

A.İ.Qurbanov, E.M.Məmmədov, A.S.Hüseynova

# **Kompüter texnikası və proqramlaşdırma**

**Azad İsa oğlu Qurbanov,  
Elçin Musa oğlu Məmmədov,  
Aygün Seyfəddin qızı Hüseynova**

# **Kompüter texnikası və proqramlaşdırma**

Azərbaycan Respublikası Təhsil Nazirinin  
25.02.2010 tarixli 243 sayılı əmri ilə Ali  
məktəb tələbələri üçün dərs vəsaiti kimi  
təsdiq edilmişdir.

**Bakı -2010**

**Elmi redaktor:** BDU-nun “İnformatika” kafedrasının müdiri  
dosent M.S.Xəlilov

**Rəy verənlər:** Professor K.Ş.Məmmədov  
Dosent İ.Ə.Qurbanov  
Dosent N.S. Süleymanov  
H.Ə.Cəfərova

A.İ.Qurbanov, E.M. Məmmədov, A.S.Hüseynova **Kompüter texnikası və proqramlaşdırma**-Bakı:- “Təhsil” NPM, 2010.- 169 s.: şəkilli

Kitab Ali məktəblərin bakalavr təhsil pilləsi üçün hazırlanmışdır. Dərslikdə kompüter və telekommunikasiya texnikası, kompüter şəbəkəsində informasiya mübadiləsi, alqoritmlər, alqoritmik dillər, Turbo Pascal 7.0 alqoritmik dili haqqında mövzular şərh olunmuşdur.

## ÖN SÖZ

Tarixən informasiyanın ifadə edilməsi və yadda saxlanması tələbatı nitqin, yazının və təsviri incəsənətin yaranmasına səbəb olmuşdur. Sonralar informasiyanın saxlanması və mübadiləsi tələbatı kitab çapının və poçtun yaradılmasına gətirib çıxarmış, teleqraf, telefon, radio və televiziyanın yaranması böyük mətn informasiya selinin və təsvirinin işıq sürəti ilə ötürülməsinə imkan vermişdir. Lakin bu informasiyanın məqsədyönlü işlənməsi kompüter texnologiyası yaranana qədər yalnız insan tərəfindən yerinə yetirilirdi.

Kompüter texnologiyası böyük sürətlə inkişaf edərək, elmi-texniki, iqtisadi tərəqqinin, ictimai-siyasi münasibətlərin inkişafetdirici qüvvəsinə çevirilmişdir. Kompüter texnikasının tətbiqi iqtisadiyyatda istehsal texnologiyasının kökündən dəyişdirilməsinə, əmək məhsuldarlığının artmasına, insanların əmək şəraitinin yaxşılaşdırılmasına səbəb olur. Kompüter texnologiyası həmçinin qlobal informasiya mühitinin yaradılmasında, informasiyanın əmtəyə çevrilməsində, informasiya və bilik bazarının yaradılmasında və inkişafında, təhsil sisteminin təkmilləşməsində, beynəlxalq, milli və regional səviyyədə informasiya mübadiləsi sistemlərinin imkanlarının genişlənməsi hesabına cəmiyyətin peşə və ümumi mədəniyyət səviyyəsinin artmasında aparıcı rol oynayır.

Kompüterlərə mexaniki və yorucu işləri həvalə etməklə insan yaradıcı fəaliyyət üçün azad olunur. İstifadəçi kompüterlərdə tələb olunan məsələləri həll etmək üçün dəqiq başlanğıc informasiyalar, səhvsiz alqoritm və səmərəli proqram təminatına malik olmalıdır. Belə proqramların yaradılması isə kompüter texnikasının arxitekturasını və funksional imkanlarını, alqoritmik dilləri bilməyi və proqramlaşdırma vərdişinə malik olmağı tələb edir. Buna

görə də ali məktəblərin texniki və informasiya profilli fakültələrində kompüter texnikası və proqramlaşdırma fənninin tədrisi və onların tədrisinin keyfiyyət və kəmiyyət baxımından yüksəldilməsi müasir dövrün tələbidir. Bunun üçün də ilk növbədə müasir təlabata cavab verən dərslik və dərs vəsaitləri hazırlanmalı və nəşr olunaraq tədrisdə istifadə edilməlidir. Bu baxımdan, dosent A.İ.Qurbanovun, E.M.Məmmədovun və A.S.Hüseynovanın ali məktəblərin bakalavr təhsil pilləsi üçün birgə hazırladıqları «Kompüter texnikası və proqramlaşdırma» adlı dərsliyi mühüm əhəmiyyət daşıyır. Dərslik eyni adlı fənnin profilinə və fənn proqramına uyğun yazılmışdır. Dərslikdə kompüter və telekommunikasiya texnikası, kompüter şəbəkəsində informasiya mübadiləsi, alqoritmlər, alqoritmik dillər, əsas tədris alqoritmik dili hesab olunan Turbo Pascal 7.0 alqoritmik dili haqqında mövzular şərh olunmuşdur. Dərslik həddən artıq oxunaqlı olub, sadə dildə yazılmış, şərh olunan mövzularda çoxsaylı misallar, şəkillər, qrafik və cədvəllər verilmişdir.

*İsmayılova Nigar Nurəddin qızı*

## GİRİŞ

XX əsrin ən böyük kəşflərindən biri hesab edilən kompüter texnologiyası bəşəriyyətə hələ məlum olmayan böyük bir sürətlə inkişaf edərək, cəmiyyətin inkişafetdirici qüvvəsinə, həyatımızın ayrılmaz tərkib hissəsinə çevrilmişdir. Bu gün əqli fəaliyyətin, iqtisadiyyatın elə bir sahəsi yoxdur ki, orada kompüter texnologiyası tətbiq edilməsin. Bu konkret fəaliyyət növünü avtomatlaşdıran tətbiqi proqramlar və müvafiq kompüter və telekommunikasiya avadanlıqlarının yaradılması, informasiya şəbəkələrinin təşəkkül tapması ilə mümkün olmuşdur.

Aydındır ki, kompüter texnologiyasının inkişafı və tətbiqi bu texnologiyanı öz peşə fəaliyyətində tətbiq etməyi bacaran peşəkar kadrların hazırlanmasını aktual edir. Bu isə ilk növbədə ali və orta-ixtisas təhsilinin müasir səviyyəyə cavab verməsini tələb edir. Yüksək səviyyəli təhsil isə müvafiq dərslər, dərslər və metodik vəsaitlərin, texniki avadanlıqların, yeni elmi informasiya təminatının olmasını tələb edir. Son illər ölkəmizdə təhsilin maddi-texniki, elmi-informasiya bazasının möhkəmləndirilməsi sahəsində böyük işlər görülür. Xüsusilə, müasir tələblərə cavab verən ana dilində dərslərlərin və dərslər vəsaitlərinin nəşrinə xüsusi diqqət yetirilir. Azərbaycan təhsilinin flaqmanı olan Bakı Dövlət Universitetinin əməkdaşları tərəfindən hazırlanmış “Kompüter texnikası və proqramlaşdırma” adlı dərslər bu sahədə atılan addımlardan biridir.

Təqdim olunan dərslər ali məktəbin bakalavr pilləsi üçün nəzərdə tutulmuşdur. Kitab BDU-nun “Kitabxanaçılıq-informasiya” fakültəsində tədris olunan eyniadlı fənn proqramı əsasında yazılmışdır.

Dərslər iki fəsildən ibarətdir. Birinci fəsildə kompüter texnikasının inkişaf yoluna nəzər salınır, kompüterlərin təsnifatı verilir. Fərdi kompüterlərin arxitekturası və tərkib

hissələri haqqında məlumat verilir, fərdi kompüterlərin proqram təminatının təsnifatı şərh olunur. Birinci fəsilə həmçinin alqoritmlə anlayışı, alqoritmin xassələri və strukturu izah olunur, alqoritmik dillər haqqında məlumat verilmiş, kompüter şəbəkələrinin təsnifat, topologiyası, arxitekturası, şəbəkə avadanlıqları şərh olunmuşdur.

Dərsləyin ikinci fəsilə əksər dünya ölkələrində proqramlaşdırmanın əsaslarını öyrətmək üçün əsas öyrədici alqoritmik dil kimi istifadə olunan Turbo Pascal 7.0 alqoritmik dili şərh olunmuşdur. Dilin struktur elementləri- əlifbası, işçi sözlər, standart adlar, identifikatorlar, ifadələr, proqramın strukturu, sabitlər və dəyişənlər, onların tipləri və proqramda elan olunması şərh olunmuşdur. Dilin operatorları, onların proqramda istifadə qaydaları misallarla izah edilmiş, prosedural və funksiya altproqramları, massivlər, çoxluqlar, fayllar, obyektlər haqqında məlumat verilmişdir. İkinci fəsilə həmçinin göstəricilər, modullar, standart prosedural və funksiyalar, dilin qrafiki imkanları mövzuları verilmişdir.

Qeyd edək ki, müəlliflər izah etdikləri mövzuları sadə və geniş oxucu kütləsinin başa düşəcəyi dillə şərh etməyə çalışmışlar və ümid edirlər ki, təkcə tələbələr deyil, geniş oxucu kütləsi, xüsusilə proqramlaşdırmanın əsaslarını öyrənmək istəyən hər bir şəxs bu kitabdən özü üçün maraqlı informasiya əldə edə biləcəkdir.

# **Fəsil 1**

**Kompüterlərin texniki-proqram  
təminatı**



## **BU FƏSİLDƏ**

- ◆ Kompüterlər və onların inkişaf tarixi
- ◆ Kompüterlərin təsnifatı
- ◆ Masaüstü fərdi kompüterin tərkib hissələri
- ◆ Alqoritm anlayışı, alqoritmin əsas xassələri, tipləri və təsvir üsulları.
- ◆ Kompüterlərin proqram təminatı
- ◆ Alqoritmik dillər
- ◆ Kompüter şəbəkələri

## 1.1 Kompüterlər və onların inkişaf tarixi

Kompüterlər müvafiq proqram təminatı vasitəsilə rəqəmli informasiyanın yaradılmasını, emal edilməsini, etibarlı saxlanılmasını və operativ mübadiləsini təmin edən universal qurğudur. Başqa sözlə, kompüterlər informasiya ilə işləmək üçün nəzərdə tutulmuş proqramla idarə olunan elektron qurğulardır. Kompüterlərin yaradılması və istehsalı ilə müasir sivilizasiyanın ən mühüm nailiyyətlərindən biri hesab olunan kompüter texnologiyası meydana gəlmişdir. Kompüter texnologiyası öz növbəsində bəşər tarixində analoqu olmayan böyük sürətlə inkişaf edərək, elmi-texniki, iqtisadi tərəqqinin, ictimai-siyasi münasibətlərin inkişafetdirici qüvvəsinə çevrilmişdir. Məhz onun sayəsində cəmiyyət öz inkişafında yeni mərhələ hesab olunan – informasiyalaşmış vətəndaş cəmiyyəti quruculuğuna qədəm qoymuşdur. Kompüter texnologiyası informasiyalaşmış vətəndaş cəmiyyətinin əsas göstəricisi olan qlobal informasiya mühitinin yaradılmasında, informasiyanın əmtəyə çevrilməsində, informasiya və bilik bazarının yaradılmasında və inkişafında, təhsil sisteminin təkmilləşməsində, beynəlxalq, milli və regional səviyyədə informasiya mübadiləsi sistemlərinin imkanlarının genişlənməsi hesabına cəmiyyətin peşə və ümumi mədəniyyət səviyyəsinin artmasında aparıcı rol oynayır.

İlk kompüterlər böyük həcmli və yüksək dəqiqlik tələb edən hesablamalar aparmaq, hesablama proseslərini avtomatlaşdırmaq zərurətindən yaradılmışdır<sup>1</sup>. Onların istehsalından yetmiş ilə yaxın vaxt keçməsinə baxmayaraq, bu müddət ərzində kompüterlərin istehsalı çox böyük kəmiyyət və keyfiyyət dəyişikliyinə məruz qalmışdır. Kompüterlər böyük həcmli, xidmət üçün çoxsaylı yüksək ixtisaslı işçi heyəti tələb edən və yalnız hesablama əməliyyatı aparmağa qadir olan,

---

<sup>1</sup> Kompüter sözü ingilis dilindən tərcümədə hesablayıcı deməkdir.

istehsalı və istismarı böyük xərc tələb edən “lampalı nəhənglərdən” kiçik ölçülü, kütləvi istehsal edilən, maya dəyəri az, insan fəaliyyətinin bütün sahələrinə nüfuz etmiş bir “əmək alətinə” çevrilmişdir. Kompüterlərin sürəti, böyük həcmli informasiyaları qəbul etmək, saxlamaq və emal etmək qabiliyyəti kəskin sürətdə yüksəlmiş, onlar istifadəçi ilə əlverişli ünsiyyət, kommunikasiya xidməti, özünü diaqnostika, nasazlıqların aradan qaldırılması, optik obrazların və insan nitqinin tanınması, multimedia və s. imkanlar əldə etmişlər.

Kompüterlərin əsas iş prinsipləri XX əsrin 40-cı illərində Amerika alimləri Con Fon Neyman, Q.Qoldsteyn və A.Beris tərəfindən verilmişdir. Həmin prinsiplər 1946-cı ildə ABŞ-ın Pensilvaniya ştatında Con Mokli və Presper Ekkertın rəhbərliyi altında ENIAC<sup>1</sup> adlı universal kompüterin yaradılması ilə həyata keçirilmişdir. Məhz həmin tarix kompüterlərin yaranma tarixi hesab olunur. Lakin, qeyd etmək lazımdır ki, bir sıra ədəbiyyatlarda ilk kompüterin 1943-cü ildə məşhur alim Allan Turingin rəhbərliyi altında Böyük Britaniyada yaradıldığı və ikinci dünya müharibəsində alman məxfi məlumatlarının “oxunması” üçün istifadə edilən «Collos» kompüterinin olması iddia olunur. Lakin əksər alimlər «Collos» kompüterinin texniki imkanlarının zəif olması səbəbindən onu yalnız kompüter yaradılmasında atılan önəmli addımlardan biri hesab edərək, “kompüter erasının” məhz ENIAC kompüterilə başladığını qəbul edirlər.

Kompüterlər sürətli təkamül yolu keçmişlər və bu gün də təkmilləşməkdə davam etməkdədirlər. Kompüterlərin təkamülü beş mərhələ ilə xarakterizə olunur:

**I nəsil kompüterlərin istehsal olunması.** Bu mərhələ ENIAC kompüterinin yaradılması ilə başlayır və 1946-1958-ci illəri əhatə edir. Birinci nəsil kompüterlərin element bazası çoxsaylı elektron lampalar idi. Onlar elmi-texniki məsələlərin həlli üçün istifadə olunurdu. Həddən artıq baha olduğundan

---

<sup>1</sup> Electronic Numerical Integrator and Computer

əsasən hərbi-sənaye kompleksində və dövlət təşkilatlarında istifadə olunurdu. Bu kompüterlər böyük həcmə və kütləyə malik idi. Elektron lampalar böyük həcmli elektrik enerjisi sərf edirdi və tez qızırdı. Buna görə də əlavə soyutma sistemindən istifadə olunurdu. Məsələn, ENIAC kompüterində 178468 elektron lampası, 7200 kristallik diod, 4100 maqnit elementləri istifadə olunmuşdur və o, 30 ton ağırlığında olub, 200 kvadrat metr sahəni tuturdu, 150 Kvt elektrik enerjisi sərf edirdi<sup>1</sup>.

Birinci nəsil kompüterlər yalnız saniyədə 10-20 min əməliyyat yerinə yetirə bilirdi. Belə ki, adicə loqarifmik və triqonometrik funksiyaların hesablanması üçün bir dəqiqədən artıq vaxt tələb olunurdu. Əməliyyatlar yalnız ardıcıl – bir əməliyyat qurtardıqdan sonra digər əməliyyat yerinə yetirilirdi. Elektron lampaların istismar müddəti qısa olduğundan kompüterlər texniki cəhətdən də etibarlı deyildi. Kompüterlərin operativ yaddaşı zəif olub, 512-2048 bayt<sup>2</sup> təşkil edirdi. Operativ yaddaş qurğusu olaraq ilk vaxtlar civə doldurulmuş kicik diametrlili borulardan, sonra isə ferromaqnitlərdən istifadə olunmuşdur.

Birinci nəsil kompüterlər üçün proqramlar konkret kompüter üçün unikal olan “maşın dilində” tərtib edilirdi. Bu proqramlar həddən artıq mürəkkəb olub, onları yazmaq üçün proqramlaşdırıcıdan kompüterin arxitekturasını dəqiq bilməyi tələb edirdi. Proqramın daxil olunması və yerinə yetirilməsi idarəetmə blokunun açarları vasitəsilə həyata keçirilirdi. Bu proses mürəkkəb olduğundan və proqramda meydana çıxan səhvlərin aradan qaldırılması zərurəti proqramın daxil edilməsi və yerinə yetirilməsi prosesinin proqramlaşdırıcı tərəfindən yerinə yetirilməsini tələb edirdi.

Bu nöqsanlara baxmayaraq birinci nəsil kompüterlər o dövr üçün elmi-texniki tərəqqinin ən böyük nailiyyəti hesab olunur və mürəkkəb, əvvəllər həlli mümkün hesab edilən

---

<sup>1</sup> ENIAC kompüterini 9 il – 1955-ci ilə qədər fəaliyyət göstərmişdir.

<sup>2</sup> Bayt informasiyanın ölçü vahididir.

elmi, mühəndis məsələləri həll etdi. Bunun nəticəsində bir sıra elm və sənaye sahələrində fundamental nəticələr əldə olundu. Xüsusilə, atom fizikası və energetikası, riyaziyyat, astronomiya və hərbi-sənaye kompleksi sürətlə inkişaf etməyə başladı.

Qərbi dünyası ilə müqayisədə keçmiş SSRİ-də, o cümlədən Azərbaycanda kompüterlər bir qədər gec, keçən əsrin 50-ci illərinin əvvəllərindən başlayaraq istifadə edilmişdir. Sovet sənayesi MESM<sup>1</sup>, BESM, Strela, M-1, M-2, M-3, Minsk-1, M-20, Ural-1, Razdan və s. birinci nəsil kompüterlər istehsal etmişdir.

**II nəsil kompüterlərin istehsalı.** Bu mərhələ 1955-1967 –ci illəri əhatə edir. İkinci nəsil kompüterlər element bazası əsasən yarımkeçiricilərdən ibarət kompüterlərdir. Bu kompüterlərdə elektron lampaları Bell Laboratories (ABŞ) firmasında Nobel mükafatı laureatları Uilyam Şokli, Uolter Bratteyn və Con Bardin tərəfindən yaradılmış tranzistorlar əvəz etdi. Tranzistorlar kiçik ölçüyə malik olub, daha davamlı, elektrik enerjisinə az tələbkar idi və təqribən 40 ədəd elektron lampanı əvəz edə bilirdi. Bunun nəticəsində kompüterlərin həcmi və kütləsi dəfələrlə kiçildi, elektrik enerjisinə tələbat azaldı, maya və istismar dəyəri ucuzlaşdı. Eyni zamanda məhsuldarlığı, funksional imkanları, etibarlığı on dəfələrlə yüksəldi. Məsələn, Mark-1 adlı birinci nəsil kompüter  $2,5 \times 17 m$  ölçüyə malik olub, maya dəyəri 500 min dollar idisə, PDP-1 ikinci nəsil kompüterini artıq məişət soyuducusu həcmində olub 20 min dollara satılırdı.

İkinci nəsil kompüterlərdə displeylərdən istifadə edilməyə başlandı. Bu informasiyanın etibarlı qorunmasını təmin etməklə, informasiyanın daxil və xaric olunmasını xeyli

---

<sup>1</sup> MESM (Малая Электронная Счетная Машина) 1950-ci ildə S.A. Lebedevin rəhbərliyi altında Ukrayna Elmlər Akademiyasının Elektrotexnika İnstitutunda hazırlanmışdır və SSRİ-də yaradılan ilk komputer hesab edilir.

asanlaşdırdı. Operativ yaddaşın həcmi artaraq minimum 32 kbyat<sup>1</sup> oldu.

İkinci nəsil kompüterlər üçün xüsusi sistem proqram təminatı, o cümlədən informasiyanın paket emalı sistemləri, sonralar əməliyyat sistemləri yaradıldı. Kompüterin arxitekturasından asılı olmayan yüksək səviyyəli alqoritmik dillərin<sup>2</sup> (Fact, MathMatic, Algol, Fortran, Kobol və s.) yaranması və onlar üçün müvafiq kompilyatorun<sup>3</sup> və standart altproqramlar kitabxanasının yaradılması ilə proqramlaşdırılma olduqca asanlaşdı və həll olunan məsələlərin əhatə dairəsi genişləndi.

Kompüterlərin ucuzlaşması və proqram təminatının inkişafı onların tətbiq sahəsini xeyli genişləndirdi. Kompüterlər artıq elmi-texniki hesablamalar aparmaqla yanaşı maliyyə-iqtisadi, istehsal-texnoloji proseslərin avtomatlaşdırılması üçün tətbiq edilməyə başlandı. Artıq onlar daha çox biznes müəssisələrində, təhsil və layihələndirmə mərkəzlərində geniş istifadə olundu. Azad kompüter bazarı formalaşdı, kompüter istehsalı sürətlə artdı. Bir sıra böyük firmalar, məsələn, IBM, CDC, DEC və s. kompüter istehsalına başladı.

Keçmiş SSRİ-də BESM-3,4, Ural-11,14,16, Minsk-22, M-222, Mir, Nairi ikinci nəsil kompüterlər istehsal olunmuş və istifadə edilmişdir. Onların məhsuldarlığı saniyədə 50-100 min əməliyyat olmuşdu.

**III nəsil kompüterlərin istehsalı.** Bu mərhələ 4-1975-ci illəri əhatə edir. Üçüncü nəsil kompüterlərin element bazalarını inteqral sxemlər və ya mikrosxemlər təşkil edir<sup>4</sup>. İlk mikrosxemlər 1958-ci ildə Cek Kilbi və Robert Noys tərəfindən yaradılmışdır. Mikrosxemlər təqribən 10  $\text{mm}^2$  ölçüyə malik olub, on min tranzistoru əvəz etməyə qadirdir. Bu isə

<sup>1</sup> Kbyat informasiyanın ölçü vahididir və 1024 bayta bərabərdir.

<sup>2</sup> Bax: Alqoritmik dillər.

<sup>3</sup> Bax: Proqram təminatı.

