sayda proqram şablonları yaradılmışdır. Bu cür proqram şablonlarından ibarət olan kitabxanalar ya özünün hadisə dispeçerinə malik olur,ya da əməliyyat sisteminin vasitələrindən istifadə edirlər.

Vizual proqramlaşdırma- Son vaxtlar proqramlaşdırmaya vizual yanaşma geniş yayılmışdır. Vizual proqramlaşdırma obyekt-yönlü və hadisə yönlü proqramlaşdırmanın sonrakı inkişafı nəticəsində yaranmışdır. Vizuallaşdırma mürəkkəb proseslərin kompüterin ekranında qrafik primitivlər (fiqurlar) şəklində əks etdirilməsidir. İstənilən prosesi idarə etməni,quraşdırmanı,şəkil çəkməyi və s.vizuallaşdırmaq olar. Vizuallaşdırmanın sadə variantı icra etmənin və ya inkişafın gedişinin hər hansı füqur vasitəsilə əks etdirilməsidir. Məsələn: düzbucaqlının içərisinin doldurulması faizi hər hansı əməliyyatın nə qədər icra olunmasını göstərir. Proqram təminatının interfeysini vizuallaşdırmaqla istifadəçinin proqram məhsulu ilə işlənməsini sadələşdirmək olar. İnterfeysin elementlərindəki şəkillər və yazılar həmin elementlərin funksiyalarını asan qavramağa kömək edir.

Proqram təminatının vizual komponentinə sadə misal kimi ekranda vizual düyməni göstərmək olar. Həmin düymə əsl idarəetmə düyməsini imitasiya edir. Onu əsl düymə kimi “basmaq” olar.

Vizual proqramlaşdırmadan hazırda bir sıra proqramlaşdırma sistemlərində istifadə edilir. Bunlara misal olaraq Visual Basic,Visual C++,C++ Builder və s. sistemlərini göstərmək olar. Proqramlaşdırma dillərindən əlavə,vizual yanaşmadan digər sistemlərdə də istifadə olunur. **Məsələn :** Visual ForPro, Paradox, for Windows, MS Office paketinin proqramları və s.

Bütün bu sistemlərdə,o cümlədən,hazırda ən populyar hesab edilən Visual Basic və Delphi sistemlərində vizuallaşdırılan model kimi pəncərə istifadə edilir.

9. Turbo Paskal işləmə əmrləri

Turbo Pascal 7.0 inteqrallaşdırılmış mühitində ən çox istifadə edilən əmrlər

File/New *Yeni proqram faylı yaratmaq*

File/Open... F3 *Mövcud pascal–proqram faylını açmaq*
File/Save F2 *Proqram mətnini yadda saxlamaq*
File/Save As... *Proqramı yeni adla yadda saxlamaq*
File/Exit Alt+X *İnteqrallaşdırılmış mühitdən çıxmaq*
Edit/Cut Shift+Del
Seçilmiş proqram fraqmentini kəsib
buferdə yerləşdirmək
Edit/Copy Ctrl+Ins
Seçilmiş proqram fraqmentini buferə
köçürmək
Edit/Paste Shift+Ins
Buferdə olan proqram fraqmentini
kursorla göstərilən mövqeyə yerləşdirmək
Edit/Clear Ctrl+Del *Seçilmiş proqram fraqmentini pozmaq*
Compile/Make F9
Proqramı sazlamaq (səhvləri aşkar
etmək)
Run/Run Ctrl+F9 *Proqramı yerinə yetirmək*
Compile/Compile Alt+F9 *Proqramı kompilyasiya etmək*
Debug/User
screen
Alt+F5
İstifadəçi pəncərəsini ekranda göstərmək
və ya gizlətmək
Run/Trace into F7 *Proqramı sətirlərlə yerinə yetirmək*
Run/Program
Reset
Ctrl+F2 *Proqramın icrasından imtina etmək*
Debug/Add
Watch
Ctrl+F7
Dəyişənləri müşahidə pəncərəsinə əlavə
etmək
Ctrl+Break *Proqramın icrasını dayandırmaq*
F10 *Menyu sətrini aktivləşdirmək*

10. TURBO PASKAL DILININ LÜĞƏTI

Hər hansı dilin əlifbası sonlu sayda simvollar ardıcıllığıdir. Turbo Pascal dilinin əlifbası aşağıdakı qruplara bölünür.

1. A-dan Z-ə kimi böyük və kiçik latın hərfləri.
2. 0-dan 9-a kimi onluq rəqəmlər.
3. Xüsusi simvollar: . () , # $ ^ @ * _ + " : > < =
4. Boşluq simvolu.

Münasibət əməlləri.

Tam ədədlərdən ibarət cütlər üzərində təyin olunmuş və nəticədə məntiqi qiymət verən əməllər də vardır.

Həmin əməllər bunlardır:

= bərabərdir	< kiçikdir
<= kiçikdir və ya bərabərdir	<> bərabər deyil
> böyükdür	>= böyükdür və ya bərabərdir

Bu əməllər həqiqi ədədlər üzərində də təyin olunub.

Simvol qiymətlər üzərində isə, yalnız = və <> əməlləri təyin olunmuşdur.

11.Turbo Pascal alqoritmik dilin əsas elementləri

Dilin əlifbası. Turbo Pascal dilində proqram müəyyən hərflər, rəqəmlər və simvollar vasitəsilə yazılır. Bu simvollar dilin əlifbasını təşkil edir. Dilin əlifbası ASCII kodlaşmış simvollar cədvəlinə əsaslanır və dilin baza elementi sayılır. Onun
vasitəsilə dilin digər struktur elementləri – sabitlər, dəyişənlər, ifadələr, identifikatorlar, operatorlar, altproqramlar, modullar və digər obyektlər tərtib olunur. Əlifbanın simvollarını şərti olaraq aşağıdakı qruplara bölmək olar:

1. Latın əlifbasının hərfləri və _ simvolu;

2. Onluq say sisteminin 0-dan 9-a qədər hind-ərəb rəqəmləri;

3.Onun müasir sələfi 1974-cü ildə yaradılmış İcon dilidir.