<sup>4</sup> Bir sıra ingilis dilli ədəbiyyatlarda mikrosxem əvəzinə Chip işlədilir.

kompiuterlərin sürətlə miniatürləşməsini və ucuzlaşmasını təmin etdi. Bununla yanaşı üçüncü nəsil kompiuterlərin məhsuldarlığı kəskin yüksəldi. Artıq kompiuterlərdə 100 kilobaytlarla ölçülən operativ yaddaşlar tətbiq olunurdu, onlar saniyədə 10 milyon əməliyyat yerinə yetirə bilirdi. Miniatürləşmə ilə paralel olaraq kompiuterlərin arxitekturasında köklü dəyişiklik baş verdi. Xarici yaddaş qurğusu olaraq maqnit disklərindən və diski oxuyan qurğulardan-diskovodlardan istifadə edildi. Bir sıra xüsusi təyinatlı kompiuterlər məsələn, kosmik raketlərdə, aviasiyada və gəmilərdə istifadə edilən bortkompiuterlər meydana gəldi. Bununla yanaşı az - təqribən 4 kilovat enerji sərf edən mini kompiuterlərin istehsalına başlandı. Məsələn, ABŞ-da RDR, VAX mini kompiuterləri və SSRİ-də isə onların analoqu olan CM-1/2/3/4/1420 və s. istehsal olundu. Mini kompiuterlər sənayedə texnoloji proseslərin idarə edilməsində, məlumat banklarının yaradılmasında və informasiyanın idarə edilməsində geniş istifadə olunurdu. Üçüncü nəsil kompiuterlər seriya şəklində istehsal olunurdu. ABŞ-da İBM 360, SSRİ və Şərqi Avropanın keçmiş sosialist ölkələrində EC seriyalı kompiuterlər istehsal olunurdu.

Üçüncü nəsil kompiuterlərdə ilk dəfə olaraq multi-proqram rejimi tətbiq olundu. Bu rejim paralel olaraq bir neçə istifadəçinin proqramının yerinə yetirilməsini təmin etdi.

**IV nəsil kompiuterlərin istehsalı.** Bu mərhələ böyük inteqral sxemlərin kompiuter istehsalında tətbiqi ilə 1974-1976 –cı ildən başlamışdır. Böyük inteqral sxemlər bir kristal üzərində minlərlə tranzistor və digər elektron komponentləri birləşdirməyi təmin etdi. Bunun nəticəsində əvvəllər ayrı-ayrı elementlərdən ibarət olan kompiuterin funksional qurğularını böyük inteqral sxemlər əvəz etdi. Məsələn, böyük inteqral sxemlərdən ibarət olan mikroprosessorlar, xarici qurğular kontrollerləri, operativ yaddaş və digər qurğular yaradıldı. Bunun nəticəsində kompiuterlərin məhsuldarlığı və etibarlılığı on

dəfələrlə yüksəldi. Böyük inteqral sxemlərə keçid kompüterin qurğularını kiçik ölçülü bir lövhə-“ana lövhə” üzərində asanlıqla yerləşdirməyi mümkün etməklə birlövhəli kompüterlərin istehsalını təmin etdi. Altair adlı ilk mikro kompüter 1974-75-ci ildə Intel 8080 mikroprosessoru əsasında istehsal olundu və bununla da insanlar ucuz, az enerji sərf edən, istifadə üçün həddən artıq rahat və əlavə işçi heyəti tələb etməyən, şəxsi istifadə üçün nəzərdə tutulmuş və bu səbəbdən fərdi kompüterlər adlanan kompüterlər əldə etdilər. 1981-ci ildə IBM firması fərdi kompüterlərin kütləvi istehsalına başladı və mikroprosessorlar istehsal edən Intel firması ilə əməkdaşlıq edərək fərdi kompüterlərin əsas istehsalçısına çevrildi.

Mikroelektronikada əldə edilən uğurlar və müasir proqram təminatının yaradılması sayəsində fərdi kompüterlərin sürətli təkamülü təmin olunmaqdadır. Bu gün çoxsaylı firmalar müxtəlif konfigurasiyalı və təyinatlı fərdi kompüterlər istehsal edirlər və insan fəaliyyətinin bütün sahələrində tətbiq olunurlar.

**V nəsil kompüterlərin yaradılması.** Bu mərhələ müasir dövrü əhatə edir və məqsəd yeni və ən yeni elektron texnologiyalara əsaslanan müasir və gələcəyin kompüterlərinin istehsalını təşkil etməkdir. V nəsil kompüterlər çox yüksək məhsuldarlığa və etibarlılığa malik olmaqla, keyfiyyətə yeni funksional tələblərə, o cümlədən sözlə, biliklər bazaları ilə işləməyə, süni intellekt sistemlərinin təşkilinə, istifadəçi ilə nitq və görmə vasitəsi ilə ünsiyyəti təmin etməyə, ən yeni proqram vasitələrinin yaradılması prosesini sadələşdirməyə və s. imkan verməlidirlər. Hal-hazırda neyron, qeyri-səlis məntiq - Fuzzu nəzəriyyəsinə əsaslanan yeni texnologiyaları və nano texnologiyaları tətbiq etməklə bio və optik neyrokompüterlərin yaradılması nəzərdə tutulur.



## 1.2 Kompüterlərin təsnifatı

Kompüterləri təyinatına, uyğunluq parametrlərinə, ölçülərinə görə təsnifatlaşdırmaq mümkündür. Kompüterlərin təyinat üzrə təsnifatı daha geniş kompüterlər sinfini əhatə edən təsnifat formasıdır. Bu təsnifata görə kompüterlər tətbiq olunma sahələrinə görə qruplaşmışlar. Təyinatı üzrə təsnifat prinsipinə görə kompüterlər aşağıdakı qruplara bölünür:

- Mainfreymlər,
- Mini-kompüterlər,
- Mikrokompüterlər və ya fərdi kompüterlər.

Mainfreymlər çox böyük müəssisələrdə istifadə edilərək, elektron informasiyanın mərkəzləşmiş saxlanılmasını və paralel işlənilməsinə təmin edir. Bu sinif kompüterlərin istehsalına 1964 –cü ildə IBM/360 kompüterinin istehsalı ilə başlanmışdır və onların əsas istehsalçıları IBM, Amdahl, ICL, Siemens Nixdorf və digər firmalar hesab olunur. Mainfreymlərin IBM 370, IBM ES/9000, Cray 3, Cray 4, VAX-100, Hitachi, Fujitsu VP2000 modelləri geniş yayılmışdır. Bu gün hərbi-sənaye və biznes məlumatlarının 70% mainfreymlərdə emal olunur. Təkcə dünyanın 25 000 təşkilatında IBM firmasının istehsal etdiyi mainfreymlərdən istifadə olunur.

Müasir mainfreymlər saniyədə on milyardlarla əməliyyat yerinə yetirən, paralel işləyən, 64 tərtibli, bir neçə mərkəzi və perferiya prosessorundan ibarət çoxprosessorlu sistemdir. Hesablama əməliyyatı mərkəzi prosessorlarda yerinə yetirilir. Perferiya prosessorları isə çoxsaylı perferiya qurğularını idarə edir. Mainfreymlər çox istifadəçi rejimində fəaliyyət göstərir. Belə ki, mainfreymin əsas blokuna <sup>1</sup>

---

<sup>1</sup>Mərkəzi ayrıca otaqda yerləşir. Bu otaqda temperatur və rütubət normaları gözlənilir, elektromaqnit maneələrdən, tozdan və tüstüdən yüksək qorunma həyata keçirilir.

terminallar<sup>1</sup> qoşulur ki, onlar vasitəsilə qəbul edilmiş 1000-dən artıq istifadəçi məsələsini paralel olaraq eyni zamanda yerinə yetirmək mümkün olur.

Mainfreymlər həddən artıq etibarlı kompüterlərdir. Onların dayanmadan işləmə müddəti 12-15 il hesab olunur. İnformasiyalar aparat<sup>2</sup> və proqram<sup>3</sup> vasitələri ilə etibarlı qorunur.

Mainfreymlərin istismar xərci və maya dəyəri fərdi kompüterlərə nisbətən dəfələrlə çoxdur. Onlara xidmət çox saylı ixtisaslı işçi heyət – sistem proqramlaşdırıcılarından<sup>4</sup>, tətbiqi proqramlaşdırıcılardan<sup>5</sup>, mühəndis-texniklər<sup>6</sup> və verilənləri hazırlayan operatorlardan<sup>7</sup> ibarət kadr potensialı tələb edir. Buna görə də onların istismarı vaxtın optimal bölünməsi prinsipi əsasında həyata keçirilir. Belə ki, daha çox zəhmət tələb edən və uzun sürən hesablamalar, yerinə yetirilən məsələlərin sayı minimum olduğu gecə saatlarına planlaşdırılır. Gündüz vaxtlarında isə mainfreym daha az zəhmət tələb edən çoxsaylı məsələləri çoxistifadəçi rejimində icra edir. Bu rejimdə eyni vaxtda bir neçə məsələ ilə və uyğun olaraq bir neçə istifadəçi işləyir. Kompüter bu məsələləri paralel olaraq yerinə yetirir.

**Minikomputerlər.** Minikomputerlər mainfreymlərə nisbətən kiçik ölçülərinə görə, uyğun olaraq daha az

---

<sup>1</sup> Terminal monitordan və klaviaturadan ibarətdir.

<sup>2</sup> Məsələn, kriptografik qurğular vasitəsilə.

<sup>3</sup> Məsələn, RACF və ya VM:SECURE proqramları vasitəsilə

<sup>4</sup> Sistem proqramlaşdırıcıları hesablama sistemlərinin proqram-aparat interfeysini təmin edir. Onlar mainfreymlərin baza proqram təminatının işlənilib hazırlanması, sazlanması və istismarı ilə məşğul olur.

<sup>5</sup> Tətbiqi proqramlaşdırıcılar qrupu hesablama sistemlərinin istifadəçi interfeysini təmin edir. Onlar verilənlərlə konkret əməliyyatları yerinə yetirmək üçün proqramlar yaratmaqla məşğul olurlar.

<sup>6</sup> Mühəndis–texniklər təminat qrupu mainfreymlərə texniki xidmətlə, təmir və qurğuların sazlanması ilə, həmçinin başqa bölmələrin işləməsi üçün zəruri olan yeni qurğuların qoşulması ilə məşğul olur.

<sup>7</sup> Operatorlar tətbiqi proqramlaşdırıcıların yaratdığı proqramların işləyəcəyi verilənlərin hazırlanması ilə məşğul olurlar.

məhsuldarlığına və qiymətinə görə fərqlənirlər. Belə kompüterlər əsasən sənaye, elmi, tədris müəssisələrində, banklarda və orta biznes kateqoriyasına aid olan firmalarda istifadə olunur. Minikomputerlər multiprocessorlu arxitekturalara, çoxsaylı diskovodlara<sup>1</sup> və periferiya qurğularına malik olub, 200-ə qədər terminalın qoşulmasını təmin edir. Onların istismarı üçün də ixtisaslı proqramlaşdırıcılar, mühəndis heyəti tələb olunur, lakin onların sayı mainfreymələrə xidmət heyətindən dəfələrlə az olur.

### **Mikrokomputerlər və ya fərdi kompüterlər.**

Mikrokomputerlər maya dəyərinin, istismar xərclərinin az olması, onlarla işləmək rahat və sadə olması səbəbindən bütün tip müəssisələrdə fərdi olaraq istifadə olunur. Müəssisədə mikrokomputerlərin sayı kifayət qədər çox olduqda və kompüterlər lokal şəbəkəyə<sup>2</sup> qoşulduqda kompüterlərə və şəbəkəyə azsaylı - tərkibi bir neçə adamdan ibarət olan qrup xidmət edir. Onlar informasiya təhlükəsizliyini təmin edir və sistem-texniki xidmətlər göstərirlər.

### **Mikrokomputerlərin təsnifat növləri**

**İxtisaslaşma səviyyəsi üzrə təsnifat.** İxtisaslaşma səviyyəsi üzrə mikrokomputerlər iki yerə bölünür: universal və ixtisaslaşmış. Universal kompüterlər müvafiq proqram təminatı olduqda ixtiyari tətbiqi məsələni həll etməyə xidmət edir.

İxtisaslaşmış mikrokomputerlər konkret təyinatlı məsələlərin həlli üçün nəzərdə tutulmuşdur. Belə kompüterlərə avtomobillərin, gəmilərin, təyyarələrin, kosmik aparatların bort kompüterlərini, tibbi avadanlıqlar qoşulmuş minikomputerləri, serverləri aid etmək olar.

**Forma ölçülərinə görə təsnifat.** Mikrokomputerlərin forma ölçülərinə görə aşağıdakı kimi təsnifatı aparılır:

- Masaüstü (desktop),

---

<sup>1</sup> Diskovod- maqnit diskli informasiya daşıyıcılarını oxuyan qurğudur.

<sup>2</sup> Bax: Kompüter şəbəkələri.

- Portativ (Portable),
- Cib (Palmtop)

Masaüstü minikompyuterlər funksional imkanlarının böyüklüyünə, maya dəyərinin ucuz olmasına, daha əvvəl və kütləvi istehsalına görə daha geniş yayılmışdır. Onlar əsasında Avtomatlaşmış İşçi Yerləri yaradılır, fərdi olaraq bir əmək aləti kimi istifadə edilir. Masaüstü kompüterlər asanlıqla təmir olunur və modernləşdirilə bilər. Zərurət olduqda onun tərkibinə daxil olan ixtiyari qurğunu başqası ilə əvəz etmək mümkündür.

Portativ kompüterlər daşınma üçün rahatdır. Onlardan əvvəllər əksər vaxtlarını ezamiyyət və tədbirlərdə keçirən biznesmenlər, komersantlar, müəssisə və idarə rəhbərləri, jurnalistlər istifadə edirdilər. Son illərdə onların funksional imkanları xeyli artdığından və qiymətlərinin ucuzlaşması səbəbindən onlardan istifadə edənlərin əhatə dairəsi sürətlə genişlənməkdədir. Hal-hazırda –Notbook, Lap top və Sub Netbook<sup>1</sup> portativ kompüterləri istehsal olunur. Onlardan adətən İnternet-lə və sənədlərlə işləmək üçün, telefon rabitəsi vasitəsilə istənilən coğrafi nöqtədən telefon xəttinə qoşulmuş digər kompüterlə informasiya mübadiləsi etmək üçün istifadə edilir.

Əvvəllər cib kompüterləri (şəkl.1) “intellektual qeydiyyat kitabçası” kimi yaradılmışdır və bu səbəbdən Personal Digital Assistant (PDA) – “şəxsi rəqəmli katibə” adlandırılmışdır. Bu gün müasir cib kompüterləri geniş funksional imkanlara malikdir. Onlar istifadəçilərə aşağıdakı xidmətləri təklif edir:

- Elektron kitabların oxunmasını;
- İnternet xidməti;
- Elektron xəritələrlə iş. GPS<sup>2</sup> modulu və marşrutu planlaşdırma proqramları olduqda naviqasiya xidməti;

---

<sup>1</sup> Sub Netbook kompüterləri Notbooklardan nisbətən kiçik ölçüsünə və kütləsinə görə fərqlənir

<sup>2</sup> GPS-Global Mövqe Sistemidir.

- Orqanayzer;
- Qeyd və məlumat kitabçası;
- Diktafon;
- Əlyazmanın daxil edilməsi;
- Mətnlərin daxil edilməsi;
- Multimedia fayllarına-rəsmlərə, video çarxlara, filmlərə baxış, musiqi və səs fayllarına qulaq asmaq;
- Oyunlar;
- Qrafiki redaktorla iş;
- Distant idarəetmə<sup>1</sup>;
- Ofis proqramları ilə iş;
- Fotoaparət və videokamera;
- Telefon;
- SMS, MMS mübadiləsi.



Şək. 1.

Cib kompüterləri USB kontrolleri ilə təhciz olunmuşdur ki, onun vasitəsilə müxtəlif qurğular, o cümlədən fleş yaddaş, printer qoşmaq olar<sup>2</sup>. Cib kompüterlərində aşağıdakı əməliyyat sistemləri<sup>3</sup> istifadə olunur:

---

<sup>1</sup> Bütün infraqırmızı portlu məişət texnikası cib kompüterilə idarə oluna bilər.

<sup>2</sup> Kontroller, fleş yaddaş, printer haqda sonrakı paraqrafda məlumat veriləcək.

<sup>3</sup> Əməliyyat sistemləri hər bir kompüterin baza proqramıdır. Bax: Kompüterlərin proqram təminatı.

- Windows Mobile (əvvəllər Pocket PC və Windows CE versiyaları) ;
- PalmSource;
- Research In Motion;
- Symbian OS (əvvəllər EPOC versiyası);
- GNU/Linux;
- Mac OS X (əvvəllər iPhone OS versiyası).

Son zamanlar cib kompüterlərinin və mobil telefonların imkanlarını birləşdirən qurğular-smartfonlar (şək.2) və kommunikatorlar (şək.3) istifadə edilməkdədir.



Şək.2.



Şək.3.

**Uyuşqanlıqına görə təsnifat.** Dünyada çoxlu adda müxtəlif konfigurasiyalı minikomputerlər istehsal olunur. Onlar müxtəlif istehsalçılar tərəfindən satışa buraxılır, müxtəlif detallardan yığılır və müxtəlif proqramlarla işləyirlər. Bu halda çox vacib məsələ müxtəlif kompüterlərin öz aralarında uyuşqan olmasıdır. Uyuşqanlıqdan kompüterin qurğularının qarşılıqlı əvəzlənə bilməsi, bir kompüterdən digərinə proqramların

köçürülməsi imkanı və eyni bir verilənlərlə müxtəlif tip kompüterlərdə birgə işləmə imkanı asılıdır.

Aparat uyuşqanlığına görə kompüterlər, aparat platforması adlanan anlayışa görə fərqlənirlər. Fərdi kompüterlər sahəsində bu gün daha geniş iki aparat platforması yayılmışdır: IBM və Apple Macintosh. Bunlardan başqa, yayılması ayrı-ayrı regionlarla və ya sahələrlə məhdudlaşan digər platformalar da vardır. Kompüterlərin eyni bir aparat platformasına aidiyyəti, onlar arasında uyuşqanlığı artırır, müxtəlif platformalara məxsusolma isə uyuşqanlığı azaldır.

Aparat uyuşqanlığından başqa proqram səviyyəsində uyuşqanlıq mühümdür. Qeyd edək, IBM və Apple Macintosh kompüterləri fərqli proqram təminatından istifadə edir yəni, proqram səviyyəsində uyuşan deyil.

### 1.3 Masaüstü fərdi kompüterin tərkib hissələri və perferiya qurğuları

Müxtəlif forma və konfigurasiyaya malik olmasına baxmayaraq, masaüstü fərdi kompüterlər eyni arxitekturaya malik olub, əsasən sistem blokundan və ona qoşulan monitordan, klaviaturadan, siçandan və digər əlavə perferiya qurğulardan ibarətdir.

**Sistem bloku.** Sistem bloku üfiqi və ya şaquli formada olan və fərdi kompüterin daxili qurğularını özündə saxlayan “qutudur”. Sistem blokunda qidalanma bloku, ana lövhə,<sup>1</sup> montaj oyuqları, elektrik xəttinə qoşulma, yenidən yükləmə düymələri yerləşir. Montaj oyuqları disk sürüşdürücülərinin-diskovod, CD-ROM/DVD-ROM üçün nəzərdə tutulmuşdur.

Qidalanma bloku metal qutu şəklində sistem blokunun arxa tərəfinə bərkidilir. Qida blokunda transformator,

---

<sup>1</sup> Buna bəzən əsas plata (Mainboard) və ya sistem platası da deyirlər.

düzləndirici və ventilyator yerləşdirilmişdir. Transformator şəbəkədən sistem blokuna daxil olan 220 volt gərginliyi transformasiya edərək müxtəlif qiymətli, gərginliklərə<sup>1</sup> çevirir. Düzləndiricinin vəzifəsi fərdi kompüterin bəzi elementlərini sabit gərginlik ilə təmin etməkdir. Ventilyatordan qidalanma blokunun daxilində yerləşmiş transformatorun və düzləndiricinin iş prosesində qızmasının qarşısını almaq üçün istifadə edilir.

Qidalanma blokundan çıxan naqillər dəsti müxtəlif qurğuları gərginliklə təmin etmək üçündür və ana löhvəyə, disk sürüşdürücülərinə birləşdirilir.

Ana löhvə kompüterin ən əsas qurğusudur. Ana plata kompüterin daxili və xarici qurğularının əlaqəli idarəsini təmin edir. Ana löhvənin əsas xarakteristikası formafaktordur. Hər bir formafaktor ölçüsünə, özündə saxlayan elementlərin sayına, elementlərin yerləşmə konfigurasiyasına, istifadə etdiyi gərginliyə və gərginliyə qoşulma formasına və s. parametrlərə görə fərqlənir.

Müasir kompüterlərdə aşağıdakı tip formafaktorlu sistem platalar istifadə olunur:

- **ATX;**
- **Flex-ATX;**
- **Mini-ATX;**
- **NLX;**
- **Mini-ITX;**
- **Monobook.**

Ana löhvə daxildən sistem blokunun korpusuna bərkidilir. Bunun üçün onun formafaktoru korpusa uyğun gəlməlidir.

Ana löhvə öz üzərində aşağıdakı bir çox komponentləri birləşdirir:

---

<sup>1</sup>Məsələn, xarici yaddaş qurğuları 5 və 12 volt, mərkəzi prosessor və bir sıra mikrosxemlər 1,0-1,5 volt gərginlik tələb edir.



• Xüsusi yuvaya birləşdirilmiş mərkəzi prosessor-mikroprosessor. Adətən onu soyutmaq üçün ventilyatordan-cooler istifadə olunur;

- Məntiqi sistem mikrosxemləri-çipsetlər;
- İkinci səviyyəli (xarici) keş-yaddaş mikrosxemi;
- Operativ yaddaş<sup>1</sup> yerləşdirmək üçün yuvalar<sup>2</sup> ;
- Kart genişlənmələri (videoadapter, səs kartı, modem və s.) üçün yuvalar;

- BIOS Kompüterin test yoxlamasını, əməliyyat sisteminin yüklənməsini, qurğuların drayverini və s. yaddaşda saxlamaq üçün proqramlaşdırılmış yaddaşa malik mikrosxem;

- Interfeys kabellərinin, ardıcıl və paralel portların qoşulması üçün yuvalar;

- BIOS-un cari sazlanmasını yadda saxlamaqdan ötrü CMOS tipli mikrosxem və onu cərəyanla təmin etmək üçün batareya və elektron saat (sistem saati).

Ana platanın bütün komponentləri bir-birilə informasiya mübadiləsini aparan keçirici xətlərlə birləşdirilir. Bu cür keçirici xətlər birlikdə informasiya şini və ya sadəcə, şin adlanır.

**Mikroprosessorlar.** Mikroprosessor – kompüterin “beyni” hesab edilir və kompüterdə verilənlərin emalını, ötürülməsini və xarici qurğuların idarə edilməsini saniyənin milyonda bir hissəsində təmin edir. Mikroprosessorun əsas xarakteristikası onun işçi gərginliyi, takt tezliyi, takt tezliyinin<sup>3</sup> daxili çoxaltma əmsalı, keş-yaddaşının tutumu və dərəcəliliyidir. Mikroprosessoru işçi gərginlik ilə ana palata təmin edir. Intel firmasının əvəllər istehsal etdiyi

---

<sup>1</sup>Operativ yaddaş. Kompüter şəbəkəyə qoşulan zaman verilənləri müvəqqəti saxlamaq üçün istifadə olunan mikrosxem toplusudur.

<sup>2</sup> SIMM tipli operativ yaddaş üçün 72 oyuqlu, DIMM tipli operativ yaddaş üçün 168 oyuqlu yuva istifadə olunur.

<sup>3</sup> Takt tezliyini bəzən kompüterin və mikroprosessorun sürəti də adlandırırlar.

mikroprosessorlarda gərginlik 5.0 volta bərabər idisə, indiki zamanda firma tərəfindən buraxılan mikroprosessorlarda bu qiymət 3.0 volta çatdırılmışdır. İşçi gərginliyinin qiymətinin aşağı həddə endirilməsi mikroprosessorun daha da məhsuldar işləməsini təmin edir.

Takt tezliyi bir saniyə ərzində kompüterdə yerinə yetirilən əməliyyatların (məsələn, toplama və vurma) sayını və həmin əməliyyatların hansı sürətlə yerinə yetirildiyini göstərir. Takt tezliyi meqaherslərlə (Mhs) ölçülür. Takt tezliyi artdıqca mikroprosessorun qiyməti ilə yanaşı onun məhsuldarlığı da artır və mikroprosessor çoxlu sayda əməlləri yerinə yetirə bilər. Takt siqnallarını mikroprosessor ana lövhədən qəbul edir. Mikroprosessoru daha yüksək takt tezliyi ilə təmin etmək üçün onun daxilindəki takt tezliyinin daxili çoxaltma əmsalından istifadə olunur (Məsələn, 3.0, 3.5, 4.0, 4.5, 5.0 və daha artıq).

Takt tezliyinin daxili və xarici növü vardır. Daxili takt tezliyi mikroprosessorun yerinə yetirdiyi əməliyyatların tezliyini, xarici takt tezliyi isə kompüterin operativ yaddaşı ilə mikroprosessor arasında informasiyanın dəyişmə tezliyini göstərir.

Prosessorun daxilində reqistr adlanan müəyyən sahə var ki, mikroprosessor emal etdiyi verilənləri orada saxlayır. Prosessorun dərəcəliliyi mikroprosessorun daxilində yerləşən reqistrlərin dərəcəliliyi ilə təyin olunur və bir takt ərzində prosessorun neçə bit<sup>1</sup> informasiya mübadilə etdiyini xarakterizə edir. Qeyd edək ki, ilk istehsal olunan mikroprosessorlarda dərəcəlikli 8 idi. Müasir dövrdə istehsal olunan mikroprosessorlarda bu parametr 64-ə çatdırılmışdır.

**Çipsetlər.** Çipsetlər kompüterin sistem komponentlərinin qarşılıqlı əlaqəsini təmin edir. Ana lövhə üzərində şimal və cənub körpü adlanan iki çipset yerləşir.

---

<sup>1</sup> Bit informasiyanın ən kiçik ölçü vahididir.

**Operativ yaddaş.** Kompüterin digər əsas elementi onun operativ yaddaşdır<sup>1</sup>. Mikroprosessorlar əməliyyat əmrlərini və ilk verilənləri operativ yaddaş vasitəsilə əldə edir. Operativ yaddaşın mənfəət cəhəti kompüter elektrik şəbəkəsindən ayrıldıqda onda olan informasiyanı özündə saxlaya bilməməsidir. Texniki ədəbiyyatlarda operativ yaddaş çox vaxt RAM (Random Access Memory) adlandırılır.

Operativ yaddaş digər mikrosxemlərdən fərqli olaraq tranzistorlardan deyil, hər biri tutumu bir bit olan mikrokonkondensatorlardan yaradılıb.<sup>2</sup>

Operativ yaddaşın tutumu kompüterin işləmə sürətinə təsir edir. Operativ yaddaşın tutumu kifayət qədər olmayanda istənilən informasiyanın mikroprosessor tərəfindən axtarılması üçün mikroprosessor artıq əməliyyatları yerinə yetirməyə məcbur olur. Nəticədə proqramın yerinə yetirilmə vaxtı uzanır. Bu səbəbdən operativ yaddaşın tutumu qənaətbəxş olmayanda bəzi proqramlar kompüterdə ya işləməyəcək, ya da ki, həddindən artıq ağır işləyəcəkdir. Məsələn, kompüterlərin Windows XP əməliyyat sistemi ilə işləməsi üçün operativ yaddaşın tutumu minimum 128 Mbayt-a bərabər olmalıdır.

Fərdi kompüterlərdə operativ yaddaşın müxtəlif növlərindən istifadə edilir:

*Adi yaddaş.* Ona bəzən DRAM da deyirlər. Yaddaş istənilən kompüterdə qurula bilər. Onun əsas xüsusiyyəti yaddaşa müraciət etmə vaxtıdır. Bu parametrlər 60 n.saniyəyə bərabərdir. Qeyd etmək lazımdır ki, 70, 80, 90 və 100 n.saniyəlik yaddaşlar da istehsal olunurdu. Amma qeyri-müəyyən səbəblər üzündən onların istehsalı dayandırıldı.

*EDRAM yaddaş.* Ondan istifadə etdikdə DRAM tipli yaddaşla müqayisədə kompüterin işləmə sürəti orta hesabla 2%

---

<sup>1</sup> Yaddaş həddindən artıq cəld işlədiyinə görə operativ adını almışdır.

<sup>2</sup> Məsələn, 256 mбайt tutumu olan operativ yaddaş 2,147,483,648 sayda mikrokonkondensatordan təşkil olunmuşdur.

artmış olur. Belə yaddaşlara müraciət vaxtı 70 n.saniyə bərabər hesab edilir.

*SDRAM yaddaş.* SDRAM yaddaşa müraciət vaxtı çox kiçikdir, təxminən 10-12 n. san-yə bərabərdir. Yaddaşdan istifadə etməklə kompüterin işləmə sürətini 10% artırmaq mümkündür.

**Keş yaddaş** (ingiliscə cache). Operativ yaddaşa müraciəti sürətləndirmək üçün kompüterlərdə xüsusi hazırlanmış yaddaşdan - keş yaddaşdan istifadə edilir. Keş yaddaşı mikroprosessorla əsas yaddaş arasında yerləşən kiçik tutuma və yüksək işləmə sürətinə malik yaddaşdır. Ondan kompüterin məhsuldarlığını artırmaq üçün istifadə edilir. Bütün əsas yaddaşın sürətlə işləyən keş yaddaş kimi hazırlanması texnoloji cəhətdən çox baha başa gəlirdi. Odur ki, iqtisadi cəhətdən kiçik tutuma malik yaddaş sahəsinin sürətinin artırılması əlverişlidir.

Kompüterin yaddaşına müraciət zamanı verilənlər keş yaddaşda axtarılır. Buna əsas səbəb odur ki, keş yaddaşına verilənləri axtarmaq üçün edilən müraciət vaxtı operativ yaddaşa edilən müraciət vaxtından bir neçə dəfə azdır. Kompüterlərdə keş yaddaşın iki növündən istifadə edilir: daxili və xarici.

Xarici keş yaddaşı müxtəlif mikrosxemlər üzərində yığılır və operativ yaddaşın işini sürətləndirməyə xidmət edir.

Daxili keş yaddaş isə mikroprosessorun daxilində yerləşərək mikroprosessorun daxilindəki reqistrlərin işini sürətləndirir. Belə keş yaddaşlardan ilk dəfə 80486 mikroprosessorlarında istifadə edildi. Onların tutumu 8 Mbayt-a bərabər idi.

Intel firmasının istehsalı olan Pentium, Pentium Pro və sonrakı model mikroprosessorlarında keş yaddaş birbaşa mikroprosessorun daxilində yerləşdirilir. Bu çox əlverişlidir. Çünki keş yaddaş üçün əlavə lövhə hazırlamaq lazım gəlmir.

**BIOS (basic input-output system).** Ana lövhə üzərində enerjidən asılı olmayan daimi yaddaş qurğusu BIOS yerləşir. Bu mikrosxemdə fərdi kompüterin qurğularının testləşdirən (məlumatı ekrana çıxarmaqla), əməliyyat sisteminin yüklənməsini təmin edən proqramlar toplusu yerləşir. Digər yaddaş qurğularından fərqli olaraq onun tutumu elə də böyük olmur. Qeyd edək ki, BIOS Setup proqramı vasitəsilə BIOS-un konfigurasiyasını dəyişmək olar.

**Portlar.** Mikroprosessorun xarici qurğular (printer, sican və sairə) ilə informasiya mübadiləsini həyata keçirən yuvalara portlar deyilir. Əsas portlar sistem blokunun arxa panelindən görünür. Paralel – LPT (paralel printer portu), ardıcıl -COM (universal ardıcıl port) və USB (Universal Serial Bus-universal ardıcıl şin) portlardan istifadə olunur. Ardıcıl portlardan siçan və klaviaturanın qoşulması üçün istifadə olunur. Paralel portlar printerlərin qoşulması üçün istifadə olunur. Ardıcıl portlar sürətli məlumatlar mübadiləsi ilə fərqlənilirlər. Qeyd edək ki, hal-hazırda əksər müasir kompüterlər paralel və ardıcıl portlarsız istehsal olunur. Son dövrlərdə meydana gələn USB portları asanlıqla onları əvəz edir. USB portları flaş disklərin, klaviatura, siçan, printer, mobil bərk disklərin, skanerin və bir sıra periferiya qurğularının qoşulmasını təmin edir. USB portları daha sürətli olub, informasiya mübadiləsini 480 Mbayt/san sürətlə təmin edir.

**Kontroller və şin.** Kompüterin işləməsi üçün operativ yaddaşda proqramlar və verilənlər olmalıdır. Belə proqramlar və verilənlər operativ yaddaşa müxtəlif qurğular-klaviatura, diskovod və s. vasitələrlə daxil olurlar. Kompüterdə emal edilmiş informasiya isə istifadəçiyə monitor, disklər, printer və s. vasitəsilə çatdırılır. Başqa sözlə, kompüterdə qurğular və operativ yaddaş arasında daima informasiya mübadiləsi prosesi gedir. Bunun üçün, hər xarici qurğu özünə məxsus elektron

sxem ilə təchiz edilir ki, belə sxemlərə də kontroller qurğusu deyirlər.

**Kompüter şini və ya sistem şini** – prosessorla yaddaş arasında və ya giriş-çıxış qurğularının kontrollerləri (idarə sxemləri) arasında verilənlərin və idarə siqnallarının ötürülməsini təmin edən naqillər yığımından ibarətdir. Kompüterin bütün hissələri bir-biri ilə sistem şini vasitəsilə birləşmişdir.

Bütün kontrollerlər kompüterin daxilində yerləşdirilmiş mikroprosessorlar ilə magistral sistemi vasitəsilə əlaqə yaradır ki, buna da verilənlər şini deyirlər. Verilənlər şini prosessorla yaddaş arasında və ya giriş-çıxış qurğularının kontrollerləri (idarə sxemləri) arasında verilənlərin və idarə siqnallarının ötürülməsini təmin edən naqillər yığımından ibarətdir.

İnformasiya mikroprosessorla verilənlər şini vasitəsilə ötürülür. Əgər verilənlər şini 8 siqnal ötürən naqıldən ibarət olarsa, onda paralel olaraq 8 bit, 16 naqıldən ibarət olarsa, 16 bit informasiya göndərmək olar və s.

Prosessorun daxilində və fərdi kompüterin ana lövhəsində verilənlər şini və ünvan şini mövcuddur. Verilənlər şini ötürücü və yardımçı elementlər sistemi olub, informasiyanın mikroprosessorla verilməsi və ondan alınması prosesini həyata keçirir. Verilənlər şini müxtəlif dərəcəliliyə malik olur. Şinin dərəcəliliyinin informasiyanın ötürülmə sürətinə təsir etməsi nəticəsində kompüterin işləmə sürəti də dəyişir.

Ünvan şini də ötürücü və yardımçı elementlər sistemi olub, fərdi kompüterin yaddaşında saxlanılacaq və yazılacaq informasiyanın tutduğu sahənin yerini təyin edir. Ünvan şini də dərəcəliliyə malikdir. Ünvan şininin dərəcəliliyi mikroprosessorun müraciət etdiyi yaddaşın tutumuna təsir edir.

İstehsal olunan fərdi kompüterlərdə iki, bəzi hallarda isə üç verilənlər şinindən istifadə olunur. Kompüterdə aşağıdakı şinlər istifadə olunur:

- IBM tipli fərdi kompüterlər üçün xüsusi olaraq yaradılmış ISA (Industry Standard Architecture) şini. Şinin müsbət cəhəti sadə və çox ucuz qiymətə, mənfi cəhəti isə informasiya ötürməsinin aşağı sürətdə olmasıdır.

- Yüksək sürətli qlobal EISA (Extended Industry Standard Architecture) şini. Şinin müsbət cəhəti yüksək sürətlə buraxma qabiliyyəti, mənfi cəhəti isə qiymətinin yüksək olmasıdır.

- Yüksək sürətli və ucuz qiymətli PCI (Peripheral Component Interconnect bus) lokal şini.

Son zamanlar xüsusi olaraq qrafik materialların ekrana çıxarılmasını sürətləndirmək və mərkəzi mikroprosessoru iş prosesində yükəndən azad etmək üçün PCI ilə müştərək funksiya yerinə yetirən AGP lokal şini də istifadəyə buraxılmışdır.

Fərdi kompüterə əlavə qurğuların (sican, klaviatura, rəqəmli kamera və s.) qoşulmasını sadələşdirmək məqsədilə istifadəçilər USB lokal şinindən də geniş istifadə edirlər.

**Xarici yaddaş qurğusu.** Xarici yaddaş qurğusu kimi sərt maqnit disklərindən (Hard Disk Drive (HDD)) və ya vinçestrədən<sup>1</sup> istifadə edirlər. Sərt disklər operativ yaddaşı nəzərə almasaq, yaddaş qurğuları içərisində, verilənlərə ən tez girişi (adətən 7-20 milli saniyə) və onların ən sürətli (5 Mbayt/saniyə) oxunmasını və yazılmasını təmin edən qurğulardır. Belə disklərə informasiya xüsusi texnologiya ilə yazılır. Sərt maqnit disk qurğusu 2-6 maqnit diskindən ibarət olub, möhkəm korpusda saxlanılır. Onun idarə olunması eyniadlı kontroller vasitəsilə həyata keçirilir. Sərt maqnit disklərində əməliyyatlar sistemləri, tətbiqi proqramları, müxtəlif verilənlər də saxlayırlar. Sərt maqnit diskləri kompüter şəbəkədən ayrıldıqda belə proqram və verilənləri uzunmüddətli saxlayır.

---

<sup>1</sup> Diskin adı "Vincester 30/30" markalı tufəngin adından götürülüb. Belə ki, haqqında danışılacaq ilk maqnit disklərin də markası 30/30 idi.

*Sərt maqnit diskləri ana lövhəyə qoşulma üsuluna görə aşağıdakı siniflərə bölünür:*

1. IDE interfeysli (və ya E-IDE interfeysli);
2. SCSI interfeysli (Ultra SCSI, Ultra II SCSI, Ultra Wide SCSI və s. interfeysi ilə).

İkincilər birincilərə nisbətən qiymətə baha olmaqla yanaşı, onların işləmə sürətləri də çoxdur.

Sərt maqnit disklər ilk dəfə kompüterə qoşulduqda formatlaşdırılmalıdır. Bu zaman diskdə və verilənlərin saxlanması üçün işçi sahə yaradılır. Sərt maqnit diskləri bir-birindən aşağıdakı xüsusiyyətlərinə görə fərqlənilir:

- tutumu, yəni informasiyanın diskə hansı həcmdə yerləşməsinə görə;
- tezliyi, yəni informasiyanın diskə hansı sürətlə yazılması və oxunmasına görə;
- interfeys, yəni sərt diskin qoşula biləcəyi kontrollerlərin tipinə görə.

İnformasiyanın diskdən oxunma/yazılma sürəti kompüterdə istifadə edilən mikroprosessorun tezliyindən, kontrollerlərin və şinlərin tipindən, həmçinin istifadə edilən sərt maqnit diskinin özündən çox asılıdır. Müasir texnologiya ilə hazırlanmış sərt maqnit disklərində bu parametr 1.3-5 Mbayt/s arasında dəyişir.

Sərt maqnit disklərinin kompüterdə istifadə olunması kompüterdən istifadə etməyi həddindən artıq rahat edir. İndiki zamanda sərt maqnit diski olmayan kompüterlərdən demək olar ki, istifadə edilmir.

**Monitor.** Fərdi kompüterə daxil edilən və oradan oxunan informasiyanı əks etdirmək üçün monitordan (displaydən-“display” əksətdirmə deməkdir) istifadə edilir. Monitorların idarə olunması fərdi kompüterlərin daxilində formalaşan rəqəmli videosiqnallarının vasitəsilə baş verir. Qeyd edək ki, monitorda lazımi informasiyanı əks etdirmək üçün kompüterə daxil olan məlumatları videosiqnal şəklində



formatlaşdırmaq lazımdır. Bunun üçün istifadə edilən elektron sxemlərini videokontrollerlər adlandırırlar. Videokontrollerlər əsasən xüsusi elektron lövhə formasında hazırlanır. Əksər hallarda isə kompüterin daxilində yerləşən ana lövhənin tərkibinə daxil olurlar. IBM tipli kompüterlərdə istifadə olunan videokontrollerlər iki-mətn və qrafiki rejimdə işləyirlər.

Elektron şüa borulu ( CRT – Cathode Ray Tube ), maye kristallı (Liquid Crystal Display) monitorlardan istifadə olunur. CRT monitorların əsas elementi elektron şüa borusudur. Borunun içərisi lyuminofor adlanan materialla örtülüb və içərisində elektron dəstələri vardır. Lyuminofor qatı triadalarından təşkil olunub. Triadalar 3 nöqtədən ibarət elementlərdir. Bu üç nöqtə qırmızı (Red) , yaşıl (Green) və mavi (Blue) rəngi əks edən elementlərdir. Borunun dar ucunda elektron topu (elektronlar mənbəyi) yerləşir. Bunlar şüalar buraxır. Həmin şüalar triadaların üzərinə lazımi intensivlikdə düşərək lazımi rəngi verir. Məsələn, ağ rəngi vermək üçün RGB (255,255,255) olmalıdır, yəni qırmızı, yaşıl və mavi elementlər maksimum intensivliyə malik olur və ağ rəngi verir. Qara rəng üçün RGB(0,0,0)-dir. Beləliklə, hər pikselə bir triada uyğun gəlir və pikselin rəngi triadanın elementlərinə düşən intensivlikdən asılıdır.

Funksional təyinatına görə CRT monitorları hərf-rəqəm və qrafik rejimdə işləyən monitorlara ayrılırlar. Hərf-rəqəm monitorları hərf-rəqəm tipli informasiyanı əks etdirmək üçündür. Belə monitorun ekranında hər birinin uzunluğu 80 simvol olan 24-25 sətir yerləşdirmək mümkündür. Başqa sözlə, hərf-rəqəm monitorlarının ekranında təxminən 2000 simvolu təsvir etmək olur. İndiki zamanda mətn rejimində işləyən monitorlardan demək olar ki, istifadə olunmur.

Monitorun əsas göstəriciləri ekranın ölçüsü, ekranda təsvir olunan nöqtələrin sayı, təsvirin əks olunmasının maksimal tezliyi və ekranın müdafiə dərəcəsidir.

Monitorlarda təsvirin yenidən əks olunması bir saniyə ərzində təsvirin tamamilə ekranda təzələnməsi ilə səciyyələndirilir. Əksər hallarda bu parametri kadrların tezliyi də adlandırırlar. Parametr monitorun və videoadapterin xüsusiyyətlərindən asılı olaraq dəyişə bilər.

Əks olunma tezliyi herslərlə ölçülür. Tezlik artdıqca ekranda təsvir bir o qədər də təmiz alınır. Müasir monitorlarda tezlik 75-100 hers arasında dəyişir.

Ekranın müdafiə dərəcəsi ümumdünya standartlarına (məsələn, MPR-II, TCO, TCO-95 və s.) uyğun təyin edilir.

İstifadə olunma sahələrinə görə monitorlar müxtəlif tənzimləmə qurğular ilə təmin edilirlər.

Monitorlarda əsas göstəricilərdən biri ekranda təsvir olunan nöqtələrin sayıdır. Müxtəlif monitorlarda nöqtələrin sayı müxtəlif olur. Məsələn, adi monitorlarda üfiqi və şaquli istiqamət üzrə nöqtələrin sayı  $640 \times 480 = 307200$ -ə qədər, professional monitorlarda isə  $1600 \times 1280 = 2048000$ -ə qədər olur. Rəqəmlərin sayından görünür ki, monitorlarda nöqtələrin sayı milyondan çoxdur. Aydın məsələdir ki, monitorlarda istənilən təsvirin dəqiqliyini artırmaq üçün nöqtələrin sayını tənzimləmək lazımdır. Həmçinin ekranda əks olunan təsvirlərin rənglərinin sayını da tənzimləmək olar, bu rənglər 16-16,8 milyon intervalında ola bilər.

Müasir dövrdə daha çox maye kristal monitorlardan istifadə olunur. Bu tipli monitorların ekranları adi elektron saatlar, mobil telefonlar və s. olan ekranlarla eynidir. Bu ekran iki təbəqədən ibarətdir. Həmin təbəqələrin arasında milşəkili molekullar yəni maye kristallı qarışıqlar vardır. Kristallar işığı polarizatorlardan keçirirlər. Elektrik cərəyanı mayedən keçəndə kristalları bir istiqamətdə yönəldir. Kristallar işığı polarlaşdıran müstəvi qatlarını döndərir. Nəticədə işıq buradan keçə bilmir və həmin hissə tünd rəngə boyanır.

LCD ekranlarda CRT monitordan fərqli olaraq cəmi 64 rəng çaları yaratmaq mümkündür<sup>1</sup>. Hər pikseldə 3 element qırmızı, yaşıl, mavi (RGB) olmaqla hər element 64 rəng çaları yarada bilər. Bununla da  $64*64*64=262\ 144$  sayda rənglərdən alınan kombinasiyadan birini almaq olar. CRT ekranlarda isə bu kombinasiya  $256*256*256=16\ 777\ 216$  sayda olur. LCD ekranlarda “təsviri” insan gözünün rahat görməsi üçün parlaqlığı artırmaq üçün elementlər (backlight) yerləşdirirlər.

**Videoadapter.** Monitorun ekranında alınmış təsviri formalaşdırmaq üçün videoadapterdən (videokartdan) istifadə olunur. Videoadapter mətn və ya qrafik rejimdə işləyə bilər.