4. Bu dil XVII əsr böyük fransız alimi Paskalın şərəfinə adlandırılmışdır.

5. 1995-ci ildə dilin yeni versiyası ADA -95 yaradılmışdır.

Xüsusi simvollar və məhdudlaşdırıcılar:
+ - * / = < > > < <=>= := ( ) {} [] ^ # $ (**) , . ; ’

Cüt simvollar

: <> < = > = := (*;*);

Probel və idarəedici simvollar.

Turbo Pascal dilində probel ayırıcı funksiyasını yerinə yetirir. Ardıcıl yazılmış bir neçə probel işarəsini kompilyator bir probel kimi qəbul edir. Idarəedici simvollar sətir və simvol tipli sabitlərin təsviri üçün istifadə olunur. Sətir və simvol tipli sabitlərin təsvirində və proqramda yazılmış şərhlərdə həmçinin ASCII kodlaşmış simvollar cədvəlinin kodu 128-255 intervalında dəyişən simvollardan,o cümlədən psevdoqrafika simvollarından istifadə etmək olar.

İşçi sözlər. İşçi sözlər proqramın operatorların, alt proqramların yazılmasında istifadə olunur və ad kimi istifadə oluna bilməz. Turbo Pascal dilində aşağıdakı işçi sözlər nəzərdə tutulmuşdur:

and ,array,as,begin,div,do,file, if in,is mod,not,of,var,xor

Sabitlər və dəyişənlər. Hər bir proqram kompyuterdə yerinə yetirilən zaman verilənlər üzərində müəyyən əməliyyatlar aparır. Verilənlər iki yerə - sabitlərə və dəyişənlərə bölünür:

Dəyişənlər proqramın yerinə yetirilməsi prosesində müxtəlif qiymətlər alan adlı kəmiyyətlərə deyilir. Dəyişənlərdən fərqli olaraq *sabitlər* əvvəlcədən məlum olan və proqramın yerinə yetiriləcəyi müddətdə dəyişməz qalan kəmiyyətlərdir. Sabit və dəyişənlərə onların adları ilə müraciət olunur. Sabitlər və dəyişənlər müəyyən tipə malikdir.

Tip verilənin kompyuterin yaddaşındakı ifadə forması ilə təyin edilir və onun qiymətlər oblastını, onunla aparılan əməliyyatlar coxluğunu müəyyən edir.

Sabitlər:

adsız və adlı kəmiyyət kimi proqramda istifadə oluna bilər. Adlı sabitlər və dəyişənlərin tipi mütləq proqramın təsvir bölməsində elan edilməlidir. Bu zaman sabitin tipi aşkar şəkildə proqramlaşdırcı tərəfindən verilə bilər. Əks halda kompilyator sabitin tipini onun qiymətinə əsasən müəyyən edə bilər.

1. Cüt simvollar arasında probel işarəsinin qoyulması yolverilməzdir.
2. Hər bir dəyişən proqram daxilində unikal ada malik olur.

Adsız sabitlər və adlı sabitin aldığı qiymət kimi Turbo Pascal dilində tam, həqiqi, onaltılıq ədədlər, məntiqi
FALSE
(yalan) və ya
TRUE
(doğru) sözləri,
1. simvollar, sətirlər, çoxluq konstruktorları
2. və qeyri-müəyyən göstərici əlaməti olan
istifadə edilə bilər.

İdentifikatorlar. Turbo Pascal dilində proqramlara, sabitlərə, dəyişənlərə, tiplərə, modullara, prosedura və funksiyalara ad vermək üçün identifikatorlardan
istifadə olunur. İdentifikator latın hərflərindən, "_" simvolundan və onluq say sisteminin rəqəmlərdən ibarət simvollar ardıcıllığıdır və onun birinci simvolu mütləq hərf olmalıdır. İdentifikatorun uzunluğu, yəni onu təşkil edən simvolların sayı qeyri-məhdud ola bilər. Lakin, proqram tərtib edərkən nəzərə almaq lazımdır
ki, Turbo Pascal kompilyatoru identifikatorun ilk 63 simvolunu fərqləndirir. Standart adlar və işçi sözlər identifikator olaraq işlədilə bilməz. Standart adlar standart funksiyaların, prosedurların, standart faylların və sabitlərin, tiplərin adları
ola bilər. Turbo Pascal dilində işlədilən standart adlar aşağıdakılardır:
ArcTan False Pi
Assign FilePos Port
Aux FileSize
Pos
AuxInPtr FileChar Pred
AuxOutPtr Flush

12.Standart tiplər

Standart tiplərin adları əvvəldən təyin olunmuş identifikatorlardır və proqramın ixtiyari yerində iştirak edə bilər. Bu tiplər standart system modulunda təsvir olunur.

Turbo Pascalda standart tiplərə aşağıdakılar aiddir:

tam tiplər

Tam dəyişən və sabitləri təsvir etmək üçün beş tip mövcuddur ki, onların xarakteristikaları cədvəldə verilmişdir:

Tipin adı	identifikator	ədədin təsvir diapazonu	Yaddaş ölçüsü
İşarəli qısa tam	Shortint	-128.. 127	1 bayt
İşarəli tam	integer	-32768..32767	2 bayt
İşarəli uzun tam	longint	-2147483648.. 2147483647	4 bayt
Işarəsiz qısa tam	Byte	0..255	1 bayt
İşarəsiz tam	Word	0..65535	2 bayt

Həqiqi tiplər. Bu qrupa beş tip daxildir ki, bunlar cədvəldə göstərilmişdir:

Tipin adı	identifikator	ədədin təsvir diapazonu	Mantissadakı rəqəmlərin sayı	daş ölçüsü
rqat dəqiqlikli həqiqi				
	Single	1.5·10-45÷3.4·1038	7..4	4 bayt

Həqiqi	real	2.9·10-39÷1.7·1038	11.. 12	6 bayt
iqat dəqiqlikli həqiqi	Double	5.0·10-324÷1.7·10308	15.. 16	8 bayt
ksək dəqiqlikli həqiqi	Extended	3.4·10-4932÷1.1·104932	19..20	l0 bayt
am həqiqi formatda	comp	63+1.. 263-1 və ya təqribi - 9.2·10-18÷9.2·1018	19.. 20	8 bayt

Bu qrupda tam və həqiqi tiplərin qarışığı Comp tipini xüsusi qeyd etmək lazımdır.