*Videoadapterin iki növündən istifadə edilir:*

- 16 rəngli 80x25 və ya 80x50 simvolla mətn rejimində və 600x350 və ya 640x480 nöqtələrə qrafik rejimdə, həmçinin 256 rənglə 320x200 nöqtələrlə qrafik rejimdə ekranı təmin edən VGA videokartı;

- 16 milyon rənglə 640x480 simvollarla müxtəlif rejimlərdə ekranı təmin edən SGVA və ya super VGA videokartı.

Qeyd etmək lazımdır ki, videoadapterin yaddaşını artırmaqla (məsələn, 1 Mbayt-dan 2 Mbayta qədər) rənglərin və simvolların rejimlərə uyğun ekranı təmin etməsini çoxaltmaq olar.

Videoadapter sətirlərarası (Interlaced) və ya sətir-sətir (Noninterlaced) genəltmə rejimlərində işləyə bilər. Sətirlərarası rejimdə monitorun ekranında alınmış təsvir iki kadr ilə formalaşır. İkinci rejimdə isə təsvir bir kadr vasitəsilə formalaşmış olur. Birinciyə nəzərən monitorun ekranında alınmış təsvir daha aydın və dəqiq olur.

Videokartı səciyyəyləndirən digər parametrlər ekranda təsvirin dəyişmə tezliyidir. Bu parametrlər 50-120 hers tezlik diapazonunda dəyişir. Təcrübə göstərir ki, fərdi kompüterlə

---

<sup>1</sup> 256 rəng çaları yaratmaq mümkündür.

işləyən istifadəçinin normal işləməsini təmin etmək üçün, həmçinin onun iş prosesində gözünün yorulmamasını əldə etmək üçün videoadapteri sətir-sətir genəltmə rejimində işlətmək məqsəduyğundur. Bu zaman kadrların dəyişmə tezliyi 75 hersdən az olmamalıdır. Ümumilikdə isə, 100 Hs tezlik optimal sayılır. Bəzi hallarda videoyaddaşın lazımı səviyyədə tezliyə malik olmaması nəticəsində monitorun ekranında göstərilən təsvir dumanlı şəkildə alınmış olur. Bu isə videokartın lazımı səviyyədə işləməməsinə dəlalət edir.

Videoadapterin müasir proqramlar paketi ilə işləməsi üçün o, mütləq sürətləndirmə funksiyasına malik olmalıdır. Başqa sözlə desək, videoakselerator rolunu ifa etməlidir. Videoadapterin belə rejimdə işləməsi mərkəzi mikroprosessorun yükünü azaldır, nəticədə kompüterin işləmə sürəti çoxalır.

Sürətləndirmə funksiyası ikiölçülü (2D) və üçölçülü (3D) sinfə bölünür. İkiölçülü sürətləndirmə rejimi bütün müasir proqramların (ofis paketlərindən tutmuş kompüter oyunlarına qədər) icra olunma sürətini artırmağa imkan verir. Üçölçülü sürətləndirmə rejimində isə kompüter oyunları ilə yanaşı, kompüterdə istifadə edilən stimulyasiya və modelləşdirmə proqramlarının həyata keçirilməsinə şərait yaranır.

İndiki zamanda istehsal olunan müasir videoadapterlər ikiölçülü sürətləndirmə funksiyasına malikdirlər.

**Səs kartı (səs adapteri).** Səsləri<sup>1</sup> (musiqi, danışiq və s.) səsləndirmək üçün kompüterə akustik sistemlər (səs ucaldanlar) və səs kartları quraşdırılır.

Müasir kompüterlərdə dərəcəsiindən asılı olaraq (8 və ya 16 bitlik) səs kartları istifadə olunur. İstifadəçi çalışmalıdır ki, istifadə etdiyi səs kartları Sound Blaster 8 (8 bitlik səs kartı

---

<sup>1</sup> Səs dedikdə, insan qulağının 16 Hs-dən 25000 Hs-ə kimi hava titrəyişini qəbul etməsi başa düşülür. Səs müxtəlif tezliklərdə olub, amplitudası və fazası ilə səciyyələndirilir.

üçün standart) və ya Sound Blaster 16 (16 bitlik səs kartları üçün standart) səs kartları ilə uzlaşmış olsun. Onda istifadəçi iş prosesində meydana gələcək bütün problemlərdən yan keçmiş olar.

Qeyd edək ki, kompüter CD/DVD sürüşdürücüləri və səs kartı ilə təmin olunmuşsa, belə fərdi kompüterləri multimediyalı kompüterlər adlandırırlar. Onlardan təhsildə, istirahətdə və əyləncədə istifadə edirlər.

**Klaviatura.** Fərdi kompüter ilə istifadəçi arasında əlaqə yaradan qurğu klaviaturadır. Klaviaturanın istənilən düyməsi mexaniki və ya membran tipli miniatur çeviricidən ibarətdir. Çeviricinin vəziyyətinə nəzarəti klaviaturanın daxilində yerləşmiş mikrosxem yerinə yetirir. Mikrosxemin vəzifəsi klaviatura üzərində olan istənilən düymənin basılmasına uyğun məlumatı mərkəzi mikroprosessorla göndərməkdir. Bu məlumat mikroprosessorun daxilində olan əmrlər sisteminin köməyi ilə təhlil edilir.

Adətən, klaviatura üzərində müxtəlif sayda düymələr yerləşdirilir. Məsələn, IBM PC XT markalı köhnə tipli kompüterlərdə 84/86 düyməli klaviaturadan (müasir dövrdə belə klaviaturadan istifadə edilmir), IBM PC AT markalı müasir tipli kompüterlərdə isə 102 və ya 104 düyməli klaviaturadan istifadə edilir.

Klaviaturanın müxtəlif modifikasiyalarının hazırlanmasına baxmayaraq, onlar üzərində yerləşdirilən düymələrin funksiyaları demək olar ki, dəyişməz qalır. Düymələrin funksiyalarını dəyişməsi yalnız və yalnız kompüterin istifadə olunma sahəsinin kompüterin daxilində olan proqramlar dəstinin dəyişməsi ilə baş verə bilər.

Ümumiyyətlə, klaviaturanın düymələrini 4 qrupa bölmək olar. Birinci qrupa daxil olan düymələr klaviaturanın mərkəzində yerləşir. Onlar hərf, rəqəm və bəzi başqa simvolları kompüterə daxil etmək üçündür. Məsələn, Shift düyməsini sıxıb saxlamaqla əlifbanın baş hərflərini, həmçinin

klaviaturanın birinci qrupuna daxil olan yuxarı reqlitr simvollarını kompüterə daxil etmək mümkündür. Əgər istifadəçi müəyyən müddət ərzində yalnız baş hərflərlə işləyəcəksə, bu zaman Caps Lock düyməsini sıxmaq kifayətdir. Caps Lock düyməsini sıxdıqda istifadəçi yenidən kompüterdə əlifbanın kiçik hərfləri ilə işləyə bilər. Bəzi tip klaviaturaların üzərində Caps Lock rejimini qeyd edən işıqlı indikator da quraşdırılır (bəzi modellərdə Caps Lock-dan latın əlifbasından digər əlifba kodlaşmasına keçid düyməsi kimi istifadə edilir. Caps Lock baş hərflərin qeyd olunması deməkdir).

İkinci qrup düymələr funksional düymələr adlanır və onlar klaviaturanın yuxarı hissəsində yerləşdirilmişdir. Üzərində F1-F12 yazıları həkk olunmuş düymələrin təyinatı xüsusi proqramlarla müəyyən edilir. Bir çox hallarda istifadə edilən sistem proqramlarından asılı olaraq funksional düymələr Ctrl, Alt və Shift düymələri ilə birgə sıxılaraq öz funksiyalarını dəyişirlər.

Üçüncü qrup düymələr kursurun hərəkətini istiqamətləndirmək və ədədi simvollarla işləmək üçündür. Bu qrup düymələr əsasən iki rejimdə istifadə edilir. Birinci rejim rəqəmləri aktivləşdirmə rejimi adlanır (Num Lock rejimi) və bu rejimdə 9 rəqəmdən əlavə dörd hesab əməlinin işarələri kompüterə daxil edilir. İkinci rejimdən düymələr qrupundan kursoru idarə etmək üçün istifadə olunur. Yəni, düymələri basmaqla həm kursoru idarə edən düymələrin, həm də Insert və Delete düymələrinin yerinə yetirdiyi funksiyalar təkrar edilir. Hər iki rejim Num Lock düyməsini sıxmaqla müəyyən olunur.

Klaviaturanın üzərində yerləşən ←, ↑, →, ↓, Home, End, PgUp və PgDn düymələri kursurun hərəkətini idarə edən düymələr adlanır. Adları çəkilən düymələrdən birini sıxmaqla kursoru istənilən istiqamətdə hərəkət etdirməklə yanaşı, monitorun ekranındakı məlumatı “vərəqləmək” mümkündür. İstifadə məqsədindən asılı olaraq düymələr (həmçinin onların

Ctrl, Alt və Shift düymələri ilə kombinasiyaları) bəzən digər funksiyaları da yerinə yetirə bilirlər.

Qeyd etmək lazımdır ki, 102 sayılı düymələri olan klaviaturada kursurun hərəkətini idarə edən düymələrlə bir blok üzərində Insert və Delete düymələri də yerləşdirilmişdir.

İstifadəçi kompüterdə istənilən mətni “redaktə etmək” rejimində işləyirsə, adları çəkilən düymələrdən istifadə etməklə, monitorun ekranındakı mətni vərəqləyə bilər. Bu məqsəd üçün istifadəçi PgUp (Page Up)-“səhifəni yuxarı” və PgDn (Page Down)-“səhifəni aşağı” istiqamətləndirici düymələrdən istifadə edə bilər. Əgər kursoru sətirin əvvəlinə və ya sonuna gətirmək tələb edilirsə, o zaman istifadəçi uyğun olaraq Home və ya End düymələrini basa bilər.

Erqonomik tələbatı nəzərdə tutan xüsusi formalı klaviaturalar - *erqonomik klaviaturalar* istifadə olunur. Onlardan böyük həcmli informasiyaları daxil etmək üçün nəzərdə tutulan iş yerlərində, məqsədyönlü şəkildə istifadə edirlər. Erqonomik klaviaturalar təkcə mətn daxil edilməsinin məhsuldarlığını artırmır və iş günü ərzində ümumi diqqəti və bir sıra xəstəliklərin (məsələn, bel sütununun yuxarı hissəsinin osteoxondrozunu) ehtimalını, inkişaf dərəcəsini azaldır.

Sistem bloka qoşulma metodlarına görə klaviaturalar *simli* və *simsiz* klaviaturalara ayrılırlar. Simsiz klaviaturada informasiya ötürülməsi infraqırmızı şüalarla həyata keçirilir. Belə klaviaturaların adətən fəaliyyət radiusu bir neçə metrədən ibarət olur. Sinyalın mənbəyi isə klaviatura olur.

**Disk sürüşdürücüləri.** Disk sürüşdürücüləri disketlərin, CD və DVD disklərin oxunmasını və informasiyanın onlara yazılmasını təmin edir.

**Siçan.** Siçan plastik qurğu olub, kompüterlə istifadəçi arasında əlaqə yaratmaqla proqramların idarə olunmasında, mətnlər, şəkillər üzərində çoxsaylı əməllərin yerinə yetirilməsində böyük rol oynayır. Siçan qurğuları diyircəkli, optik formalarda istehsal olunur.

Siçanın sol düyməsi əsas əməlləri yerinə yetirir, sağ düyməsi isə seçilmiş obyekt üzərində əməliyyatlar aparmaq üçün kontekst menyu açır. Siçanın göstəricisi ekranda kursordur. Siçanlar qoşulma tipinə görə də fərqlənilirlər. Belə ki, Com, PS/2 və USB interfeysli siçanlar istehsal olunur.

**Xüsusi manipulyatorlar.** Adi siçan qurğusundan başqa, digər manipulyator tipləri də vardır. Məsələn: *trekbollar, penmauslar, infraqırmızı siçanlar*.

*Trekbol* siçan qurğusundan fərqli olaraq stasionar qaydada quraşdırılır və onun kürəciyi əlin ovucu ilə qaydaya salınır (nizamlanır). *Trekbolun* üstünlüyü ondan ibarətdir ki, onun hamar işçi müstəviyə ehtiyacı olmur, ona görə də *trekbollar* portativ (yığcam) fərdi kompüterlərdə geniş tətbiq olunur.

*Penmaus* özündə diyircəkli avtomat qələmin analoqunu ifadə edir, ucunda yazan detal yerinə, yerdəyişmə kəmiyyətinin qeydiyyatını edən hissə (detal) quraşdırılıb.

*İnfracırmızı siçan* adi siçan qurğusundan sistem blokla simsiz rabitə qurğusunun olması ilə fərqlənir.

Kompüter oyunları üçün və bəzi ixtisaslaşdırılmış imitatorlar üçün həmçinin coystiklərdən və onunla analoji olan *coypadlar, qeympadlar* və *sükan-pedallı* qurğulardan istifadə edirlər. Bu tipli qurğular, səs kartında olan xüsusi porta və ya USB portuna qoşulurlar.

## **Qrafiki verilənləri daxil etmə qurğuları.**

Qrafiki informasiyaları daxil etmək üçün skanerlər, qrafiki planşetlər (dicitayzerlər) və rəqəmli fotokameralar adlanan qurğulardan istifadə edirlər. Skanerlər bir neçə modifikasiyada istehsal olunur:

**1. Planşetli skanerlər.** Planşetli skanerlər şəffaf (aydın) və qeyri-şəffaf (tutqun) vərəqdən qrafiki informasiyaları daxil etmək üçün nəzərdə tutulmuşdur. Bu qurğuların fəaliyyət prinsipi ondan ibarətdir ki, materialın səthindən əks olunmuş (və ya aydın materialın içərisindən keçən) işıq şüası xüsusi



elementlərlə qeyd olunurlar. Qeyd edək ki, planşetli skanerlərin köməyilə kağız üzərindəki mətnləri də daxil etmək olur. Bu halda mətnin qrafiki obrazı kompüterə daxil olunur və bundan sonra xüsusi OCR (optik simvolların oxunması) proqramı vasitəsilə qrafiki obraz elektron mətnə çevrilir. Planşetli skanerlərin əsas parametrləri bunlardır:

- skanerə etmə imkanı (skanerə etmə zamanı bir düymədəki nöqtələrin miqdarı);
- məhsuldarlıq;
- dinamik diapazon (dinamik diapazon təsvirin daha işıqlı sahələrin parlaqlığının daha tutqun sahələrin parlaqlığına nisbətinin loqarifmi ilə təyin olunur);
- skanerə olunan materialın maksimal ölçüsü.

Ofis işlərində istifadə edilən planşetli skanerlərin skanerətmə imkanı tipik göstəricisi: 600-1200 dpi (dpi-dots per inch-bir düymədə olan nöqtələrin miqdarı) aralığıdır.

**2. Əl skanerləri.** Əl skanerlərinin iş prinsipi əsasən planşetli skanerin iş prinsipinə uyğun gəlir. Fərq onunla bağlıdır ki, skanerləşmənin müntəzəmliyi və dəqiqliyi qeyri-qənaətbəxş təmin olunur. Əl skanerinin, skanerətmə imkanı 150-300 dpi-dən ibarətdir.

**3. Barabanlı skaner.** Bu tip skanerlərdə skanerləşmə üçün nəzərdə tutulan əsas material yüksək sürətlə fırlanan barabanın silindrik səthinə bərkidilir. Bu tip qurğu fotoelektron artırıcıları sayəsində ən yüksək skanerətmə imkanı (2400-5000 dpi aralığında) təmin edirlər. Onları yüksək keyfiyyətə malik olan ilkin təsvirlər üçün istifadə edirlər, lakin onlar az xətti ölçüyə (fotoneqativlər, slaydlar və b.) malik olan təsvirlər üçün yaramır.

**4. Forma skanerləri.** Belə skanerlər, verilənləri mexaniki və ya əl ilə doldurulmuş standart formalardan daxil etmək üçün nəzərdə tutulmuşlar. Belə zərurət əhalinin siyahıya alınmasında, səsvermənin nəticələrinin hesablanmasında və anket verilənlərinin analizində meydana gəlir.

Forma skanerlərindən skanirləşmənin yüksək dəqiqliyi tələb olunmur.

**5. Ştrix-kod skanerlər.** Əl skanerlərinin bu növü, ştrix-kod şəklində kodlaşmış verilənlərin daxil edilməsi üçün nəzərdə tutulmuşdur. Belə qurğular pərakəndə ticarət şəbəkələrində geniş tətbiq olunurlar.

**Qrafiki planşetlər (dicitayzerlər).** Bu qurğular qrafiki informasiyaları daxil etmək üçün nəzərdə tutulmuşdur. Qrafiki planşetlərin fəaliyyətinin bir neçə müxtəlif prinsipləri vardır, lakin onların hamısının əsasında planşetə nəzərən xüsusi qələmin yerdəyişməsinin qeydiyyatı durur. Belə qurğular rəssamlar və illüstratorlar üçün əlverişlidir. Çünki, bu onlara ənənəvi alətlər (karandaş, pero, fırça) vasitəsilə qazanılmış adi (adət halını almış) qaydalarla ekran təsvirlərini yaratmağa imkan verir.

### **Verilənləri xaricətmə qurğuları.**

Verilənləri xaricətmə qurğusu kimi sənədlərin kağız və şəffaf daşıyıcılar üzərində nüsxələrini əldə etməyə imkan verən çap qurğuları (printerlər) istifadə olunur. Fəaliyyət prinsiplərinə görə printerlər *matrisli*, *lazer*, *işıqdiodlu* və *şırnaqlı* printerlərə ayrılırlar.

**Matrisli printerlər.** Bu printerin iş mexanizmi 1964-cü ildə Yaponiyanın Seiko Epson şirkəti tərəfindən ixtira edilib. Çap iynəciklərdən ibarət matrislər vasitəsi ilə həyata keçirilir. İynələr lazım olan simvola uyğun şəkildə düzülür və rəngli lent vasitəsi ilə simvollar kağızın üzərinə vurulur. Çap olunan məlumatın keyfiyyəti matrisdə olan iynələrin sayından asılıdır. Standart olaraq 9,12,14,18 və 24 iynəli matrislər mövcuddur. Bu printerlər maddi cəhətdən sərfəli olduğuna görə hal-hazırda da işlədilir.

Matrisli printerlərin işinin məhsuldarlığı saniyədə çap olunan işarələrin miqdarı, sayı ilə qiymətləndirilir (cps-characters per second). Matrisli printerlərin rejimləri bunlardır: draft-qaralama çap rejimi, normal-adi çap rejimi və NLQ (Near

Letter Quality) rejimi, sonuncu rejim çap makinasının keyfiyyətinə yaxın, çap keyfiyyətini təmin edir.

**Lazer printerləri.** Lazer printerlərində çap 1938-ci ildə Çester Karlson tərəfindən ixtira edilən lazer vasitəsi ilə həyata keçirilir. Nəticədə təsvir daha dəqiq olur və printer daha sürətli işləyir. Bu texnologiyanı 1972-ci ildə Xerox şirkəti tətbiq edə bildi. EARS adlanan ilk lazer printerini istehsal etdi. Lazer printerləri yüksək çap keyfiyyətini təmin edir, bir çox hallarda isə poliqrafik çapı da ötürür keçir. Onlar, həmçinin yüksək çap etmə sürətilə fərqlənilirlər. Matrisli printerlərdə olduğu kimi, yekun təsvirlər ayrı-ayrı nöqtələrdən formalaşır.

Lazer printerlərinin əsas parametrlərinə aşağıdakılar aiddir:

- düymədə nöqtələri yerləşdirmə imkanı, dpi (dots per inch-düymədə nöqtələr)
- məhsuldarlıq (dəqiqədə səhifələr);
- istifadə olunan kağızın formatı;
- özünün operativ yaddaşının həcmi.

Lazer printerlərinin əsas üstünlükləri yüksək keyfiyyətli təsvirlərin alınması imkanları ilə bağlıdır. Orta sinif modellər 600 dpi çap, yarı peşəkar modellər-1200 dpi, peşəkar modellər 1800 dpi çap imkanını təmin edir.

**İşıq diodlu printerlər.** İşıq diodlu printerlərin iş prinsipi lazer printerlərin iş prinsipinə oxşayır. Fərq ondan ibarətdir ki, işığın mənbəyi burada lazer başlığı deyil, işıq diodlarının xətkəsidir. Belə ki, bu xətkəş çap olunan səhifənin bütün eni üzrə yerləşir üfqi açılmaları formalaşma mexanizminə zərurət aradan qalxır və bütün konfigurasiya daha sadə, etibarlı və ucuz alınır. İşıq diodlu printerlər üçün çap imkanının tipik həcmi 600 dpi-dir.

**Şırnaqlı printerlər.** Şırnaqlı çap qurğularında təsvirlər boyayıcının kağız üzərinə düşən damcısından yaranan ləkədən formalaşır. Boyayıcının mikrodamcı tullaması, pezokristal vasitəsilə yerinə yetirilir. Bu tip modellərdə damcı,

pyezoelektrik effekt nəticəsində silkmə ilə tullanılır. Bu metod sferik damcıya yaxın daha stabil damcı formasını təmin etməyə imkan verir.

Təsvirlərin çapının keyfiyyəti çox hallarda damcının formasından və onun ölçüsündən, həmçinin maye boyayıcının kağız üzərinə hopmasının xarakterindən asılı olur. Bu şərtlərdə əsas rolu boyayıcının yapışqanlıq xassəsi və kağızın xassələri oynayır.

Şırnaqlı çap qurğularının müsbət xassələrinə, nisbətən az miqdarda hərəkət edən mexaniki hissələrini, uyğun olaraq qurğunun sadəliyini, etibarlılığını və onun nisbətən aşağı dəyərini də aid etmək lazımdır. Lazer printerlə müqayisədə əsas çatışmamazlığı əldə olunmuş çap imkanının qeyri-sabitliyidir ki, bu da ağ-qara yarımton (şəkildə açıq tondan tünd tona keçidi təşkil edən rəng) çapda onların tətbiq edilmə imkanlarını məhdudlaşdırır.

Bu gün şırnaqlı printerlər rəngli çapda çox geniş tətbiq olunur. Konstruksiyasının sadəliyi sayəsində onlar dəyər (qiymət) göstəricilərinə görə rəngli lazer printerləri ötüb keçirlər. 600-dpi-dən yuxarı çap imkanında onlar fotokimyəvi metodlarla alınmış rəngli təsvirlərin keyfiyyətindən üstün olan rəngli şəkillər əldə etməyə imkan verirlər. Şırnaqlı printeri seçdikdə bir şəklin, yazının çap dəyəri, qiyməti parametrini hökmən nəzərə almaq lazımdır. Əlavə olaraq, şırnaqlı çap qurğularının qiyməti lazer printerlərindən nəzərəcarpacaq dərəcədə aşağıdır. Onlarda bir şəklin, yazının çap dəyəri, xüsusi kağız lazım gəldiyindən on dəfələrlə yüksək ola bilər.

#### **1.4 Alqoritm anlayışı, alqoritmin əsas xassələri, tipləri və təsvir üsulları.**

Alqoritm sözü görkəmli alim Əbu Muhəmməd İbn Musa əl Xarəzminin (787-850) adının təhrif olunmuş formasıdır. Belə ki, alimin "Kitab əl cəbr və əl-muqabala" əsəri XII əsrdə latın dilinə tərcümə edilmiş və bu zaman tərcüməçi alimin adındakı əl

Xarəzmi sözünü təhrif edərək Alqoritmi kimi verilmişdir və əsəri “Algoritmi de numero indorum”, yəni “Alqoritmi hind hesabı haqqında” adlandırmışdır. Bu tərcümə sayəsində alqoritm sözü Avropa dillərində, sonradan isə əksər dünya dillərində, o cümlədən azərbaycan dilində bir termin kimi istifadə olunmaqdadır<sup>1</sup>.

Orta əsrlərdə Avropada alqoritm dedikdə onluq say sistemində ədədlərlə hesablama qaydaları başa düşülürdü. Müasir dövrdə isə bu anlayış daha geniş mənada, verilmiş məsələni həll etmək və ya müəyyən məqsədə çatmaq üçün zəruri olan əməliyyatlar ardıcılığının yerinə yetirilməsi haqqında icraçıya verilən dəqiq və aydın göstərişlərin-əmərlərin təsviri kimi başa düşülür. Alqoritm əməlləri sonlu sayda olub ardıcıl yerinə yetirilir. Hər bir alqoritm konkret icraçı üçün qurulur. İcraçı alqoritmi yerinə yetirməyə qadir olmalıdır. Başqa sözlə alqoritm bütünü əməlləri isifadəçi üçün “başa düşülən” olmalıdır. Məsələn, aydındır ki, birinci sinif şagirdi riyaziyyatın ilk alqoritmələrindən hesab olunan Evklid alqoritmını yerinə yetirə bilməz.

Alqoritm anlayışı bir termin kimi əksər elm sahələrində, o cümlədən informatikada istifadə olunur. Alqoritm informatikanın fundamental anlayışlarından biridir. Belə ki, informatikanın əsas tədqiqat obyektini ilkin informasiya əsasında yeni rəqəmli informasiyanın yaradılmasını, informasiyanın saxlanılmasını, işlənilməsini, operativ mübadiləsini, təhlükəsizliyini, insan əməyinin avtomatlaşdırılmasını təmin edən, effektiv və kompüterdə icra olunan alqoritmələrin yaradılmasıdır. Bu baxımdan informatikada alqoritm anlayışına aşağıdakı kimi tərif vermək olar:

*Alqoritm – verilmiş məsələni həll etmək üçün ilkin verilənlərlə icra olunan hesabi və məntiqi əməliyyatların sonlu sayda ardıcılığıdır.*

---

<sup>1</sup> Bu tərcümə sayəsində avropalılar onluq say sistemilə tanış olmuşdur.

Alqoritmlər çox rəngarəng olsalar da, onların hamısı aşağıdakı ümumi xassələrə malikdir:

**1. Diskretlilik xassəsi.** Hər bir alqoritm məsələnin həll prosesini sadə addımların yerinə yetirilməsi ardıcılığı şəklində ifadə edir və hər bir addımın yerinə yetirilməsi üçün sonlu zaman fasiləsi tələb olunur, yəni başlanğıc verilənlərlə icra olunan hesabi və məntiqi əməliyyatların yerinə yetirilməsi və nəticənin alınması zamana görə diskret yerinə yetirilir.

**2. Müəyyənlik xassəsi.** Hər bir alqoritm dəqiq, birqiymətli olmalıdır. Bu xassəyə əsasən alqoritm yerinə yetirildikdə istifadəçinin və onun istifadə etdiyi kompüterdən asılı olmayaraq eyni nəticə əldə edilməlidir.

**3. Kütləvilik xassəsi.** Müəyyən sinif məsələnin həlli üçün qurulmuş alqoritm bu sinfə aid olan yalnız başlanğıc qiymətləri ilə fərqlənən bütün məsələlərin həllini təmin etməlidir. Məsələn,  $ax^2 + bx + c = 0$  kvadrat tənliyi üçün qurulmuş alqoritm  $a, b, c$  – nin ixtiyari qiymətləri üçün məsələni həll edir.

**4. Nəticəlilik və sonluluq xassəsi.** Alqoritm sonlu sayda addımdan sonra başa çatmalı və verilmiş məsələnin həlli tapılmalıdır.

Alqoritm aşağıdakı üsullarla təsvir olunur:

- Sözlə
- Blok-sxemlə
- Alqoritmik dillə
- Qrafalarla

Alqoritmlərin ən sadə təsvir forması təbii dillə - *sözlə* yazılış formasıdır. Alqoritmın mahiyyətinin izahını vermək üçün bu üsul çox əlverişli olur. Məsələn, verilmiş iki  $A$  və  $B$  ədədlərindən böyük olanının tapılması alqoritmını sözlə ifadə edək:


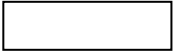

1.  $A$  ədədindən  $B$  ədədini çıxın.
2. Nəticədə mənfə qiymət alınarsa,  $B$  ədədinin böyük olmasını bildirin.




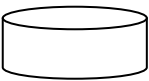
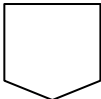
3. Müsbət qiymət alınarsa, A ədədinin böyük olmasını bildirin.

4. Sıfır alınarsa, hər iki ədədin bərabər olduğunu bildirin.

Alqoritmi əyani təsvir etmək üçün onu blok-sxem şəklində vermək məsləhətdir. Bu təsvir forması sözlə yazılışa nisbətən daha yığcam və dəqiqdir.

Blok-sxem blok adlanan aşağıdakı standart həndəsi fiqurlardan ibarət olur:

Blokun təsviri	Blokun adı
	Başlangıç/son blok
	Hesablama və ya əməliyyat bloku
	Daxiletmə bloku

	Dövr bloku
	Altprogram bloku
	Çap bloku
	Fayl bloku
	Səhifələrarası birləşdirici

Hər bir blokda alqoritmin konkret bir və ya eyni tipli bir neçə əmrini təsvir etmək olar. Bloklar bir-birilə şaquli və ya üfüqi xətlərlə və ya oxlarla əlaqələndirilir. Şaquli xəttlər üçün istiqamət olaraq yuxarıdan - aşağıya, üfüqi xəttlər üçün isə soldan- sağa qəbul olunmuşdur.

Blok-sxemi Proqram Sənədlərinin Vahid Standartı əsasında tərtib edilməlidir. Məsələn, bu standarta əsasən hesablama və ya əməliyyat blokunun minimal eni 10 mm olmalı, bir blok-sxemdə verilən eyniadlı blokların ölçüləri eyni olmalıdır. Məntiqi blok istisna olmaqla digər bloklardan yalnız bir xətt və ya ox çıxıb bilər<sup>1</sup>. Məntiqi blokdan isə iki xətt və ya ox çıxmalıdır. Blok - sxem bir səhifəyə sığışmadıqda səhifələrarası birləşdirici vasitəsilə blok-sxemi növbəti səhifədən davam etdirmək olar. Səhifələrarası birləşdiricidə birləşdirici xəttin və ya oxun haradan haraya istiqamətləndirildiyini göstərmək lazımdır.

Alqoritmin alqoritmik dillə təsvir üsulu onun kompüterdə icrası üçün tərtib olunur. Alqoritmin belə yazılışı – kompüter üçün proqram adlanır.

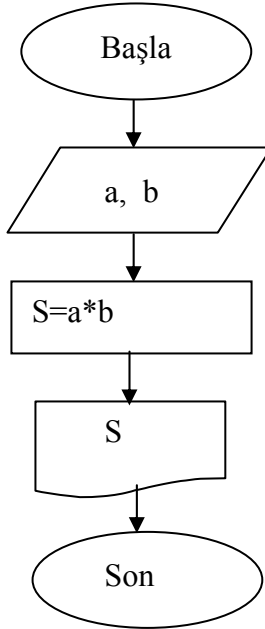
Alqoritmlər xətti, budaqlanan və dövrü struktura malik ola bilər:

*Heç bir mərhələsi buraxılmadan və təkrarlanmadan, bütün mərhələləri tam ardıcılıqla yerinə yetirilən alqoritmə xətti alqoritm deyilir.* Məsələn, Ücbucağın sahəsinin hesablanması məsələsinin alqoritm xətti alqoritmidir.

---

<sup>1</sup> Lakin bir neçə ox girə bilər.





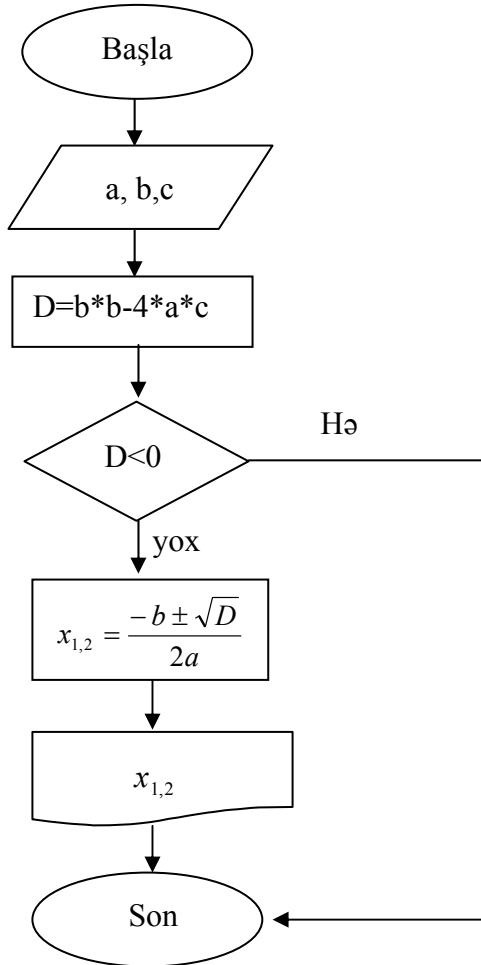
*Qoyulmuş şərtdən asılı olaraq alqoritm məsələnin həlli variantlarından birini seçməyə imkan verirsə, belə alqoritmə budaqlanan alqoritm deyilir.*

Budaqlanmadan iki halda istifadə olunur:

1. Alqoritmədə elə addımlar olur ki, müəyyən şərtdən asılı olaraq onları buraxmaq lazım gəldikdə;
2. Müəyyən şərtdən asılı olaraq alqoritm müəyyən əməliyyatlardan birini yerinə yetirdikdə.

Blok-sxemdə budaqlanma şərti məntiqi blokla göstərilir və ondan iki ox çıxır: "hə" oxu şərt ödəndikdə, "yox" oxu isə şərt ödənmədikdə yerinə yetiriləcək bloku müəyyən edir.

Məsələn, kvadrat tənliyin həqiqi kökünün hesablanması alqoritmə budaqlanan alqoritmədir.



Alqoritmin müəyyən əməliyyatları sonlu sayda təkrar oluna bilər. Təkrarlanan əməliyyatlara malik alqoritmlər dövrü strukturlu alqoritmlər adlanır. Blok - sxemdə təkrarlanan əməliyyatlar dövr bloku vasitəsilə təsvir olunur. Dövrü struktur sadə və mürəkkəb olur. Sadə dövrü struktur bir dövr blokuna, mürəkkəb struktur isə biri digərinə daxil olan iki və daha çox dövrədən ibarətdir. Məsələn, birdən yüzə qədər ədədlərin cəminin hesablanması alqoritm sadə dövrü alqoritmidir.

$s = \sum_{i=1}^{100} \sum_{k=1}^{50} (i+5)k$  məsələsinin həll alqoritmisi isə mürəkkəb dövrü alqoritmidir.

### 1.5 Kompüterlərin proqram təminatı

Proqram təminatı kompüterlərin istifadə etdiyi proqramlar toplusundan ibarətdir. Proqram isə öz növbəsində tələb olunan məsələnin həll alqoritminin kompüterin “başa düşəcəyi dildə” yazılmış təsvir formasıdır. Kompüter proqramı yerinə yetirməklə alqoritmə uyğun əməliyyatları yerinə yetirir. Müvafiq proqram təminatı kompüterini insan fəaliyyətinin ən müxtəlif sahələrində tətbiq edilməsinə imkan verir. Məhz onun sayəsində kompüterlər çoxsahəli bir əmək aləti olaraq, insan həyatının zəruri bir elementinə çevrilməkdədir.

Öz funksional təyinatına görə proqramlar üç kateqoriyaya bölünür:

1. Sistem proqramları (*System Software*)
2. Instrumental sistemlər
3. Tətbiqi proqramlar

#### Sistem proqramları

Sistem proqramları əməliyyat və istifadəçi interfeysi mühitinin yaradılmasını, kompüterin qurğularının və digər proqramların birgə fəaliyyətini, kompüter şəbəkəsinin idarə olunmasını, diaqnostika və profilaktika, bir sıra köməkçi

texnoloji əməliyyatların (məsələn, informasiyanı arxivləşdirmə, ehtiyat sürətinin yaradılması və s.) yerinə yetirilməsini təmin edir. Bu kateqoriyaya aid proqramlar kompüterin arxitekturasını nəzərə almaqla və keyfiyyət standartlarına<sup>1</sup> uyğun hazırlanır.

Sistem proqramlarını şərti olaraq iki qrupa bölmək olar:

1. Əməliyyat sistemləri
2. Xidməti proqramlar

Əməliyyatlar sistemi hər bir kompüter üçün zəruri olan, onun informasiya, proqram və aparat təminatını idarə edən, istifadəçi ilə kompüter arasında dialoqu təmin edən proqramlar kompleksindən ibarətdir. Əməliyyatlar sistemi adətən xarici yaddaş qurğusunda saxlanılır və kompüter elektrik dövryyəsinə qoşulduqda operativ yaddaşa yüklənərək, məsələnin həlli üçün tələb olunan fiziki və məntiqi ehtiyatlar<sup>2</sup> ayırır və istifadəçiyə rahat və sadə işçi interfeysi təklif edir.

Xarici yaddaş qurğusu olmadığından ilk fərdi kompüterlər əməliyyatlar sisteminə malik olmamışlar. Yalnız bu qurğular yaradıldıqdan sonra faylların idarə olunması üçün əməliyyatlar sisteminin yaranması zərurəti yaranmışdır. Qeyd edək ki, fərdi kompüterlər üçün ilk əməliyyatlar sistemi Harri Kilde<sup>3</sup> tərəfindən yaradılan CP/M (Control Program/Microcomputer) əməliyyatlar sistemi hesab olunur. IBM PC tipli kompüterlər 1982-ci ildən 1995-ci ilə qədər əsasən MS-DOS (Microsoft Discs Operating System)<sup>4</sup>, PC DOS<sup>5</sup>, sonradan isə Windows seriyasında olan Windows-

---

<sup>1</sup> Sistem proqramlarının keyfiyyəti onun etibarlılığı ilə, çoxfunksiyalılığı ilə, istifadə üçün rahat və effektiv olması ilə təyin edilir.

<sup>2</sup> Fiziki ehtiyatlara kompüterin qurğuları və perferiya qurğuları, məntiqi ehtiyatlara isə proqramlar, fayllar və s. aiddir.

<sup>3</sup> Sonradan o, Digital Research kompaniyasını yaratmışdır.

<sup>4</sup> Bu 15 ildə o, MS-DOS 1.0 versiyasından MS-DOS 6.22-yə böyük təkamül yolu keçmişdir.

<sup>5</sup> PC DOS əməliyyatlar sistemini IBM və DR DOS firmaları birgə tərtib etmişdir.

95/98/2000/XP<sup>1</sup>, Windows Vista UNIX, OS/2, Linux əməliyyatlar sistemindən istifadə edilir. Apple PC kompüterləri isə Macintosh əməliyyatlar sistemindən istifadə edir.

Kompüterlərdə müxtəlif arxitekturalı və funksional imkanlı əməliyyatlar sistemlərindən istifadə olunur. Əməliyyatlar sisteminin normal iş rejimi müvafiq konfigurasiyalı aparat təminatı olduqda təmin olunur. Məsələn, maynfreymlərdə və fərdi kompüterlərdə fərqli əməliyyatlar sistemləri istifadə olunur. Əməliyyatlar sistemində əməliyyatların yerinə yetirilməsi sürəti kompüterin operativ yaddaşından və istifadə etdiyi prosessorun asılıdır. Bu baxımdan, əməliyyatlar sisteminin istifadə etdiyi prosessorun tərtibinə görə təsnifatını vermək olar. Fərdi kompüterlərdə 16, 32, 64 -tərtibli əməliyyatlar sistemlərindən istifadə olunur. Məsələn, İBM tipli kompüterlərdə istifadə olunan MS-DOS, PC-DOS, FreeDOS əməliyyatlar sistemləri 16 tərtibli, Windows 2000 sistemi 32 tərtibli, Windows XP 64 Bit Editon sistemi 64 tərtibli əməliyyatlar sistemidir.

Əməliyyatlar sistemi yerinə yetirdikləri məsələlərin miqdarına görə birməsəlali və çoxməsəlali sistemlərə bölünür. Birməsəlali əməliyyatlar sistemində cari anda yalnız bir məsələ yerinə yetirilir. Bu cür əməliyyatlar sistemlərinin tipik nümunəsi kimi MS-DOS əməliyyatlar sistemini misal göstərmək olar.

Çoxməsəlali əməliyyatlar sistemi kompüterdən istifadəçilərin multiproqram vaxt bölgüsü rejimində kollektiv istifadəni təmin edir. Bu zaman cari anda kompüterin operativ yaddaşında bir neçə proqram və məsələlər toplusu ola bilər ki, mikroprosessor kompüterin ehtiyatlarını onların arasında effektiv bölüşdürür. Belə əməliyyatlar sisteminə misal olaraq UNIX, OS/2, Windows 9x/XP/Vista, Windows NT/Server və digər əməliyyatlar sistemlərini göstərmək olar.

---

<sup>1</sup> Windows-95. Yüz milyonlarla insanlar üçün «elektron informasiya dünyasına pəncərə» açmışdır.

İstənilən bir məsələni kompüterdə həll edərkən çoxsaylı elementar əməliyyatları yerinə yetirmək tələb olunur. Əməliyyatlar sistemi belə elementar, amma mürəkkəb əməliyyatları kompüterin daxilində olan qurğularda həll edərək bütün iş prosesini istifadəçidən gizli saxlayır, istifadəçiyə kompüterdə işləmək üçün əlverişli interfeys yaradır. İstifadəçiyə təklif etdiyi işçi interfeysinə görə əməliyyatlar sistemləri əmr və qrafiki interfeysli əməliyyatlar sistemlərinə bölünür. Əmr interfeysli əməliyyatlar sistemlərində hər bir əməliyyat müvafiq əmri klaviaturadan daxil etməklə yerinə yetirilir.

Qrafiki əməliyyatlar sistemində əmrlər əsas və kontekst menyu sistemi, alətlər panelləri və dialoq pəncərələri<sup>1</sup> vasitəsilə yerinə yetirilir. Qrafiki əməliyyatlar sistemi daha rahat və sadə işçi interfeysinə malikdir. İlk qrafiki əməliyyat sistemi 1983-cü ildə Apple firmasında Apple PC kompüterləri üçün hazırlanmış Macintosh<sup>2</sup> əməliyyatlar sistemi olmuşdur. IBM PC tipli kompüterlər üçün isə ilk qrafiki əməliyyatlar sistemi Windows-95 əməliyyatlar sistemi hesab olunur. Hal-hazırda bu tip kompüterlərdə Windows ailəsindən olan Windows-98/2000/XP<sup>3</sup>, Windows Vista və Linux əməliyyatlar sistemindən istifadə edilir.

### **Xidməti proqram təminatı**

Praktikada kompüter istifadəçiləri bir sıra hallarda proqram-texniki problemlərlə qarşılaşır. Bu problemlərin səbəblərinin müəyyən edilməsi və aradan qaldırılması xidməti proqramlar vasitəsilə yerinə yetirilir.

---

<sup>1</sup> Dialoq pəncərələrində əmrin yerinə yetirilməsi üçün parametrlər müəyyən olunur və ya əməliyyatın yerinə yetirməsi

<sup>2</sup> və ya qısa olaraq Mac

<sup>3</sup> Windows-95-in (1995-ci il) sələfi hesab etmək olar. Yüz milyonlarla insanlar üçün «elektron informasiya dünyasına pəncərə» açmışdır.

Xidməti proqramlar kompüterin aparat hissəsinin işinə nəzarət edir, kompüterdə baş verən nasazlığı və onun harada baş verdiyini aşkar edir, əməliyyatlar sisteminin əvvəlki vəziyyətinin bərpa olmasını, icra zamanı qarşıya çıxan səhvləri düzəltməyə, sərbəst proqram modulları arasında qarşılıqlı əlaqəni təmin etməyə, disklərin formatlaşdırılmasına, CD və DVD disklərinə yazılması və sairə bu kimi texnoloji işlərin həyata keçirilməsinə imkan verir.

Xidməti proqramlar qrupuna interfeys proqramları, diaqnostika, antivirus və şəbəkəyə xidmət, arxivləşdirmə proqramları, drayverlər aiddir.

İnterfeys proqramları əməliyyatlar sistemi ilə birgə fəaliyyət göstərərək istifadəçiyə daha sadə və rahat interfeys təqdim edir. Məsələn, MS-DOS əməliyyatlar sistemi üçün Norton Commander (NC)<sup>1</sup>, 1985-ci ildə isə Windows 1.0, 1990, 1992, 1993-cü illərdə Windows 3.0/3.1/3.11 qrafiki proqram örtükləri<sup>2</sup> yaradılmışdı.

Diaqnostika proqramları kompüterin qurğularını diaqnostika edərək yaranan nasazlıqları aşkar edir və imkan daxilində aradan qaldırmağa xidmət göstərir. Məsələn, maqnit disklərin diaqnostikası üçün *Scandisk*, *Norton Disk Doctor (NDD)*, diskin defraqmentləşdirilməsi üçün *Defrag*, *Speed Disk*, diskdən ləğv olunmuş fayl və qovluqların bərpası üçün *Easy Recover*, *Tiramisu*, *Drive Rescue* proqramlarından istifadə etmək olar. CD və DVD disklərinə yazılma proqramları diskin sürətinin alınmasını, multimedia disklərin, avtomatik yüklənən proqram disklərin hazırlanmasını və s. texnoloji əməliyyatları yerinə yetirirlər. Bu tip proqramlara *RegCleaner*, *Customizer XP Easy CD Creator*, *WinOnCD*,

---

<sup>1</sup> Bu örtüyün işləmə mahiyyəti ondan ibarətdir ki, klaviaturanın müəyyən düymələrini basmaqla bu və ya digər əmri yerinə yetirmək mümkündür.