Turbo Pascalda həqiqi tip ədədlər üzərində əməliyyatların yerinə yetirilməsi üçün kodun generasiyasının iki üsulu vardır:

-80x87 soprosesoru olduqda (aparat üsulu);

- 80x87 soprosesoru olmadıqda (proqram üsulu);

Bu üsulların seçilməsi kompilyatorun $N və $E direktivləri ilə həyata keçirilir.

13.Dəyişənlər(Tam,məntiqi,Bul)

Həqiqi dəyişənlər-həqiqi ədədlər mantissası 11 ikilik mənbə olanhəqiqi dəyişənlər 2,9E-39+1,7E+38 qiymətlər alır və bu tipli dəyişənlər yaddaşda 6 bayt yer tuturlar. Həqiqi dəyişənlər üzərində aşağıdakı əməliyyatları aparmaq olar; =; +; -; *; /;

Əgər ifadədə həm həqiqi, həm də tam dəyişən iştirak edirsə,onda tam dəyişən həqiqi dəyişənə çevrilir.

Məntiqi və ya Bul dəyişənlər-Turbo Paskalın 6.0 versiyasına kimi yalnız bir Boolean bul tipi var idi ki,bul-a iki məntiqi qiymət True(doğru) və False(yalan) qiymətlər ala bilər. Turbo Paskalın 7.0 versiyasına daha üç Byte bool,Word Bool, Long Bool Bul tipləri daxil edilib.

Tipin identfikatoru	False-nin qiyməti	True-nun qiyməti	Yaddaş ölçüsü
boolean	0 ədədi	Sıfırdan fərqli	1 bayt
Byte Bool	0 ədədi	ədəd	1 bayt
Word Bool	Hər 2 baytda 0 ədədi		2 bayt
Long Bool	Bütün baytlarda 0 ədədi		4 bayt

Bu tipə malik olan dəyişlər iki qiymət ala bilər; True(doğru) və False(yalan). Yaddasda bir bayt yer tutur.Qeyd etmək lazımdır ki,buradakı yeni bul tipləri, Windows mühitində proqramların yaradılmasını təmin etmək üçün daxil edilmişdir. False qiymətində 0 True qiymətində isə 0 dan fərqli ixtiyari ədəd uyğundur. Bu qiymətlər üçün aşağıdakı münasibətlər doğrudur. True>False

Bul dəyişənlər üzərində aşağıdakı əməliyyatlar aparmaq olar.

Or- məntiqi toplama əməli

And- məntiqi vurma əməli

Not- inkar

Bul tip dəyişənlər üzərində müqayisə əməliyyatlarını da aparmaq olar. $\leq, \geq, <>, =, >, <$

14. Sətir və simvol tipli dəyişənlər

Simvol tipli verilənlər- CHAR tipi kimi göstərilir.Simvol tipli verilənlər üzərində müqayisə və mənimsətmə əməliyyatları aparmaq olar. Bu tipə aid olan sabit və dəyişənlərin qiymətləri dırnaq arasında verilir. Qeyd etmək lazımdır ki hər simvola 0...255 arasında bir ədəd kod uyğundur. 0...31 arasındakı kodlar xidməti kodlardır. Əsas simvollara uyğun kodlar 32...127 aralığında yerləşir. Məsələn; 4 rəqəminin simvol kodu 52,A hərfinin simvol kodu 65,a hərfinin (kiçik və baş hərflərin kodları fərqlidir) simvol kodu 97-dir. Simvol tipli dəyişənlərin

qiymətləri üzərində əməliyyatlar aparmaq üçün çoxlu standart funksiyalar var; Chr,Ord.

Sətir tipli dəyişənlər(string)- sətir tipli dəyişənlər simvol tipli dəyişənlərin bir növüdür. Sətir tipli dəyişənləri təmin edən zaman sətrin maksimal uzunluğunu (simvol sayını) göstərmək lazımdır. Bu tipə malik olan dəyişən STRİNG sözünün köməyi ilə təyin olunur. Bu sözdən sonra kvadrat mötərizənin içərisində sətrin maksimal uzuluğu yazılır.Bu ədəd tam ədəd (0...255 aralığında)olmalıdır. Əgər uzunluq göstərilməzsə,onda 255 qəbul edilir. Sətir tipli dəyişənlər yaddaşda tutduqları yer onların uzunluğundan 1 bayt artıqdır. Sətrə aid olan simvollar 1-dən başlayaraq sətrin uzunluğunu göstərən ədədə qədər indeksləşdirilir. Sətrin hər bir simvoluna müraciət etmək üçün bu indekslərdən istifadə olunur.Sətir və simvol tipli dəyişənlər eyni bir ifadədə iştirak edə bilərlər. Paskal dilində istifadə olunan digər tiplərdən biri də Sadalanan tipdir.