<sup>2</sup> Məhz onları ilk qrafiki əməliyyat sistemi olan Windows-95-in (1995-ci il) sələfi hesab etmək olar.

---

*Nero Burning ROM, Direct CD* və s. proqramları göstərmək olar.

Arxivləşdirmə proqramları faylları xüsusi riyazi alqoritm əsasında sıxaraq daha kiçik həcmdə, bir və ya bir neçə arxiv faylında saxlayır. Hal-hazırda faylları arxivləşdirmək üçün ARJ, ZIP, WINZIP, RAR, WINRAR proqramlarından istifadə edilir.

Antivirus proqramları isə kompüterin virusa yoluxmasının qarşısını alır, vaxtaşırı faylları yoxlayır və onlarda əmələ gəlmiş virusları aradan qaldırır.

### **İnstrumental sistemlər**

İnstrumental sistemlər yeni sistem və tətbiqi proqram vasitələri yaratmaq üçün istifadə edilir. Instrumental sistemlər maşın dilindən fərqli və istifadəçi üçün asan olan alqoritmik dillərdə işləməyi təmin edir.

Kompüter yalnız maşın dilini başa düşdüyü üçün alqoritmik dildə yazılan proqram icra prosesindən qabaq mütləq maşın koduna çevrilməlidir. Bu məqsəd üçün instrumental sistemlər qrupuna daxil olan translyator adlanan proqramlar kompleksindən istifadə olunur.

Translyatorun işi iki üsulla təşkil oluna bilər: interpretasiya və kompilyasiya yolu ilə. Buna uyğun olaraq çox vaxt translyatoru interpretator və ya kompilyator adlandırırlar. Onlar arasında fərq çevrilən proqramın mətninin müxtəlif üsullarla emal olunmasıdır.

İnterpretator proqramın operatorlarını bir-bir təhlil edir və onu bütövlükdə operativ yaddaşa yükləyir. Nəticədə proqramın işləmə vaxtı uzanır. Bu isə fərdi kompüterdən istifadə edən istifadəçi üçün əlverişli deyil.

Kompilyator isə bütün proqramı maşın koduna çevirərək mövcud səhvlər haqqında məlumatları vaxtında istifadəçiyə çatdırır. Burada operatorların təhlil edilməsi və maşın koduna çevrilməsi birdəfəlik aparılır. Odur ki, kompüterin işləmə sürəti artır, proqramın icra olunması



prosesdən asılı olmur. Nəticədə proqramın operativ yaddaşa yüklənməsinə ehtiyac duyulmur, operativ yaddaşdan digər məqsədlər üçün istifadə etməyə imkan yaranır.

### **Tətbiqi proqramlar**

İnsan fəaliyyətinin konkret fəaliyyət sahələrinə aid məsələləri həll etmək üçün nəzərdə tutulan proqramlara tətbiqi proqramlar deyilir. Tətbiqi proqramlar 3 formada – proqram, proqramlar paketi və standart proqramlar kitabxanası şəklində istehsal olunur.

Tətbiqi proqramlar paketi müəyyən sinif məsələləri həll etmək üçün nəzərdə tutulmuş elə proqramlar kompleksidir ki, kompleksin komponentlərindən biri idarəedici rolu oynayaraq, paketin digər bütün proqramlarının bir-biri ilə əlaqəsini təşkil edir.

Standart proqramlar kitabxanası isə kompüterdə riyazi funksiyaların hesablanmasını, standart mahiyyətli məsələlərin həllini, verilənlərin emalını və s. bu kimi işləri yerinə yetirən proqramlardan ibarətdir. Belə proqramlar adətən üsul yönümlü proqramlarla birlikdə istifadə olunur.

Müasir dövrdə fərdi kompüterlərdə müxtəlif təyinatlı yüz minlərlə tətbiqi proqramlardan istifadə edilir. Funksional imkanlarına görə tətbiqi proqramlar ümumi təyinatlı, üsul yönümlü və problem yönümlü proqramlara bölünür. Ümumi təyinatlı proqramlara mətn<sup>1</sup>, elektron cədvəl<sup>2</sup> prosessorları,

---

<sup>1</sup> Mətn prosessorları elektron mətn sənədlərinin hazırlanmasını, redaktəsini və formatlaşdırılmasını təmin edir.

<sup>2</sup> Elektron cədvəl prosessorları cədvəllərin yaradılmasını, ədədi verilənlərin təhlilini, verilənlərlə düsturlar əsasında müxtəlif əməliyyatlar aparmağı təmin edir. Onlara misal olaraq SuperCalc-4, MS Excel, Lotus 1-2-3 və s. proqramlarını göstərmək olar.

---

nəşriyyat sistemləri<sup>1</sup>, verilənlər bazasının idarəetmə sistemləri, qrafiki redaktorlar, təqdimat, multimedia proqramları, kompüter oyunları və s. aiddir.

Üsul yönümlü proqramlar müəyyən sinif məsələlərin həll üsulları əsasında fəaliyyət göstərir. Onlara statistik təhlil proqramlarını (məsələn, Statistica, StatFi 2007 və s. ), riyazi proqram paketlərini (məsələn, Matlab, Mathcad, Mathematica, Maple və s) və s. göstərmək olar.

Problem yönümlü proqramlar konkret ixtisaslı istifadəçilər üçün onların peşə fəaliyyətinə aid müəyyən sinif məsələlərin həlli üçün yaradılır. Onlar tibbdə, bank işində, müəssisələrin və istehsalatın avtomatik idarəetmə işində və s. geniş tətbiq olunur. Belə proqramlar adətən Avtomatlaşmış İşçi Yeri kimi fəaliyyət göstəriirlər. Məsələn, İRBİS 64 proqram paketi kitabxanaların avtomatlaşdırılması üçün istiafadə olunur və onun əsasında kataloqlaşdırıcı, kitab verilişi, oxucu, administrator, komplektləşdirici və s. Avtomatlaşmış İşçi Yerləri təşkil olunur.

## 1.6 Alqoritmik dillər

Kompüter istehsalının ilk dövründə proqramlar maşın dilində yazılırdı. Maşın dili kompüterin “başa düşdüyü” kodlarla ifadə olunmuş əmrlərdən ibarət olub, konkret kompüterin arxitekturasından asılı idi. Hər bir əmrdə ümumi şəkildə, aparılacaq əməliyyatın məzmunu haqqında məlumat, üzərində maşın əməliyyatı aparılacaq başlanğıc verilənlərin yerləşdiyi yer-ünvan, nəticənin ünvanı və bu əmrdən sonra yerinə yetiriləcək əmr haqqında məlumat verilirdi. Maşın

---

<sup>1</sup> Nəşriyyat sistemləri kitab, qəzet, jurnalların və digər nəşr məhsullarının hazırlanmasında geniş istifadə olunur. Bu tip proqramlara misal olaraq Adobe PageMarker, QuarkXpress proqramlarını göstərmək olar. Adətən, mətnlər mətn redaktorlarının birində yığılır, sonra yuxarıda sadalanan proqramların biri vasitəsilə tərtibat işləri yerinə yetirilir.

dilində proqramların yaradılması və onların kompüterdə yerinə yetirilməsi kifayət qədər mürəkkəb və vaxt aparan idi. Bu səbəbdən kompüter texnologiyasının inkişafının sonrakı mərhələsində təbii dilə daha yaxın olan simvolik dillər yarandı. Belə ki, ikinci nəsil kompüterlərin yaranması, konkret maşının yox, qoyulmuş məsələnin xüsusiyyətlərindən asılı olan dillərə ehtiyac yaratdı. Bu dillərə formal dillər və ya sadəcə alqoritmik dillər deyilir və bir sıra üstünlüklərə malikdir. Bu dillər əyani olub, onlarla ixtiyari alqoritmi asanlıqla ifadə etmək mümkündür. Alqoritmik dillər alqoritmin birqiymətliliyini, mürəkkəb alqoritmin daha sadə alqoritmələrin vəhdəti şəklində ifadə edilməsini təmin edir. İlk mükəmməl alqoritmik dil 1954-cü ildə İBM firmasında Con Bekusun rəhbərlik etdiyi qrupun yaratdığı FORTRAN dili idi. Bu dilin adı FORTRAN-FORMulae TRANslation - formulaların tərcüməsi sözündən götürülmüşdür. Bu dil çox sadə struktura malik olduğundan ondan hal-hazırkı vaxta qədər istifadə olunur. Fortrandə proqram operatorlar ardıcılığı şəklində yazılır. Bu dildə yazılan proqram bir və ya bir neçə seqmentlərdən (alt proqram) ibarət olur. Bütün proqramın işini idarə edən seqment əsas proqram adlanır.

Fortran dili elmi və mühəndis texniki hesablama sahələrində istifadə edilmək üçün nəzərdə tutulmuşdu. Lakin bu dildə budaqlanan strukturlu məsələlər (istehsal prosesinin modelləşdirilməsi və s.), bəzi iqtisadi məsələlər və redaktətmə məsələləri (cədvəl, arayış və s. qurulması) üçün proqramlar da qurula bilər. Sonrakı illərdə bu dilin müxtəlif modifikasiyaları yaradılmışdı.

1960-cı ildə Alqol-60 (Alhoritmic Language-alqoritmik dil) dili, 1966-cı ildə isə Fortran dili əsasında Dartmut kollecinin hesablama mərkəzində Basic dili (BASIC-Beginner's Allpurpose Symbolic Instruction Code-yeni başlayanlar üçün çoxməqsədli simvolik əmrlər dili) yaradıldı. Basic dili Visual Basic.Net versiyasına qədər böyük təkamül

yolu keçmişdir. Hal-hazırda Visual Basic.Net dilindən qrafiki interfeysli proqram əlavələrin yaradılmasında geniş istifadə olunur.

Fortran, Alqol-60 dillərilə paralel olaraq intensiv inkişaf edən elm və texnikanın yeni sahələrinin tələbatını ödəmək üçün yeni proqramlaşdırma dilləri yaradılmışdır. Məsələn, 1957-ci ildə riyazi verilənlərin emalı üçün APL (Application Programming Language) dili, 1959-cu ildə İBM firması tərəfindən böyük həcmli verilənlər massivinin emalı üçün Cobol (Common Business Oriented Language) dili, mətn informasiyanın emalı üçün 1962-ci ildə Snobol<sup>1</sup>, 1969-ci ildə çoxluqlar üzərində əməliyyatlar aparmaq üçün SETL dili yaradılır.

Üçüncü nəsillə kompüterlərin yaranması, universal alqoritmik dillərin yaradılması məsələsini qarşıya qoydu. Bu cür dillərin yaradılması üçün edilən cəhdlərdən biri nəticəsində İBM firması tərəfindən PL/1 (Programming-Language/1-proqramlaşdırma dili-1) dili yaradılır. Bu dil Fortran, Alqol və Cobol dillərinin əsasında yaradılmış və bu dillərin üstünlüklərini özündə birləşdirmişdi.

1971-ci ildə Nikuls Virt tərəfindən Paskal<sup>2</sup> dili yaradılır. Bu dil struktur proqramlaşdırma ideyasının, yəni proqramın kiçik, dəqiq təyin edilmiş prosedurlardan tədricən qurulması ideyasının həyata keçirilməsini təmin edən ilk dildir. 1983-cü ildə Pentaqonda Ada<sup>3</sup> dili yaradılır. Bu dil əsasən hərbi və böyük sənaye layihələrinin yaradılmasında istifadə edilir. Dil birinci proqramlaşdırıcı – qrafikada Ada Lavlaysın şərəfinə adlandırılmışdır. 1972 -ci ildə Kreniqan və Ritçi tərəfindən C dili, 1986-cı ildə onun bazasında Brayn Straustrup tərəfindən problem yönümlü C<sup>++</sup> dili yaradılır. C/C<sup>++</sup> dili-universal dil olub, sistem proqramlarının yaradılmasında, o cümlədən geniş

---

<sup>1</sup> Onun müasir sələfi 1974-cü ildə yaradılmış Icon dilidir.

<sup>2</sup> Bu dil XVII əsr böyük fransız alimi Paskalın şərəfinə adlandırılmışdır.

<sup>3</sup> 1995-ci ildə dilin yeni versiyası ADA -95 yaradılmışdır.

yayılmış UNIX əməliyyat sistemi üçün kodların yazılmasında geniş istifadə olunur. 1995-ci ildə Sun Microsystems kompaniyasında Java dili yaradılır. Qeyd etdiyimiz dillərdən başqa dillər də mövcuddur və bu dillərin yaradılması prosesi davam etdirilir.

XX əsrin 90-cı illərindən başlayaraq Web proqramlaşdırma dilləri meydana gəldi. Web proqramlaşdırma dilləri Web səhifələrin idarə olunmasında istifadə olunur. Onlara misal olaraq HTML, XML, JavaScript, VbasicScript, Perl, Payton göstərmək olar.

### 1.7. Kompüter şəbəkələri

Kompüter şəbəkələri rabitə vasitələrinin köməyiylə bir-birilə əlaqələndirilmiş və müəyyən protokollarla informasiya mübadiləsi etmək qabiliyyətinə malik kompüterlər, serverlər<sup>1</sup> və periferiya qurğuları sistemidir. Kompüterlər arasında məsafədən asılı olaraq rabitə vasitəsi olaraq kabel, telefon rabitəsindən, radiorabitədən, infraqırmızı və lazer şüalandırmadan istifadə olunur.

Şəbəkələr tətbiq olunduğu əhatə dairəsinə görə lokal və qlobal kompüter şəbəkələrinə bölünür. Lokal kompüter şəbəkələri məhdud ərazidə, adətən bir müəssisə daxilində fəaliyyət göstərir və şəbəkə ehtiyatlarından<sup>2</sup> birgə istifadəni və kompüterlər arasında interaktiv əlaqəni təmin edir. Rabitə vasitəsi olaraq əsasən kabeldən istifadə olunur.

Lokal kompüter şəbəkələrinin tətbiqi ayrı-ayrı qurğulara qənaət etməyə, operativ informasiya mübadiləsinə nail olmağa imkan yaradır. Məsələn, şəbəkə printeri şəbəkənin istənilən kompüterindən informasiyanın çapını təmin edir ki, hər bir kompüterə printer qoşulması zərurəti aradan qalxır. İnformasiyadan, proqram və əlavələrdən birgə istifadə

---

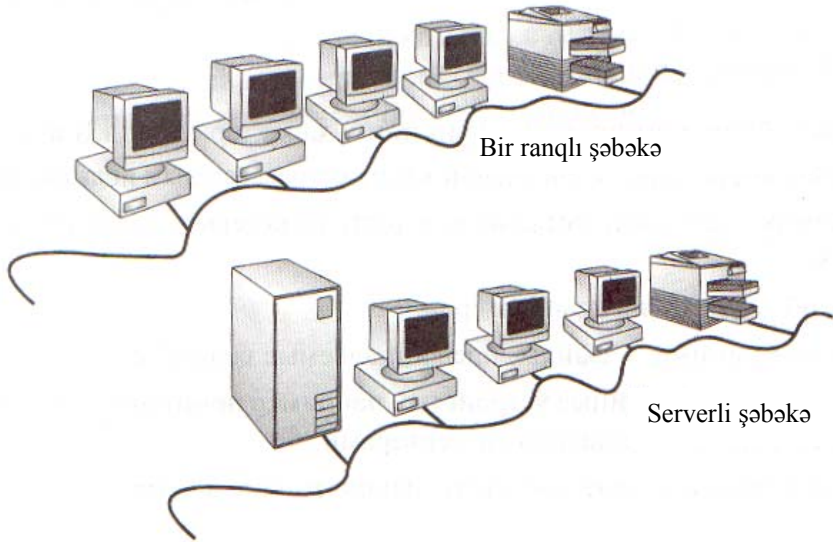
<sup>1</sup> Server-yüksək texniki göstəricilərə malik kompüterdir.

<sup>2</sup> Şəbəkə ehtiyatları dedikdə informasiya ehtiyatı, proqram və əlavələr, periferiya qurğuları nəzərdə tutulur.

informasiya daşıyıcılarına qənaətə, hamının eyni proqram və əlavələrdən istifadəsinə şərait yaratmaqla, iş prosesi zamanı meydana çıxan bir sıra problemlərin məsələn, proqram versiyasının uyğunsuzluğu, şriftlərlə bağlı və s. problemləri avtomatik aradan qaldırır.

Müəssisənin xarakterindən asılı olaraq lokal kompüter şəbəkələri öz növbəsində bir rəngli, serverli və kombinasi edilmiş şəbəkələrə bölünür.

Şək. 1



Bir rəngli şəbəkələr az saylı istifadəçilər üçün nəzərdə tutulmuşdur. Bu şəbəkədə kompüterlərin sayı ondan artıq deyildir və informasiyanın etibarlı surətdə qorunmasına nail olmaq çətindir. Şəbəkədə vahid administrasiya xidməti olmadığından hər bir istifadəçi kompüterin ehtiyatlarını ümumi istifadə üçün özü müəyyən edir. Server əsasında yaradılmış şəbəkə böyük həcmli informasiyanı eyni zamanda çoxsaylı istifadəçinin birgə istifadəsini təmin edir. Server şəbəkənin işlənməsinə nəzarət, diaqnostika və informasiyanın

qorunması funksiyalarını həyata keçirir. Şəbəkədə bir və ya bir neçə server ola bilər. Məsələn, şəbəkədə fayl serverlə yanaşı kommunikasiya, verilənlər bazası serveri və s. ola bilər. Şəbəkəyə qoşulmuş digər kompüterlər klient və ya işçi yerlər adlanır. İşçi yerləri bu və ya digər xüsusiyyətlərinə görə vahid qrupda- işçi qrupda birləşirlər. Hər bir işçi qrup serverdə qeydiyyatdan keçir. Bu işə informasiya ehtiyatlarının axtarılmasını və qorunmasını asanlaşdırır.

Kombinə edilmiş şəbəkələr özündə bir rəngli və serverli şəbəkələrin ən yaxşı keyfiyyətlərini birləşdirir və geniş tətbiq dairəsinə malikdir. Adətən bu tip şəbəkələrdə bir neçə əməliyyatlar sistemi birgə istifadə olunur. Belə ki, serverlərdə Windows NT, Windows Server, işçi stansiyalarda Windows 9x, Windows NT Workstation, Windows XP, Windows Vista əməliyyatlar sistemləri istifadə oluna bilər.

Qlobal şəbəkələr çox geniş ərazidə (şəhər, ölkə və s.) fəaliyyət göstərən, bir-birilə qarşılıqlı əlaqəli, ayrı-ayrı kompüterlər və lokal şəbəkələr şəbəkəsindən ibarətdir. Qlobal şəbəkələrin ən bariz nümunəsi İNTERNET beynəlxalq informasiya şəbəkəsidir.

### **1.7.1 Şəbəkə komponentləri.**

#### **Kabel**

Şəbəkədə istifadə olunan kabelləri 3 böyük qrupa bölmək olar:

Koaksial (coaxial);

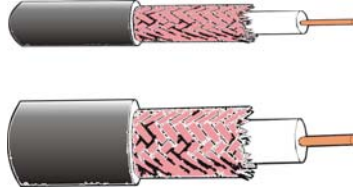
İki cütlük (twisted pair);

Optik lifli.

Koaksial kabellərdə siqnal polivinilxlorid və ya tefelon izolyasiyalı mis naqıl boyu ötürülür. Siqnalın zəifləməsinə səbəb olan elektromaqnit dalğaların təsirini aradan qaldırmaq

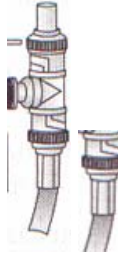
üçün ötürücü mis naqıl torşəkili qoruyucu ilə əhatə olunmuşdur. İki tip koaksial kabellər istehsal edilir (şək. 2).

- Nazik;
- Qalın.



Şək. 2.

Nazik koaksial kabellər 0,5 sm qalınlığa malik olub siqnalı 185m məsafəyə sönmədən, maneəsiz ötürməyə qadirdir. Elastik olduğu üçün montaj üçün rahatdır və praktiki olaraq bütün növ şəbəkələr üçün yararlıdır. BNC, BNCT konnektorları vasitəsilə şəbəkə platasına qoşulur (şək.3).



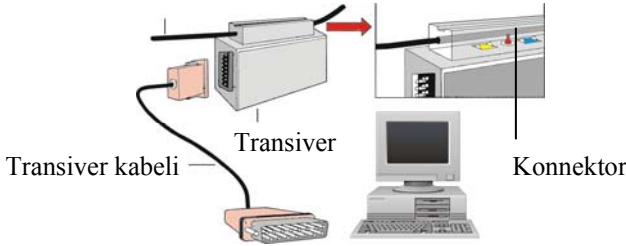
Şək.3.

İki kabel seqmentini birləşdirmək üçün borel-konnektorlardan istifadə olunur.

Qalın koaksial kabellər 1 sm diametrə malikdirlər və siqnalı daha uzaq məsafəyə – 500 m məsafəyə ötürə bilər. Ondan əsasən nazik koaksial kabellər əsasında yaradılmış bir neçə kiçik şəbəkələri birləşdirən əsas kabel - magistral kabel

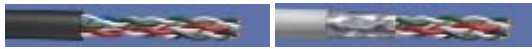


kimi istifadə olunur. Magistral kabel qoşulmaq üçün xüsusi transiverdən istifadə olunur (şək.4). Transiver iti «dişlərə» malik konnektorla təchiz olunmuşdur ki, məhz bu dişlər kabelin izolyasiyasına daxil olaraq ötürücü naqillə birbaşa əlaqəyə girir. Transiver şəbəkə kartının AUI portuna transiver kabeli vasitəsilə qoşulur.



Şək.4.

İki cütlük izolyasiyalı və bir-birinə dolaşmış mis naqillər cütündən ibarətdir. İki tip: ekranlaşmış (STP) və ekransız (UTP) kabellər istehsal olunur.



Şək. 5.

Ekran qoruyucu rolu oynayır və siqnalı nisbətən uzaq məsafəyə ötürməyə zəmin yaradır. Kabel şəbəkə kartına, kommutasiya panelinə və divar rozetkasına RJ-45 tipli konnektorlar (şək. 6) vasitəsilə qoşulur.



Şək.6.

Optik lifli kabellərdə informasiya optik lif boyunca modullaşmış işıq impulsu şəklində yayılır. Bu kabellər böyük həcmli informasiyanı çox böyük sürətlə və etibarlı surətdə ötürə bilər. Optik lif – nazik şüşə silindrdən ibarətdir. Qoruyucu şüşə örtüklə əhatə olunmuşdur. Hər bir optik lif informasiyanı yalnız bir istiqamətdə ötürməyə qadirdir. Ona görə optik-lifli kabellər xüsusi konnektorlarla təchiz olunmuş iki optik lifdən ibarət olur (şək. 7). İnformasiya ötürmə 100 Mbit/s - a qədər ola bilər.



Şək. 7.

### **Şəbəkə platası**

Şəbəkə platası kompüter və şəbəkə kabeli arasında interfeys rolunu oynayır və aşağıdakı funksiyaları yerinə yetirir:

- İnformasiyanın ötürülməsinə hazırlıq;
- İnformasiyanın ötürülməsi ;
- İnformasiya axınının idarə olunması.

Məlumdur ki informasiya kompüterdə ikilik say sistemində kodlaşmış - bitləşmiş formada saxlanılır və 16 və ya 32 tərtibli şin vasitəsilə paralel olaraq kompüterin daxili qurğularına, məsələn: mikroprosessoru, operativ yaddaşa və s. ötürülür. Kabel informasiyanı yalnız ardıcıl ötürmə qabiliyyətinə malik olduğundan informasiyanın ötürülməsinə hazırlıq mərhələsində plata ilk növbədə ötürüləcək informasiyanı paralel formadan ardıcıl formaya çevirir. Sonrakı mərhələdə plata informasiya qəbul edəcək kompüterin platası ilə "əlaqə" yaradaraq ona aşağıdakı məlumatları çatdırır:

- Ötürüləcək informasiya bloklarının maksimal ölçüsü
- İnnformasiyanın həcmi
- İnnformasiya bloklarının ötürmə intervalı
- Təsdiqedici interval
- Ötürmə sürəti.

Bundan sonra informasiya elektrik və ya optik siqnalına çevrilir və informasiya kabeli vasitəsilə şəbəkənin digər kompüterinə ötürülür.

Hər bir şəbəkə platası unikal ünvana və parametrlərə malik olur. Şəbəkədə hər bir plata bu haqda qarşılıqlı məlumata malik olur və parametrləri digər şəbəkə platasının parametrinə müvafiq uyğunlaşdırma qabiliyyətinə malikdirlər. Məsələn, şəbəkədə yüksək sürətli plata informasiya mübadilə sürətini digər plataların sürətinə uyğunlaşdırır.

İnnformasiya mübadiləsində şəbəkə platası mühüm rol oynadığından şəbəkənin fəaliyyəti, sürəti platanın seçilməsindən, düzgün quraşdırılmasından və ona uyğun program modullarından (məsələn drayverindən) çox asılıdır. Plata ilk növbədə kompüterin arxitekturasına və ona qoşulacaq kabelə uyğun olmalıdır. Qeyd edək ki, rabitə xəttlərsiz lokal şəbəkələrdə xüsusi növ şəbəkə platasından istifadə olunur. Bu platalar siqnal ötürücü və qəbuledici antenaya malik olurlar.

### **1.7.2. Şəbəkə topologiyası**

Topologiya termini ilə şəbəkənin strukturu - kompüter, server, periferiya qurğuları və digər şəbəkə komponentlərinin yerləşmə sxemi ifadə olunur. Şəbəkə qurğularının növlərindən və xarakteristikalarından, şəbəkənin genişlənmə imkanlarından, şəbəkənin idarəetmə üsulundan asılı olaraq şəbəkə əsasən 3 baza topologiyası:

- xətti (bus)
- ulduz (star)

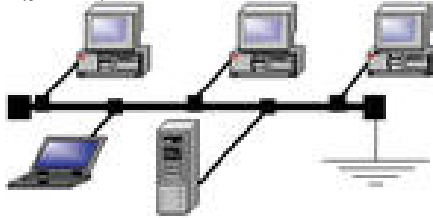
- halqa (ring)

və onların kombinasiyaları olan

- xətti-ulduz
- ulduz-halqavari

əsasında yaradılır.

**Xətti topologiya.** Bu topologiya sadə və ən geniş yayılmış topologiyadır. Şəbəkədə kompüterlər, serverlər, periferiya qurğuları magistral və ya seqment adlanan vahid kabelə ardıcıl olaraq qoşulur (şək. 8).



Şəkil 8.

Kabelin başlanğıc və sonunda siqnalın sönməsi üçün terminatorlar yerləşdirirlər.

Fayl şəklində saxlanılan informasiyanın həcmi  $4 \text{ K}^1$ -dən artıq olduqda informasiya ötürülməzdən əvvəl kompüter onu paket və ya kadr adlanan kiçik bloklara bölür. Hər paket aşağıdakı struktura malik olur:

- başlıq
- verilənlər
- treyler

Başlıq informasiya ötürücüsünün və alıcısının ünvanlarını, paketin ötürülməsi və vahid fayl şəklində birləşməsi haqda məlumatı daşıyır. Verilənlər ötürülən fayl fraqmentidir və həcmi  $4 \text{ kb}$ -dan çox olmur. Treyler özündə paketin ötürülməsinin korrekliyini yoxlayan və şəbəkə

komponentləri arasında əlaqə üsulundan asılı olaraq digər məlumatları daşıyır.

Paket elektrik siqnalı şəklində magistral boyu ötürülür. Bu zaman paketi yalnız ünvanı başlıqda daşdığı informasiya alıcısının ünvanına uyğun gələn kompüter qəbul edir. Qəbul edici kompüter başlıqdakı təlimatı uyğun olaraq paketləri vahid fayl şəklində birləşdirir və bu zaman treyler bu prosesin korrekliyini yoxlayır.

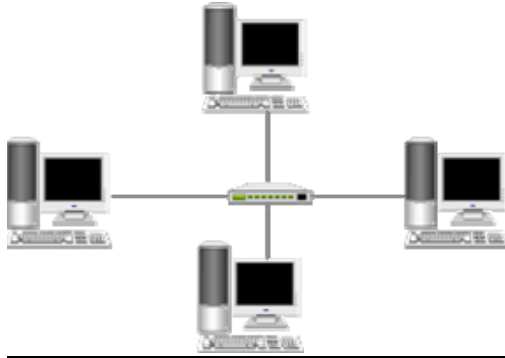
İki kompüter eyni zamanda informasiya ötürə bilərmə? Belə hadisəyə kolloziya hadisəsi deyilir (yəni paketlərin toqquşması). Bunun qarşısını almaq üçün kolloziyanı müəyyən etmək və kolloziyanı dəf etmək istifadə üsullarını tətbiq etmək olar. Kolloziyanı müəyyən etmək istifadə üsulunun mahiyyəti ondan ibarətdir ki, kompüter və ya server magistralın informasiya ötürülməsi üçün boş olduğunu müəyyən edir, sonra informasiyanı ötürür və bu zaman digər kompüterlərdən heç biri informasiya ötürə bilməz. Lakin, kabelin uzunluğu 2500 m-dən artıq olduqda magistralda siqnalın zəifləməsi nəticəsində kolloziya hadisəsini müəyyən etmək prosesi çətinləşir.

Digər üsulu tətbiq etdikdə hər bir kompüter informasiya ötürməzdən əvvəl bu haqda xəbərdarlıq siqnalı ötürür və digər kompüterlər bu siqnalları qəbul edərək informasiya ötürülməsini müvəqqəti olaraq təxirə salır. Bu üsulun tətbiqi şəbəkənin məhsuldarlığını (işləmə sürətini) azaldır.

Xətti şəbəkə passiv şəbəkədir. Belə ki, kompüterlərdən biri sıradan çıxarsa, şəbəkə işini davam etdirəcək. Şəbəkə yalnız magistral bu və ya digər səbəbdən xarab olarsa işləməyəcək. Şəbəkəni genişləndirmək zərurəti olduqda, magistralı uzatmaq üçün borel-konnektordan istifadə etmək olar. Lakin, bu ucuz başa gəlsə də, elektrik siqnalının sönməsi nəticəsində çoxsaylı borel-konnektorlardan istifadə heç də həmişə effektiv olmur. Şəbəkədə ayrı-ayrı kabel seqmentlərini birləşdirmək üçün repiterdən istifadə məqsədə uyğundur. Repiter konnek-

tordan fərqli olaraq siqnalı digər seqmentə ötürməzdən əvvəl siqnalı gücləndirir və sonradan ötürür.

**Ulduz şəbəkəsi.** Ulduz şəbəkəsində (şək. 9) bütün kompüterlər kabel seqmentləri vasitəsilə mərkəzi konsentratora (hub) qoşulur. Bu kabel sərfini artırır və mərkəzi konsentrator sıradan çıxdıqda şəbəkə öz fəaliyyətini dayandırır. Kompüterlərdən biri sıradan çıxarsa və ya kabel seqmenti aralanarsa yalnız həmin kompüter informasiya mübadiləsi etməyəcək.



Şək.9.

Konsentratorlar passiv və aktiv olur. Aktiv konsentrator repitor kimi siqnalı qəbul edib gücləndirir. Passiv konsentratorlar (məsələn montaj panelləri, kommutasiya blokları və s.) yalnız siqnalı qəbul edib ötürmək funksiyasını yerinə yetirirlər və onları aktiv konsentratorlardan fərqli olaraq qidalanma mənbəyinə qoşmağa ehtiyac yoxdur.

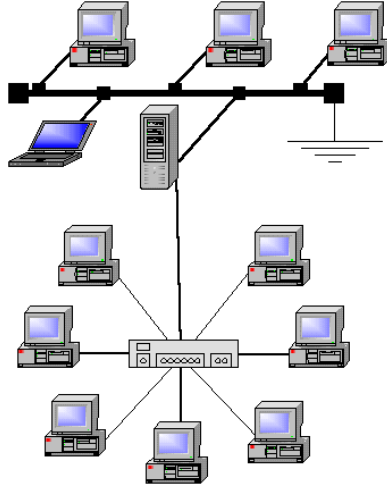
**Halqa.** Bu topologiyada kompüterlər qapalı halqa təşkil edən kabelə qoşulur. Xətti topologiyadan fərqli olaraq kabelin sərbəst ucları olmadığından terminatorlardan istifadəyə ehtiyac qalmır (şək.10).



Şək.10.

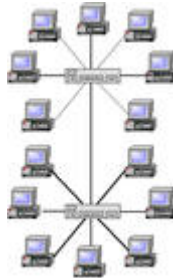
Şəbəkənin hər bir kompüteri repitor rolu oynayır, yəni siqnalı qəbul edir, gücləndirərək digər kompüterlərə ötürür. Buna görə də kompüterlərdən biri sıradan çıxarsa bütün şəbəkə öz fəaliyyətini dayandıracaqdır. İnformasiya aşağıdakı prinsiplə ötürülür: şəbəkə fəaliyyətə başlayan kimi marker yaranır və ardıcıl olaraq halqa boyu bir kompüterdən digər kompüterə ötürülür. İnformasiya ötürməyə hazırlaşan kompüter markeri qəbul edib, onu informasiya sahibinin ünvanı ilə, verilənlərlə tamamlayır. Markeri qəbul edici kompüter qəbul edib, verici kompüterə təsdiqedic məlumat göndərir. Bundan sonra informasiya göndərən kompüter yeni marker yaradaraq şəbəkəyə ötürür. Markerin ötürülməsi praktiki olaraq işıq sürətilə yayılır. Belə ki, 200 m diametrlı halqada marker 10000 dövr/san tezliyi ilə dövr edir.

**Ulduz-xətti.** Bu topologiya «ulduz» və «xətti» topologiyaların kombinasiyasıdır (şək. 11). Adətən bir neçə «ulduz» topologiyalı şəbəkəni magistral kabel vasitəsilə xətti şəkildə birləşdirirlər. Şəbəkənin kompüterlərindən hər hansı birinin xarab olması bütövlükdə şəbəkənin işləməməsinə səbəb olmayacaq. Konsentratordan biri və ya bir neçəsi sıradan çıxdıqda isə şəbəkənin yalnız müəyyən hissəsi- həmin konsentratora qoşulmuş kompüterlər və konsentratordakı kompüterlər öz fəaliyyətini dayandıracaqdır.



Şək.11.

**Ulduz-uldaz şəbəkələr.** Ulduz şəbəkələri konsentratorlar vasitəsilə birləşərək ulduz-uldaz topologiyası yaradırlar (şək.12).



Şək.13.

### 1.7.3 Şəbəkənin məntiqi arxitekturası

Protokollar informasiyanın necə paketlənəcəyini, istifadə ediləcəyini və şəbəkə üzərindən necə göndəriləcəyini müəyyən edir və onun təhlükəsizliyini təmin edir. Yəni protokollar informasiyanın ötürülməsi qaydalarıdır. Şəbəkədəki



kompüterlər, serverlər bu qaydalar çərçivəsində hərəkət edərək informasiyanı ötürürlər.

Məntiqi arxitekturaya misal olaraq ISO tərəfindən hazırlanmış 7 təbəqəli OSI (Open System InterConnect) modelini göstərmək olar. OSI modeli şəbəkənin bütün funksiyalarını bir araya cəmləyərək onları təbəqələrdə qruplaşdırır.

OSI təbəqələri və onların təyinatları aşağıdakı kimidir:

1. Fiziki təbəqə - elektrik və mexanik xüsusiyyətləri müəyyən edir.

2. İnformasiya Əlaqəsi - informasiyanın bir nöqtədən digərinə göndərilməsi və çərçivələrə bölünməsinə müəyyən edir.

3. Şəbəkə təbəqəsi - informasiya paketinin təyin olunmuş son nöqtəyə çatdırılmasını təmin edir. Bu məqsədlə düyünlərə “şəbəkə ünvanı” adlanan nömrələr verilir. Bundan sonra isə istifadə olunan protokol sayəsində informasiya təyin olunmuş ünvana çatdırılır.

4. Nəqliyyat təbəqəsi - bütün informasiyanın etibarlı şəkildə çatdırılması və yaranmış xətalara aşkar edilməsini təmin edir.

5. Sessiya təbəqəsi - şəbəkədəki iki istifadəçi arasındakı əlaqənin yaradılması, idarə olunmasını təmin edir.

6. Təqdimat təbəqəsi - tərcümə, kodlama və dekodlama, informasiyanın sıxılmasını və açılmasını, mesajı göndərən müəyyən edilməsi və təsdiqlənməsini təmin edir.

7. Tətbiq təbəqəsi - Şəbəkə resurslarına müraciəti təmin edir. Bu səviyyə üçün faylların göndərilməsi, elektron məktublaşma, şəbəkənin idarə olunması, terminal protokolları kimi sistemlər inkişaf etdirilmişdir.

Hər bir OSI təbəqəsi özündən əvvəlki təbəqələri dəstəkləyir.

### 1.7.4 Naqilsiz şəbəkələr

Adından məlum olduğu kimi bu tip şəbəkələrdə ənənəvi şəbəkələrdən fərqli olaraq rabitə vasitələri olaraq kabel və telefon xəttlərindən istifadə olunmur. Onlar üç böyük qrupa bölünür:

1. Lokal şəbəkələr
2. Genişləndirilmiş şəbəkələr
3. Mobil şəbəkələr

Lokal şəbəkələrdə informasiya 4 üsulla:

- infraqırmızı şüalanma
- lazer şüalanma
- kiçik spektrli radorabitə
- yayılmış spektrli radorabitə

vasitəsilə ötürülür.

İnfraqırmızı şüalanma geniş spektrli tezliyə malik olduğundan informasiya ötürmə sürəti 10 Mbit/s qədər ola bilər. Aşağıdakı 4 növ infraqırmızı şüalanma ilə fəaliyyət göstərən şəbəkələr geniş yayılmışdır:

1. Birbaşa görünüş şəbəkələri. Bu şəbəkələr adından məlum olduğu kimi, qəbul edici və ötürücü qurğular arasında birbaşa görünüşün olmasını tələb edir.

2. Paylanmış infraqırmızı şüalanma şəbəkələri. Bu tip şəbəkələrdə siqnal divar, döşəmə və tavandan əks olunaraq qəbulediciyə çatır. İnformasiyanın yayılma sürəti və effektiv yayılma sahəsi 30m<sup>2</sup>-lə məhdudlaşır.

3. Əks olunan infraqırmızı şüalanma. Bu tip şəbəkələrdə optik transiver siqnalı müəyyən sahəyə göndərir və siqnal oradan tələb olunan kompüterə ünvanlanır.

4. Geniş sahəli optik şəbəkələr. Bu şəbəkələr geniş imkanlara malikdir və kabelli şəbəkələrdən heç də geridə qalmırlar.

Lazer şüalanması əsasında yaradılan şəbəkələr birbaşa görünüşün olmasını tələb edir.

Geniş spektrli radorabitədə istifadəçi vericini və qəbuledicini müəyyən tezliyə kökləyir. Bu zaman 46500 m<sup>2</sup> sahəyə informasiya mübadiləsi etmək mümkün olur. Rabitə sürəti təqribən 4,8 M bit/s olur.

Yayılmış spektrli radorabitədə müəyyən tezliklər intervalında informasiya mübadiləsi baş verir. Qəbuledici adapter kompüterini sinxron olaraq bir tezlikdən digər tezliyə kökləyir. Sürət 250 K bit/s artıq olmur. Son vaxtlar açıq 3,2 km sahədə və qapalı 129 m sahədə 2 M bit/s sürətlə informasiya mübadiləsi yaratmaq mümkün olmuşdur.

Genişləndirilmiş şəbəkələrdə ayrı-ayrı binalarda yerləşən lokal şəbəkələr arasında informasiya mübadiləsi rabitə xətti olmayan körpülər vasitəsilə həyata keçirilir. Məsələn, AIRLAN/Brige Plus körpüsü aralarında məsafə 5 km olan iki şəbəkə arasında rabitəni təmin edir.

Mobil şəbəkələr mobil rabitə vasitələri əsasında fəaliyyət göstərir.

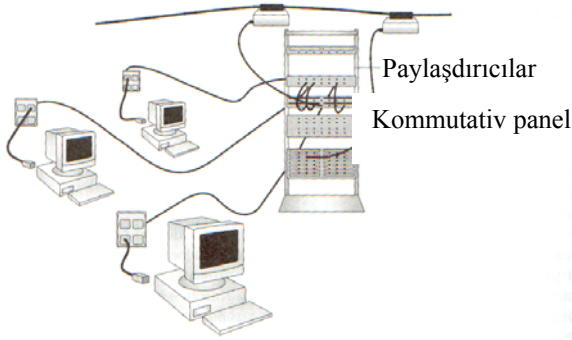
## 1.7.5 Şəbəkə arxitekturaları

### Ethernet

Ethernet-arxitekturalı ilk şəbəkə 1975-ci ildə XEROX firmasının əməkdaşları Robert Metkolf və Devid Boqqis tərəfindən yaradılmışdır. O, uzunluğu 1 km olan kabel vasitəsilə 100 kompüter arasında informasiya mübadiləsini 2,94 Mbit/s sürətilə təmin edirdi. Sonralar təkmilləşərək, hal-hazırda 10-100 Mbit/s sürətilə informasiya mübadiləsi etməyə imkan verən ən geniş yayılmış şəbəkə arxitekturasıdır. Bir neçə tip Ethernet şəbəkəsini nəzərdən keçirək.

#### 10 BASE T

10 BASE T şəbəkə koaksial kabel əsasında, ulduz-xətti topologiya bazasında yaradılmışdır (şək. 14).



Şək. 14.

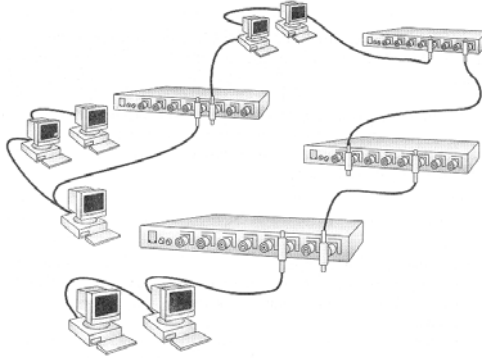
Kabel şəbəkə platasına RJ-45 konnektoru vasitəsilə qoşulur. Kabel seqmentinin maksimal uzunluğu –100 m, minimal uzunluğu isə 2,5 metrdir. Kabel seqmentlərini çox-portlu repitor rolunu oynayan aktiv konsentrator vasitəsilə birləşdirmək olar. Şəbəkədə kompüterlərin sayı 1024 ədəddən artıq olmaz. Magistral kabel olaraq yoğun koaksial kabel və ya optik lifli kabel istifadə olunur. Maksimal informasiya ötürmə sürəti 10Mbit/s-dir.

### 10 BASE 2

10 BASE 2 tip şəbəkə<sup>1</sup> nazik koaksial kabel əsasında, xətti topologiya bazasında fəaliyyət göstərir. Kabel seqmentinin maksimal uzunluğu 185 m, minimal uzunluğu 2m olub 30 ədəd kompüterin qoşulması üçün nəzərdə tutulmuşdur. Kabel şəbəkə adapterinə BNC-T konnektoru vasitəsilə qoşulur. Kompüterlər arasında məsafə 0,5 metrdən az olmamalıdır. Şəbəkə 4 ədəd repitorlar vasitəsilə 925 metrə qədər genişlənmə

<sup>1</sup> Ethernet 10 Base2 tip şəbəkə “nazik” Ethernet kimi adlandırılır. Ən ucuz və quraşdırılması asan olan lokal şəbəkədir.

bilər (şək.15). Kompüterlərin maksimal sayı 1024 olmalıdır. Maksimal informasiya ötürmə sürəti 10Mbit/s-dir.



Şək. 15.

### 10 BASE 5

10 BASE 5 şəbəkə yoğun koaksial kabel əsasında xətti topologiya bazasında<sup>1</sup> yaradılmışdır. Kabel seqmentinin maksimal uzunluğu 500 metr olub 100 ədəd kompüter qoşula bilər. Şəbəkə 4 ədəd repitor vasitəsilə 2460 metrə qədər genişləne bilər. Kabel şəbəkə kartına qoşulmaq üçün transiverdən (MAU) istifadə edilir. Transiver koaksial kabel üzərində yerləşdirilir. Transiverdə aktiv qəbuledici və ötürücü vardır. Transiverlə şəbəkə kabelini birləşdirən kabelin maksimal uzunluğu 50 m-dir.

### 10 BASE-T

Ethernet 100 Base T4-də burulmuş cütlik kabel əsasında fəaliyyət göstərir. Şəbəkədə kompüterlər

<sup>1</sup> Ethernet 10 Base5 “Qalın” Ethernet adlandırılır.

konsentratora–xaba (hub) qoşulur və kompüterlərarası informasiya mübadiləsi xab vasitəsi ilə həyata keçirilir. Bu tip şəbəkə aşağıdakı xarakteristikalara malikdirlər:

Topologiyası	Ulduz
Seqmentin maksimal uzunluğu	100 m
Kompüterlərarası məsafə	100 m qədər
Kabelin tipi	Koaksial ikilik cütlük UTP 5
Seqmentdə kompüterlərin maksimal sayı	Xabın girişlərinin sayı qədər

### 10 BASE FL

10 BASE FL tip şəbəkə optik-lifli kabel əsasında yaradılır və repitorlar arasında 200 metr məsafənin olmasına imkan verir.