Sadalanan tip- verlənlərin nizamlanmış çoxluğundan ibarətdir. Sadalanan tiplərin yazılışı bölməsində göstərilir və TYPE açar sözü ilə bağlanır. Sadalanan tip elan edilərkən dəyişənin ala biləcəyi qiymətlər mötərizədə göstərilməlidir. Məsələn;

TYUPE

İxtisaslar= (menecment, marketing, mühasibat uçotu, informasiya sistemləri,iqtisadçı-mühəndis);

Bu o deməkdirki, ixtisaslar adlı identifikator proqram daxilində mötərizədəki 5 ixtisasın adlarından birini ala bilər

15. Sabitlər. Hesabi funksiyalar

Proqramlaşdırma dillərində əsas anlayışlardan biri sabit anlayışıdır.Proqram yerinə yetirilmə prosesində qiymətini dəyişməyən proqram parametrləri sabitlər adlanır.Turbo paskal dilində sabitlərin iki növü var: sadə və tipləşdirilmiş. Sadə

sabitlərdə tip onun qiyməti ilə təyin olunur. Tipləşdirilmiş sabitlərdə tip hökmən göstərilməlidir.

Sadə sabitlər tam,həqiqi,simvol,məntiqi,sətir(simvol) və çoxluq tipində ola bilər. Sadə sabitlər aşağıdakı kimi təsvir olunur.

Const

Identifikator=sabitin qiyməti;

Tam sabitlər. Tam sabitlərin təsvirində yalnız işarə və rəqəmlərdən istifadə olunur. Onaltılıq say sistemindəki tam ədədin qarşısında S işarəsi qoyulur. Turbo Paskalda əvvəldən təyin edilmiş iki tam sabit mövcutdur:

Maxint=32767;

MaxLongint=2147483647;

Sətir və simvol sabitlər. Bir sətirdə yerləşməklə birqat dırnaqla hüdudlanmış simvollar ardıcıllığı sətir simvolları adlanır. Kampilyator 126 simvoldan çox olmayan sətri qəbul edir. Bir simvoldan ibarət olan sətir simvol sabiti adlanır. Əgər birqat dırnaq içərisində heç bir simvol yoxdursa,onda bu sıfır sətir adlanır. Əgər sətirdə birqat dırnaq işarəsindən istifadə etmək lazımdırsa,onda onu iki dəfə yazmaq lazımdır

Hesabi funksiyalar. Turbo Paskalda ifadələrdə hazır element kimi istifadə olunan əvvəlcədən hazırlanmış alt proqram funksiyalar mövcutdur. Bunların sayı artırılmış və standart bi modulda yerləşdirilir. Turbo Paskal proqramlarında sabit,tip,dəyişən,prosedur və funksiyalardan istifadə edərkən onların təyin olduğu modullar təsvir olunmalıdır. İstifadəçi tərəfindən yaradılan modulların və sistem modulunun təsviri vacib deyil. Digər modullar mütləq təsvir olunmalıdır.

16.Tipin çevrilmə funksiyaları.

Bu funksiyalar tipin çevrilməsi üçün nəzərdə tutulub. Məs., simvolun ədədə, həqiqi ədədin tama və s. Bunlar aşağıdakı funksiyalardır:

Chr(x) –ASCII kodunun simvola çevrilməsi

Funksiyanın arqumenti 0..255 intervalında olmaqla tam olmalıdır. Nətic bu koda uyğun simvoldur. Məs., chr(97)-ni nəticəsi ‘a’-dır.

High(x)-kəmiyyətin maksimal qiymətinin tapılması.

Funksiyanın arqumenti sıra, sətir və massiv tipli parametr və ya identifikator ola bilər. Nəticə sıra tipi üçün bu kəmiyyətin maksimal qiyməti, massiv tipi üçün indeksin maksimal qiyməti və sətir tipi üçün sətrin təsvir olunmuş ölçüsü.

Low(x) – kəmiyyətin minimal qiymətinin tapılması.

Funksiyanın arqumenti sıra, sətir və massiv tipli parametr və ya identifikator ola bilər. Nəticə sıra tipi üçün bu kəmiyyətin minimal qiyməti, massiv tipi üçün indeksin minimal qiyməti və sətir tipi üçün isə 0-dır.

Ord(x) –istənilən sıra tipinin tam tipə çevrilməsi.

Funksiyanın arqumenti ixtiyari nizamlı (məntiqi, simvol və sadalanan) tip ola bilər. Nəticə Longint tipinin kəmiyyətidir. Məs., ord(‘a’) nəticəsi 97-dir.

Round(x)- həqiqi ədədin qiymətinin, bu ədədə yaxın olan tamadək yuvarlaqlaşdırılması.

Funksiyanın arqumenti həqiqi, nəticə isə Longint tipində olur.

Trunc(x) – həqiqi ədədin tam hissəsinin tapılması.

Funksiyanın arqumenti həqiqi, nəticə isə Longint tipində olur.

Sıra tipinin kəmiyyətləri üçün funksiyalar. Bu funksiyalar əvvəlki və ya sonrakı elementlərin tapılması, ədədin təkliyinin yoxlanması üçündür. Bu tipə aşağıdakı funksiyalar aiddir:

Odd(x) –x-in təkliyinin yoxlanması

Funksiyanın arqumenti Longint tipində, nəticə isə arqument tək olduqda True, cüt olduqda False olur.

Pred(x) – X-in əvvəlki qiymətinin təyini.

Funksiyanın arqumenti sıra tipin ixtiyari kəmiyyəti, nəticə isə həmin tipin əvvlki qiymətidir. Məs., Pred(2)-nin nəticəsi 1-dir.

Succ(x)-X-in sonrakı qiymətinin təyini.

Funksiyanın arqumenti sıra tipin ixtiyari kəmiyyəti, nəticə isə həmin tipin sonrakı qiymətidir. Məs., Succ(2)-nin nəticəsi 3-dir.

17. Proqramın strukturu

Pascal dilində yazılmış istənilən proqram iki hissədən – *verilənlərin təsviri bölümündən* və *proqramın gövdəsindən* ibarət olur.Proqramın mətnində verilən lərin təsviri proqramın gövdəsindən əvvəl gəlir.

Bu da dilin əsas qaydasından qaynaqlanır.Verilənlər *operatorlar* vasitəsilə emal olunur. Operatorlar barəsində növbəti dərsimizdə daha ətraflı danışacağıq.

Proqramın gövdəsi begin sözü ilə başlanır və operatorlar yığınından ibarət olur. O na görə də bu hissəyə *operatorlar bölümü* də deyilir. Bu bölüm end açar sözü ilə bitir (sonda nöqtə qoyulur). Pascal dilində proqram aşağıdakı şəkildə olur:

program < proqramın adı >;

< dəyişənlərin təsviri >

begin

< operatorlar >

end.

Proqram istənilən cür sətirlərə bölünə bilər – bununla onun mənası dəyişmir (təkcə sözlərin sətirdən-sətrə keçirilməsinə icazə verilmir). Buna görə də çalış-maq lazımdır ki, proqramlar mümkün qədər anlaşıqlı yazılsın.

Şərhlər. Proqram yazarkən siz nə etməli olduğunuzu bilirsiniz. Ancaq müəy-yən müddətdən sonra həmin proqrama qayıtmalı olsanız, qəribə olsa da, görəc ək- siniz ki, çox şeyi unutmusunuz. Buna görə də həm özünüzün xatırlamanız, həm də başqalarının sizin proqramı anlaması üçün proqramın müəyyən yerlərinə şərhlər vermək yaxşı olardı.

Adından da göründüyü kimi, *şərhlər* proqramın mətnini oxuyan şəxs üçün qeyddir. Şərhlərdən proqramın nə məqsədlə yaradıldığı, onun yaradıcısı haqqı nda məlumatı, proqramın son dəyişdirilmə tarixini, proqramdakı dəyişənlərin, funksiyaların təyinatını və s. göstərmək üçün istifadə edilə bilər.

Pascal dilində şərhlər (* və *) simvollarının, yaxud { və } fiqurlu mötərizələrin arasında yazılır. Proqram maşın koduna çevrilərkən bu simvollar arasında yazılanlar nəzərə alınmır.

18. DAXİLETMƏ VƏ XARICETMƏ OPERATORLARI.

Write və ya Writeln çıxış əmri:a Sabit, dəyişən və ya hesablamaların nəticələrini ekrana və ya fayla yazmaq üçün Pascal proqramlaşdırma dilində **Wrıte** və ya **Wrıteln** çıxış əmrindən istifadə olunur. Bu əmrə bəzən çap etmə əmri də deyilir. Ondan proqramın istənilən yerində istifadə etmək olar. Verilən sabit bir yazı olabiləcəyi kimi bir dəyişən də ola bilər. Eyni zamanda ixtiyari sayda parametr göstərmək olar. Bu parametrlər mötərizələr içərisində siyahı şəklində göstərilir. Bu zaman dəyişənin öz adı deyil, onun qiyməti yazılır. Bu əmrlərin yazılış forması aşağıdakı şəkildədir:

Yazılışı:

aaaaaaaaaaa Write (ifade və ya ifadələr)

aa və ya

aaaaaaaaaaa WriteLn (ifade və ya ifadələr)

Burada göstərilən ifadə sabit, dəyişən və ya hesablama ola bilər. İfade yerinə iki apostrof işarəsi arasında sözlər də yazıla bilər. İfadə bir neçə olarsa onlar arasında vergül işarəsi qoyulur.

Yazılışı:

Write(A1, A2, A3,..., An)

aaaaaaaaaaaa **Writeln**(A1, A2, A3,..., An)

İki əmra arasındaki fərq: **Write** aəmri nəticəni həmin sətirdə qalmasını, **Writeln** əmri isə yeni bir sətirə yazılmasını yerinə yetirir. **Write** əmrində isə eyni sətir üzərində qalar, bir kursor sonrakı sətrə keçməz. **write** və **writeln** cıxış əmrlərində çıxış formatını göstərmək olar.a **Writeln** əmrində heç bir parametr göstərilməyibsə, xaricetmə aparılmır, yalnız yeni sətrə keçid baş verir. Bu format çıxış qiymətindən sonra iki nöqtə

qoyularaq rəqəmlə və ya ifadə ilə göstərilir. Rəqəm və ya ifadə çıxış sahəsinin enini, daha doğrusu yazılacaq mövqenin qiymətini göstərir.

Qeyd:Proqramda **End** sonluğundan qabaq yazılmış **ReadIn** əmri nəticə göstərilən pəncərənin ekranda saxlanması üçündür. Enter düyməsini basdıqdan sonra pəncərə yox olacaq.

Read və ya ReadIn *giriş əmri:*a Klaviaturadan kompütürin yaddaşına informasiya yazmaq üçün giriş **Read** və ya **ReadIn** əmrindən istifadə olunur. Read ile ReadIn arasındakı fərq: **Read** əmri oxuma əmri tamamlandıqdan sonra kursorun həmin sətirdə qalmasını, **ReadIn** əmri isə yeni bir sətirə keririlməsini yerinə yetirir.Bu əmrdə birdəfəyə bir neçə dəyişənin qiymətini daxil etmək olar. Bu halda verilənlər bir-birindən ya УboşluqФ simvolu ilə ayrılmalı ya da hər veriləndən sonra Enter klavişi basılmalıdır. Bu zaman nə qədər ki,a əmrin göstərilən bütün əmrləri daxil edilməyib, növbəti əmrə keçid baş verməyəcək. Pascalda verilənlərin yaddaşa oxunması üçün **Read** və **ReadIn** əmrləri aşağıdakı şəkildə yazılır:

Yazılışı:

aaaaaaaaaa Read (dəyişən və ya dəyişənlər)

aa və ya

aaaaaaaaaa ReadLn (dəyişəna və ya dəyişənlər)

a

Burada:

aaaaaaaaaaaaa Dəyişən və ya dəyişənlər müxtəlif tipə malik ola bilərlər. Nəzərə almaq lazımdır ki, dəyişənin qiymətini daxil edərkən tipi düzgün süçmək lazımdır.

19. Sadə operatorlar

Operator qoyulmuş məsələnin həlli üçün özündə verilənlər üzərində bir sıra əməlləri yerinə yetirən alqoritmik dilin əmridir. Proqram mətnində operatorlar

sətirlərdə yazılır və hər bir sətir öz aralarında nöqtə vergüllə ayrılırlar. Yəni əmrin eləcə də digər əmrlərin sonu nöqtəli vergül simvolu ilə göstərilir. Əvvəlki bölümdə biz yalnız əmrlərin struktur yazılış qaydaları ilə tanış olduq. Bu bölümdə əmrlərin yerinə yetirdiyi əməliyyatların öyrənilməsi ilə məşğul olacağıq. Delphi mühitində istifadə olunan Objekt Pascal dilinin operatorlarını məntiqi olaraq sadə və strukturlaşdırılmış əmrlər qrupuna ayırmaq olar:

Sadə operatorlar: Tərkibində digər operatorlardan ibarət olmayan operatorlara *sadə operatorlar* deyilir. Sadə operatorlara mənsubetmə, şərtsiz keçid,a boş,a prosedura proqramların çağırılması operatoru aiddir.

Strukturlaşdırılmış operatorlar: Tərkibində digər operatorlardan istifadə edilən operatorlara strukturlaşdırılmış *aoperatorlar* deyilir. Strukturlaşmış operatorlar digər Begin və End ehtiyat sözləri arasında yazılmış əmrlər ardıcıllığından, şərti keçid əmrlərindən, dövrü operatorlardan və qoşma operatorlarından təşkil olunur.
Yəni strukturlaşdırılmış operatorlara budaqlanma (şərti), dövri (təkrar)a və müraciət operatorları aiddir.
Operator: Strukturlaşmış operatorlar bir-birindən nöqtə vergüllə ayrılan və *begin* və *end* ehtiyat sözləri ilə məhdudlaşan ixtiyari sayda istənilən əmrlər qrupundan ibarətdir.

Örnək:

beginaaaa
aaaaa Operator1 ;a
aaaa E;aaaa
aaaaa OperatorN ;

end;

Mənsubetmə əmri:a *Mənsubetmə operatoru* dilin əsas operatorudur. Bu ən sadə mənimsətmə əmridir. Bu əmr sağ hissədə verilmiş ifadənin hesablanmasını təyin edir və ifadənin nəticəsini, dəyişənin qiymətini və ya sabit ədədi başqa bir operatorun sol hissəsində yerləşən dəyişənə mənimsədir.

Dəyişən və ifadənin tipi uyğun olmalıdır, məsələn, həqiqi və tam (və ya əks). Fayl tipindən başqa verilənlərin istənilən tipi mənsubetməyə mümkün ola bilər. Burada əmr := işarəsi ilə yazılır. Qeyd etmək lazımdır ki, У := Ф mənsubetmə işarəsi У=У işarəsindən fərqlənir və başqa məna daşıyır. Mənsubetmə işarəsi onu göstərir ki, ifadənin qiyməti əvvəl hesablanır, sonar göstərilən dəyişənə mənsub edilir.a

dəyişən:= ifadə

Burada,aa **aУ:=Ф** mənsubetmə operatorudur, bərabərlik işarəsi deyildir. Nəzərə almaq lazımdır ki, sağ tərəfdəki ifadənin tipi sol tərəfdəki dəyişənin tipi ilə eyni olmalıdır. Yəni müxtəlif tipləri dəyişənə mənimsətmək olmaz. Belə hallarda bir tipi başqa tipə çevirən operatordan istifadə edərək hər iki tərəfi eyni tipə çevirmək lazımdır. Sonrakı bölümlərdə Delphi mühitində belə çevirmə operatorları haqqında ətraflı verəcəyik.

Dəyişənin adı əvəzinə massivin elementinin və ya yazı sahəsini göstərmək olar. Məsələn, əgər n - ədədi dəyişəndirsə və müəyyən qiymətə malikdirsə, onda düzgün konstruksiya aşağıdakı kimi olacaqdır:

```
Const
aaaaaaaa aPi=3.14a ;
Var aaax, y :aaaaaaa real;
aa aaaaaa n :aaaaaaaaaaaa integer;
aaaaaaaaa s :aaaaaaaaaaaa string;
Begin
aaaaaaaa n := 17 * n Ц 1;
aaaaaaaaa t := CTarixT + DateToStr (Date);
aaaaaaaa y := -12.3 * sin (pi / 4);
a aaaaaaak := 23.789E+3;
End;
```

20.Ön şərtli və son şərtli dövr operatoru

Sonrakı şərtli dövr operatoru. Sonrakı şərtli dövr operatoru başlıqdan -repeat, dövrün gövdəsindən və dövrün qurtarmasını müəyyən edən şərtdən ibarətdir və aşağıdakı yazılış formatına malikdir: Repeat Until ; Burada şərt məntiqi ifadədir. Operator yerinə yetirilərkən əvvəlcə repeat və until xidməti sözləri arasında olan operatorlar yerinə yetirilir, sonra isə dövrün qurtarması şərti yoxlanılır. Əgər məntiqi ifadənin qiyməti false olarsa, onda dövrün gövdəsini təşkil edin operatorlar təkrar yerinə yetirilir. Əgər məntiqi ifadənin qiyməti true olarsa, onda dövr sona çatır və dövr operatorundan sonrakı operator 361 yerinə yetirilir. Dövrün gövdəsini təşkil edən operatorlardan biri elə olmalıdır ki, o dövrün qurtarması şərtinə təsir edə bilsin. Əks halda, dövretmə sonsuz olaraq davam edə bilər.

İlkin şərtli dövr operatoru. İlkin şərtli dövr operatoru sonrakı şərtli dövr operatoruna oxşar əməliyyatı yerinə yetirir. Fərq yalnız odur ki, dövrün qurtarmasını müəyyən edən şərt dövrün gövdəsindən əvvəl gəlir. Yazılış formatı aşağıdakı kimidir: while do ; Burada məntiqi ifadə, isə sadə və ya mürəkkəb operatordur. Operator yerinə yetirildikdə ilk öncə əvvəl məntiqi ifadənin qiyməti hesablanır. İfadə true qiyməti aldıqda dövrün gövdəsini təşkil edən operatorlar yerinə yetirilir və yenidən məntiqi ifadənin qiyməti hesablanır və ifadə true qiyməti aldıqda proses təkrarlanır. Bu proses məntiqi ifadə false qiyməti alana qədər dövrü olaraq davam edir. Bundan sonra proqramda dövr operatorundan sonra gələn operator yerinə yetirilir. Qeyd edək ki, -in qiyməti yalan (false) olarsa, onda dövrün gövdəsini təşkil edən operatorlar bir dəfə də olsun yerinə yetirilmir. Lakin sonrakı şərtli dövr operatorunda dövrün gövdəsindən asılı olmayaraq ən azı bir dəfə yerinə yetirilir. Sonrakı şərtli dövr operatorunda olduğu kimi, ilkin şərtli dövr operatorunda da gövdəsini təşkil edən operatorlardan biri elə olmalıdır ki, o dövrün qurtarması şərtinə təsir edə bilsin.

21.Çoxluqlar

Çoxluq- bir-biri ilə müəyyən qaydada əlaqəli olan eyni tipli obyektlər yığımıdır.
Turbo Paskal dilində çoxluqlar aşağıdakı kimi elan edilir:
Ad= SET OF tip
Burada. Ad- çoxluğun adı; SET (çoxluq), OF (-ın, -in, -un, -ün)- açar sözləri; tip- çoxluq element-lərinin tipidir.
Məsələn:

```
VAR

A1, A2: SET OF 0..5;

A3, A4: SET OF 3..8;
```

Burada A1, A2, A3, A4 çoxluqların adıdır. A1 və A2 çoxluqları 0-dan 5-ə qədər rəqəmləri, A3, A4 çoxluqları isə 3-dən 8-ə qədər rəqəmləri özündə birləşdirir.
Çoxluqlar üzərində aşağıdakı əməliyyatları aparmaq olar:
*- çoxluqların kəsişməsi;
+ - çoxluqların birləşməsi
- - çoxluqların fərqi; birinci yazılmış çoxluğun ikinciyə aid olmayan elementlərindən ibarət çoxluq.
Çoxluqlar arasındakı münasibətləri öyrənmək məqsədilə aşağıdakı əməliyyatlardan istifadə edilir:
=- çoxluqların ekvivalentliyinin yoxlanması; <>-çoxluqların ekvivalent olmamasının yoxlanması;
<=-birinci çoxluğun ikinciyə daxil olmasının yoxlanması;
>=-ikinci çoxluğun birinciyə daxil olmasının yoxlanması;
in- elementin çoxluğa daxil olmasının yoxlanması: əgər element çoxluğa daxildirsə TRUE (doğru), əks halda isə FALSE (yalan) qaytarılır.

22. Massivlər

Turbo Paskal dilində istifadə olunan massivlər riyaziyyatdakı matrislər ardıcıllıqlar eləcə də cəbri vektorlarla eynidir. Massivlərin fərqləndirici xüsusiyyəti onun elementlərinin eyni tipli olmasıdır. Digər tərəfdən isə massivin elementləri adətən kəmiyyət göstəricilərindən ibarət olurlar.

Massivləri elan etmək məqsədilə aşaşıdakı ümumi formadan istifadə olunur:

Tipin adı = ARRAY [massivin olçü göstəriciləri] OF tip;

Burada, tipin adı- massivə verilən addır, massivin ölçü göstəriciləri-massivin elementlərinin sayını, sətir və sütunlarının sayını göstərən göstəricilərdir, tip-massiv element-lərinin tipidir və Turbo Paskal dilinin LONGİNT tipindən başqa istənilən tipi ola bilər. Adətən məsələlərin həllində ən çox rast gəlinən massivlər birölçülü və ikiölçülü massivlərdir. Başqa sözlə daha çox ardıcıllıq və ya matrislərdən istifadə olunur. Əgər A= ARRAY [1..14] of real; verilibsə, bu o deməkdir ki, massivin adı A-dır, onun 14 elementi var və elementlər həqiqi tipli ədədlərdir. Bu massivin elementləri yaddaşda ardıcıl yerləşir və müraciət vaxtı ünvana uyğun olaraq çağrılır. Elementlərə qiymət aşağıdakı kimi mənimsədilir: A[1]:=1.2; A[2]:=76.9 və sair. Əgər massiv iki ölçülüdürsə, onda o, aşağıdakı kimi göstəriləcək:

B= ARRAY [1..3,1..5] OF INTEGER;

Burada B-massivin adıdır, onun 3 sətri, 5 sütunu var və elementləri tam tiplidirlər.Bu şəkildə massivin elementləri aşağıdakı kimi mənimsədilir: B[1,2]:=2; B[3,4]:=24;

Bu o deməkdir ki. B ikiölçülü massivinin 1-ci sətrinin 2-ci sütununun elementi 2; 3-cü sətirinin 4-cü sütununun elementi isə 24-ə bərabərdir.

23. Prosedurlar.Funksiyalar.

İnsan həcmi bir neçə yüz sətir olan alqoritmdən “baş çıxara“ bilər. Proqram sətirlərinin sayı artdıqca işin ümumi məntiqi itir. Konkret operatorların yerinə yetirdiyi əməliyyatlar elementar olsa da, onların ümumi məqsədini başa düşmək

çətinləşir. Proqramın strukturu və onun yerinə yetirilmə ardıcıllığı aydın olmur. Belə alqoritmi dəyişdirmək, yaxud düzəltmək çox çətin olur.
Bu problemi həll etmək üçün alqoritm sadə əməliyyatları yerinə yetirən ayrıayrı alqoritmlərə bölünür. Belə ayrıca alqoritmlərə yardımçı alqoritmlər deyilir.
Proqramlaşdırma dillərində yardımçı alqoritm termininin yerinə altproqram terminindən
istifadə olunur. Yardımçı alqoritmə (altproqrama) müraciət etmək üçün
onu çağırmaq lazımdır.
Adətən orta ölçülü proqramları hər biri çox da çətin olmayan əməliyyatı ye - rinə yetirən kiçik altproqramlara bölürlər. Yekun alqoritm ayrı-ayrı operatorlardan
deyil, hər birinin öz adı olan bitkin kod bloklarından ibarət olur. Bu halda altproqramlara
proqramçıların təyin etdiyi yeni operatorlar kimi baxmaq olar.
Standart altproqramlar. Bir çox yardımçı alqoritmlərdən tez-tez və müxtəlif məsələlərdə istifadə edilir. Məsələn, tez-tez tipik riyazi funksiyaları hesab la - maq, yaxud sətirlər üzərində standart əməliyyatlar aparmaq tələb olunur. Belə alqoritmləri hər proqramçı özü yazsa idi, bu çox böyük vaxt itkisinə səbəb olardı. Bu problem standart altproqramlar tətbiq etməklə aradan qaldırılır.
Standart altproqramlar adətən proqramlaşdırma dilində deyil, proqramlaşdırma sistemində (mühitində) təyin edilir. Onlar translyatora əlavə edilən altproq - ramlar kitabxanasına daxil olur. Standart altproqramların geniş kitabxanaları işi əhəmiyyətli dərəcədə yüngülləşdirir.
Yardımçı alqoritmlərin tipləri. Altproqramlar adətən, iki kateqoriyaya
bölürlər: prosedurlar və funksiyalar.
Bəzi proqramlaşdırma dillərində (məsələn, C-də) altproqramları prosedur və funksiyalara bölmürlər. Onların hamısına funksiya kimi baxılır. Belə dillərdə prosedur heç bir qiymət qaytarmayan funksiyadır.
Prosedur, sadəcə hər hansı operatorlar ardı cıl lığını yerinə yetirir.
Funksiya isə müəy yən qiyməti hesablayır və həmin qiy məti çağıran proqrama (yaxud altproqrama) ötürür.

İmtahan sualları:

1.Alqoritm nədir?
2.Alqoritmin icraçısı.
3.Alqoritmin xassələri.
4.Diskretlik xassəsi.
5.Müəyyənlik xassəsi.
6.Kütləvilik xassəsi.
7.Sonluluq və nəticəvilik xassəsi.
8.Həll alqoritminin təsvir üsulları.
9.Həll alqoritmi.
10.Alqoritmin təsvir üsulları.
11.Tipik alqoritmik strukturlar.
12.Tipik hesablama proseslərinin alqoritmləşdirilməsi.
13.Alqoritmin sözlə təsvir üsulları.
14.Alqoritmik dillə təsvir üsulu.
15.Sxemlə təsvir üsulu.
16.Budaqlanan alqoritmik strukturlar.
17.Dövrü alqoritmik strukturlar.
18.Arqumenti monoton dəyişən funksiyanın qiymətlər çoxluğunun hesablanması.
19.Sərbəst dəyişən arqumentli dövri hesablama prosesləri.
20.Mürəkkəb dövri prosessor.
21.Proqramlaşdırma dilləri.
22.Alqoritmik təsvir üsulları.
23.Proqramlaşdırmanın dilləri.
24.Proqramlaşdırma dillərinin səviyyələri.
25.Prosedur proqramlaşdırma.
26.Məntiqi proqramlaşdırma.
27.Funksional proqramlaşdırma.
28.Obyekt-yönlü proqramlaşdırma.
29.Hadisə-yonlü proqramlaşdırma.
30.Vizual proqramlaşdırma.
31.Dilin əlifbası və elementləri.
32.Turbo Paskal dilində proqramlaşdırma.
33.Turbo-Paskal işləmə əmrləri.
34.Turbo- Paskalla dilin lüğəti.
35. Turbo-Paskal dilinin elementləri.
36.Turbo-Paskal dilinin əlifbası.
37.İdentifikatorlar.
38. Struktur proqramlaşdırma.
39.Standart tiplər.
40.Tiplərin təsnifatı.

41.Verilənlərin tipi.
42.Tam tiplər.
43.Həqiqi tiplər.
44.Məntiqi(Bul) tip.
45.Simvol tipi.
46.Sətir tipləri.
47.Göstərici tip.
48.Mətn tipi.
49.Dəyişənlər.
50.Sətir və simvol tipli dəyənlər.
51.Sətir tipli dəyişənlər.
52.Simvol tipli dəyişənlər.
53.Sabitlər.
54.Sadə sabitlər.
55.Sətir və simvol tipli sabitlər.
56.Tipləşdirirlmiş sabitlər.
57.Standart funksiyalar.
58.Tipin çevrilmə funksiyaları.
59.Əməliyyatlar.
60.Hesabi əməliyyatlar.
61.Nisbət əməliyyatları.
62.Məntiqi(Bul) əməliyyatlar.
63.İnformasiya bitləri üzrə əməliyyatlar.
64.Sətir əməliyyatı(konkatensiya).
65.Çoxluqlar üzrə əməliyyatlar.
66.Ünvan əməliyyatı.
67.Proqramın strukturu.
68.Proqramın parametrləri.
69.Sadə operatorlar.
70.Operatorlar.
71.Mənsubetmə operatoru.
72.Prosedura müraciət.
73.Şərtsiz keçid operatoru.
74.Baş operator.
75.Seçmə operatoru.
76.Parametrli dövr operatorlar.
77.Ön şərti dövr operatoru.
78.Son şərti dövr operatoru.

ƏDƏBİYYAT:

1. M.Alışov, Ə.Pələngov, Q.Əliyev “İnformatika” Bakı-2005
2. N.Cəfərov, N.Rəhimova“İnformatika” Bakı-2013
3. N.Mahmudov “Fərdi kompüterdə proqramlaşdırma” Bakı-1995