Topologiyası	Nöqtə–nöqtə
Seqmentin maksimal uzunluğu	> 1000 m
Kompüterlərarası məsafə	1000 m və ondan böyük
Kabelin tipi	Optik
Seqmentdə kompüterlərin maksimal sayı	2

### Token Ring

Bu arxitektura ilk dəfə 1954-cü ildə stolüstü kompüter, mini EHM və meynfreymləri vahid şəbəkəyə qoşmaq üçün yaradılmışdır. Token Ring lokal şəbəkəsində kompüterlər halqa topologiyası vasitəsilə birləşdirilir. Token Ring lokal şəbəkəsində informasiyanı ötürmək üçün markerdən (token) istifadə edilir. Marker alan kompüter kanalı tutmuş hesab olunur, yəni öz informasiyasını göndərə bilər. İnförmasiya paketlər şəklində göndərilir. Marker alan kompüter paketini digər kompüterə ötürür. Paket öz ünvanına çatdıqda alıcı kompüter paketi özünə yazır, bu haqda paketdə lazımi qeydlər edir və paketi qonşu kompüterə ötürür. Paket yenidən onu

göndərən kömpüterə gəldikdə, həmin paketi halqadan geri alır və yeni paket varsa onu göndərir. Göndəriləcək yeni paket yoxdursa markeri qonşu kompüterə göndərir və proses təkrar olunur. Token Ring şəbəkəsi aşağıdakı xarakteristikalara malikdirlər:

Baza topologiyası	Ulduz-halqa
Kabel sistemi	Ekranlaşmış və ya ekranlaşmamış ikili cütlük
Kabel seqmentinin maksimal uzunluğu	Kabelin tipindən asılı olaraq 45-200 m
Kabel seqmentlərinin birləşdiricilərinin maksimal sayı	33 ədəd
Kompüterlərin maksimal sayı	Ekranlaşmış ikili cütlük üçün 260 ədəd Ekranlaşmamış ikili cütlük üçün 72 ədəd
Kabel seqmentinin minimal uzunluğu	2,5 m

# FƏSİL 2

## Turbo Pascal alqoritmik dili

- ◆ Dilin əsas elementləri
- ◆ Proqramın strukturu
- ◆ Verilənlərin tipləri. Sadə və sətir tiplər
- ◆ Operatorlar
- ◆ Giriş və çıxış prosedurları
- ◆ Şərti keçid operatoru
- ◆ Seçim və ya variant operatoru
- ◆ Şərtsiz keçid operatoru
- ◆ Dövr operatorları
- ◆ Çoxluqlar
- ◆ Massivlər
- ◆ Yazılar
- ◆ Prosedurlar
- ◆ Funksiyalar
- ◆ Prosedur tipli dəyişənlər
- ◆ Rekursiyalar
- ◆ Fayl tipli dəyişənlər. Fayllarla əməliyyatlar
- ◆ Modullar
- ◆ Standart prosedur və funksiyalar
- ◆ Turbo Pscal dilinin qtafiki imkanları
- ◆ Göstəricilər



## 2.1. Alqoritmik dillər

Kompüter istehsalının ilk dövründə proqramlar maşın dilində yazılırdı. Maşın dili kompüterin “başa düşdüyü” kodlarla ifadə olunmuş əmrlərdən ibarət olub, konkret kompüterin arxitekturasından asılı idi. Hər bir əmrdə ümumi şəkildə, aparılacaq əməliyyatın məzmunu haqqında məlumat, üzərində maşın əməliyyatı aparılacaq başlanğıc verilənlərin yerləşdiyi yer-ünvan, nəticənin ünvanı və bu əmrdən sonra yerinə yetiriləcək əmr haqqında məlumat verilirdi. Maşın dilində proqramların yaradılması və onların kompüterdə yerinə yetirilməsi kifayət qədər mürəkkəb və vaxt aparan idi. Bu səbəbdən kompüter texnologiyasının inkişafının sonrakı mərhələsində təbii dilə daha yaxın olan simvolik dillər yarandı. Belə ki, ikinci nəsil kompüterlərin yaranması, konkret maşının yox, qoyulmuş məsələnin xüsusiyyətlərindən asılı olan dillərə ehtiyac yaratdı. Bu dillərə formal dillər və ya sadəcə alqoritmik dillər deyilir və bir sıra üstünlüklərə malikdir. Bu dillər əyani olub, onlarla ixtiyari alqoritmik asanlıqla ifadə etmək mümkündür. Alqoritmik dillər alqoritmik birqiymətliyini, mürəkkəb alqoritmik daha sadə alqoritmiklərin vəhdəti şəklində ifadə edilməsini təmin edir. İlk mükəmməl alqoritmik dil 1954-cü ildə İBM firmasında Con Bekusun rəhbərlik etdiyi qrupun yaratdığı FORTRAN dili idi. Bu dilin adı FORTRAN-FORmulae TRANslation - formulaların tərcüməsi sözümdən götürülmüşdür. Bu dil çox sadə struktura malik olduğundan ondan hal-hazırkı vaxta qədər istifadə olunur. Fortrandə proqram operatorları ardıcılığı şəklində yazılır. Bu dildə yazılan proqram bir və ya bir neçə seqmentlərdən (alt proqram) ibarət olur. Bütün proqramın işini idarə edən seqment əsas proqram adlanır.

Fortran dili elmi və mühəndis texniki hesablama sahələrində istifadə edilmək üçün nəzərdə tutulmuşdu. Lakin bu dildə budaqlanan strukturlu məsələlər (istehsal prosesinin modelləşdirilməsi və s.), bəzi iqtisadi məsələlər və redaktətmə məsələləri (cədvəl, arayış və s. qurulması) üçün proqramlar da qurula bilər. Sonrakı illərdə bu dilin müxtəlif modifikasiyaları yaradılmışdı.

1960-cı ildə Alqol-60 (Alqoritmik Language-alqoritmik dil) dili, 1966-cı ildə isə Fortran dili əsasında Dartmut kollecinin hesablama mərkəzində Basic dili (BASIC-Beginner's Allpurpose Symbolic Instruction Code-yeni başlayanlar üçün çoxməqsədli simvolik əmrlər dili) yaradıldı. Basic dili Visual Basic.Net versiyasına qədər böyük təkamül yolu keçmişdir. Hal-hazırda Visual Basic.Net dilindən qrafiki interfeysli proqram əlavələrin yaradılmasında geniş istifadə olunur.

Fortran, Alqol-60 dillərilə paralel olaraq intensiv inkişaf edən elm və texnikanın yeni sahələrinin tələbatını ödəmək üçün yeni proqramlaşdırma dilləri yaradılmışdır. Məsələn, 1957-ci ildə riyazi verilənlərin emalı üçün APL (Application Programming Language) dili, 1959-cü ildə İBM firması tərəfindən böyük həcmli verilənlər massivinin emalı üçün Cobol (Common Business Oriented Language)

dili, mətn informasiyanın emalı üçün 1962-ci ildə Snobol<sup>1</sup>, 1969-ci ildə çoxluqlar üzərində əməliyyatlar aparmaq üçün SETL dili yaradılır.

Üçüncü nəsillə kompüterlərin yaranması, universal alqoritmik dillərin yaradılması məsələsini qarşıya qoydu. Bu cür dillərin yaradılması üçün edilən cəhdlərdən biri nəticəsində İBM firması tərəfindən PL/1 (Programming-Language/1-proqramlaşdırma dili-1) dili yaradılır. Bu dil Fortran, Alqol və Cobol dillərinin əsasında yaradılmış və bu dillərin üstünlüklərini özündə birləşdirmişdi.

1968-1970-ci illərdə İsveçrə ali politexnik məktəbinin informatika institutunun direktoru, Türinq mükafatı laureatı, professor Niklaus Virt tərəfindən Nikuls Virt tərəfindən Paskal<sup>2</sup> dili yaradılır. Bu dil struktur proqramlaşdırma ideyasının, yəni proqramın kiçik, dəqiq təyin edilmiş prosedurlardan tədricən qurulması ideyasının həyata keçirilməsini təmin edən ilk dildir. Bütün alqoritmik dillərdə olduğu kimi, Pascal dilinin də hazırda çox versiyaları var. Turbo-Pascal alqoritmik dili-bu versiyalardan biridir və fərdi kompyuterlərdə çox geniş tətbiq olunur.

1983-cü ildə Pentaqonda Ada<sup>3</sup> dili yaradılır. Bu dil əsasən hərbi və böyük sənaye layihələrinin yaradılmasında istifadə edilir. Dil birinci proqramlaşdırıcı – qrafına Ada Lavlaysın şərəfinə adlandırılmışdır. 1972 -ci ildə Kreniqan və Ritçi tərəfindən C dili, 1986-cı ildə onun bazasında Brayn Straustrup tərəfindən problem yönümlü C<sup>++</sup> dili yaradılır. C/C<sup>++</sup> dili-universal dil olub, sistem proqramlarının yaradılmasında, o cümlədən geniş yayılmış UNIX əməliyyat sistemi üçün kodların yazılmasında geniş istifadə olunur. 1995-ci ildə Sun Microsystems kompaniyasında Java dili yaradılır. Qeyd etdiyimiz dillərdən başqa dillər də mövcuddur və bu dillərin yaradılması prosesi davam etdirilir.

XX əsrin 90-cı illərindən başlayaraq Web proqramlaşdırma dilləri meydana gəldi. Web proqramlaşdırma dilləri Web səhifələrin idarə olunmasında istifadə olunur. Onlara misal olaraq HTML, XML, JavaScript, VbasicScript, Perl, Payton göstərmək olar.

## 2.2. 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;

<sup>1</sup> Onun müasir sələfi 1974-cü ildə yaradılmış Icon dilidir.

<sup>2</sup> Bu dil XVII əsr böyük fransız alimi Paskalın şərəfinə adlandırılmışdır.

<sup>3</sup> 1995-ci ildə dilin yeni versiyası ADA -95 yaradılmışdır.

3) Xüsusi simvollar və məhdudlaşdırıcılar:

+ - \* / = < > > < <=> = := ( ) { } [ ] ^ # \$ ( \*\* ) , . ; ' ,

4) Cüt simvollar<sup>1</sup>: <> <= > = := (\*;\*);

5) Probel və idarəedici simvollar.

Turbo Pascal dilində probel ayırıcı funksiyasını yerinə yetirir. Ardıcıl yazılmış bir neçə probel işarəsini kompilyator bir probel kimi qəbul edir. İdarəedici 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 simvollarından, 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
break	case	class	const
constructor	continue	destructor	div
do	downto	else	end
exit	external	externalsync	file
finalization	for	forward	function
if	in	inherited	initialization
is	mod	not	of
or	private	procedure	program
property	protected	public	shl
record	repeat	set	unit
shr	sizeof	string	while
then	to	type	with
until	uses	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<sup>2</sup>.

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 çoxluğunu 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 proqramlaşdırıcı tərəfindən verilə bilər. Əks halda kompilyator sabitin tipini onun qiymətinə əsasən müəyyən edə bilər.

<sup>1</sup> Cüt simvollar arasında probel işarəsinin qoyulması yolverilməzdir.

<sup>2</sup> 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, simvollar, sətirlər, çoxluq konstruktorları<sup>1</sup> və qeyri-müəyyən göstərici əlaməti olan NIL<sup>2</sup> istifadə edilə bilər. Tam ədədlər adi qayda ilə işarə və ya işarəsiz yazılır və -2147483648-dən +2147483647-dək qiymətlər ala bilər. Həqiqi ədədlər işarə və ya işarəsiz, adi və ya eksponensial formada yazılır. Kəsr hissə onluq nöqtədən sonra yazılır. Eksponensial hissə e (E) simvolu ilə başlayır, ondan sonra «+» və ya «-» işarəsi gələ bilər və onluq tərtib verilir. Məsələn, 0.0015 və 1.5E-3 (yəni  $1,5 \cdot 10^5$ ) yazılışları ekvivalentdir. Onaltılıq ədədlər, qarşısında \$ işarəsi olan onaltılıq say sisteminin ədədləridir. Onların dəyişmə diapazonu \$00000000-dan \$FFFFFFF-ə qədərdir. Simvollar apastrof işarələri arasında yazılır. Məsələn, 'D', 'R', '12', və s. Sətirlər isə apastrof işarələri arasına alınan ixtiyari simvollar ardıcılığıdır<sup>3</sup>. Məsələn, 'Azad Qurbanov'. Sətirdə heç bir simvol verilməzsə, belə sətir boş sətir adlanır. Çoxluq konstruktoru - kvadrat mötərizə daxilində verilən çoxluq elementlərinin siyahısıdır. Məsələn, [1,2,3. .8,12], [Aysel. Turqut], [ ], [İnformatika] və s.

**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 ardıcılığıdır 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	Ptr
BlockRead	Frac	Random
BlockWrite	GetMem	Randomize
Boolean	GotoXY	Read
Buflen	HeapPtr	Readin
Byte	Iti	Real
Chain	IOResult	Rename
Char	Input	Reset
Chr	InsLine	Rewrite
Close	Insert	Round
ClrEol	Int	Seek

<sup>1</sup> Bax: Çoxluqlar

<sup>2</sup> Bax: Göstəricilər.

<sup>3</sup> Sətir uzunluğu 255 simvoldan çox olmamalıdır. Ən qısa sətir heç bir simvola malik olmayan sətirdir boş və ya sıfır uzunluqlu sətir adlanır.

ClrScr	Integer	Sin
Con	Kbd	SizeOf
ConInPtr	KeyPressed	Sqr
ConOutPtr	Length	Sqrt
Concat	Ln	Str
Copy	Lo	Succ
ConstPtr	LowVideo	Swap
Cony	Lst	Text
Cos	LstOutPtr	Trm
CtrlExit	Mark	True
CtrlInit	MaxInt	Trunc
DelLine	Mem	UpCase
Delay	MemAvail	Usr
Delete	Move	UsrInPtr
Eof	New	UsrOutPtr
Eoln	NormVideo	Val
Erase	Odd	Write
Execute	Ord	Writeln
Exp	Output	

İşçi sözlər və standart adların mənası və tətbiq qaydası ilə növbəti paragraflarda tanış olacağıq.

**İfadələr.** İfadələr operandlar – dəyişən və sabitlər, dairəvi mütərizələr, funksiyalar<sup>1</sup> və əməl operatorları<sup>2</sup> vasitəsilə tərtib olunur və verilənlərin üzərində hesab, məntiqi əməllərin yerinə yetirilməsini təyin edir.

### 2.3. Proqramın strukturu

Turbo Pascal dilində yazılmış proqram struktur proqramlaşdırmanın prinsiplərinə əsaslanmalı, dilin semantikasını və sintaksisinə uyğun olmalıdır. Proqramın əsas keyfiyyət göstəriciləri nəticənin dəqiqliyiylə, yerinə yetirilmə vaxtının, tələb olunan yaddaşın həcmiminin minimum olması ilə xarakterizə olunur.

Turbo Pascal dilində yazılmış proqram, maksimal uzunluğu 127 simvoldan az olan sətirlərdən ibarətdir. Sətirlər istənilən mövqedən başlaya bilər və bu zaman istifadə olunan operatorların sayı üzərinə məhdudiyyət qoyulmur. Ümumiyyətlə, proqramın yazılışı üzərinə qoyulan əsas tələb onun oxunaqlı olmasıdır.

Turbo Pascal dilində yazılmış proqram standart struktura malik olub, təsvir, prosedur və funksiya, icraedici blokdən ibarətdir. Proqramda təsvir, prosedur və funksiya blokları proqramın təsvir etdiyi alqoritmindən asılı olaraq iştirak edə bilər. Lakin, proqramda icraedici blokun olması zəruridir. Əks halda proqramın mənası olmaz.

<sup>1</sup> Bax: Funksiyalar

<sup>2</sup> Bax: Operatorlar

Təsvir bloku öz növbəsində zəruri olmayan başlıqdan, kompilyatorun global direktivlərinin, modulların, nişanların, sabitlərin, istifadəçi tiplərinin və dəyişənlərin elan bölmələrindən ibarətdir. Qeyd edək ki, standart Pascal-dan fərqli olaraq Turbo Pascalda elan olunma ixtiyari ardıcılıqla həyata keçirilə bilər.

Proqram başlığı hər bir proqramın əvvəlində yazılır. Başlıq **Program** işçi sözündən, proqramın adından ibarətdir. Məsələn,

```
Program First;
```

Proqramda başlığın verilməsi zəruri deyildir. Kompilyatorun iş rejimlərini idarə etmək üçün proqram başlığından sonra global direktivlər yazılır. Direktivlər { və } mötərizələri daxilində yazılır. Direktivlərin qarşısında, fiqurlu mötərizə daxilində \$ simvolu yazılır. Beləliklə, {\$...} yazılışı həmişə kompilyatorun direktivini müəyyən edir. Kompilyatorun direktivi məsələn, sətir tipli dəyişənlərin interpretasiya üsulu ({\$H+}) ilə verilə bilər, ehtiyatlar fayllarının qoşulması ( {\$R\*.DFM}) və s. yerinə yetirilə bilər.

Proqrama digər modullar – istifadəçi və standart kitabxana modulu<sup>1</sup> qoşmaq tələb olunarsa, onların adı təsvirlər blokunda elan olunmalıdır. Bunun üçün **Uses** işçi sözündən sonra modulların adının siyahısı yazılmalıdır. Yalnız sistem modulları istisna təşkil edir. Çünki, bu modullar əvvəlcədən elan edilmiş hesab edilir. Məsələn, **Uses Graph;** yazılışı proqrama **Graph** modulunun qoşulmasını təmin edir. Bununla **Graph** modulunda müxtəlif hündəsi fiqurların çəkilməsi üçün nəzərdə tutulmuş 50 prosedur və funksiyadan müraciət etmək mümkündür.

Turbo Pascal dilində hər bir operatorndan əvvəl nişan qoymaq olar. Nişan identifikatordan və ondan sonra gələn iki nöqtədən ibarətdir və proqramın istənilən yerindən həmin nişandan istifadə etməklə onun aid olduğu operatora müraciət etmək olar. Proqramda istifadə olunan bütün nişanlar təsvir blokunda elan olunmalıdır. Təsvir blokunda nişanların elan olunması **Label** işçi sözü ilə başlayır və bu sözdən sonra nişanlar aralarında vergül qoyulmaqla sadalanılır:

```
Label < nişan_1, nişan_2..., nişan_n>;
```

Məsələn, proqramda 1, 2, 3 nişanlarından istifadə etmək üçün onlar mütləq aşağıdakı kimi elan olunmalıdır.

```
Label 1, 2,3 ;
```

Təsvir bölməsində həmçinin adlı sabitlər elan olunmalıdır. Turbo Pascal bu məqsədlə iki cür yazılış forması təklif edir:

```
Const <identifikator>= <qiymət>;
```

və ya

```
Const <identifikator>:<tip>= <qiymət>;
```

Birinci yazılış formasında sabitin tipi qeyri-aşkar, ikinci yazılış formasında isə aşkar elan olunur. Məsələn,

```
const n = -10;  
      m = 1000000000;  
      x = 2.5;
```

---

<sup>1</sup> Bax: Modullar.

```
c = 'z';
b = true;
```

və ya

```
const n: integer = -10;
      x: real = 2.5;
      c: char = 'z';
      b: boolean = true;
```

Sabit müəyyən olunduqdan sonra onun adına digər qiymət mənimsətmək olmaz.

Təsvir bölməsində həmçinin proqramlaşdırıcı tərəfindən müəyyən olunan tiplər<sup>1</sup> və dəyişənlər elan olunur. İstifadəçi tipi aşağıdakı yazılış formatı əsasında elan edilməlidir:

```
Type <adı>= <tipin qiyməti>;
```

Məsələn, aşağıdakı kimi 10 elementli vektor tipi müəyyən etmək olar:

```
Type
      vektor = array [1..10] of integer;
```

Proqramda istifadə edilən dəyişənlər isə aşağıdakı sintaksis əsasında təsvir blokunda elan edilir.

```
Var <identifikator_1,... identifikator_n> : <tip>;
```

Məsələn,

```
var
      a,b,c: integer;
      d: real;
      s,s1: string;
      ch: char;
```

yazılışı a,b,c - tam tipli dəyişənin, d - həqiqi tipli dəyişənin, s,s1- sətir tipli, ch - simvol tipli dəyişənin elan olunmasını bildirir.

Prosedur və funksiya blokunda altproqramlar yazılır. Altproqram cari proqram üçün unikal ada malik proqram vahididir. Altproqrama əsas proqramdan və digər altproqramdan adı ilə müraciət etmək olar. Turbo Pascal dilində altproqram rolunu istifadəçi tərəfindən yazılan prosedur və funksiyalar oynayır<sup>2</sup>. Altproqramı elan etmək üçün müvafiq olaraq **Procedure** və **Function** işçi sözlərindən istifadə olunur. Prosedur və funksiya müvafiq olaraq aşağıdakı yazılış strukturuna malikdir:

```
Procedure <adı> (<parametrlər>);
      <Təsvir etmə bölməsi>
begin
      <operatorlar bölməsi>
End;
```

və ya

```
Function <adı> (<parametrlər>): <tip>;
      <Təsvir etmə bölməsi>
begin
      <operatorlar bölməsi>
End;
```

<sup>1</sup>Verilənlərin tipləri haqqında növbəti paragraflarda ətraflı məlumat veriləcəkdir.

<sup>2</sup> Prosedur və funksiyalar haqqında ətraflı məlumat müvafiq olaraq «Prosedurlar, Funksiyalar» paragrafında veriləcəkdir.

Təsvir etmə bölməsində altproqramın operatorlar bölməsində istifadə olunan lokal nişanlar, sabitlər, dəyişənlər elan olunur.

Məsələn, kvadrat tənliyin həll alqoritmi aşağıdakı **Equation** proseduru vasitəsilə,  $\sum_{i=1}^k \frac{1}{i}$  sırasının ilk k həddinin cəminin hesablanması alqoritmi **sum** funksiyası vasitəsilə verilə bilər.

```

procedure Equation (a,b,c:real; var x1,x2:real);
var d:real;
  begin
    d:=sqr(b)-4*a*c;
    if d>=0 then begin
      x1:=(-b+sqr(d))/(2*a);
      x2:=(-b-sqr(d))/(2*a);
    end;
  end;
function sum (k:integer):real;
var i:integer; s:real;
begin
  s:=0;
  for i:=1 to k do s:=s+1/i;
  sum:=s;
end;

```

Proqramın icraedici bloku **Begin** işçi sözü ilə başlayır, sonra isə bir-birindən nöqtə-vergüllə ayrılan dilin operatorları yazılır. Bölmənin sonu **End** işçi sözü ilə qurtarır. **End** işçi sozündən sonra mütləq nöqtə qoyulmalıdır:

```

Begin
  <operator;>
  ...
  <operator;>
End.

```

Məsələn, kvadratın sahəsinin tapılması alqoritmi aşağıdakı kimi yazıla bilər.

```

Program P_2;
  Var a, s : Real;
  Begin
    Write('kvadratin terefini daxil edin');
    Readln(a);
    s := a * a;
    Write('Kvadratin sahəsi=', s:10:4);
  End.

```

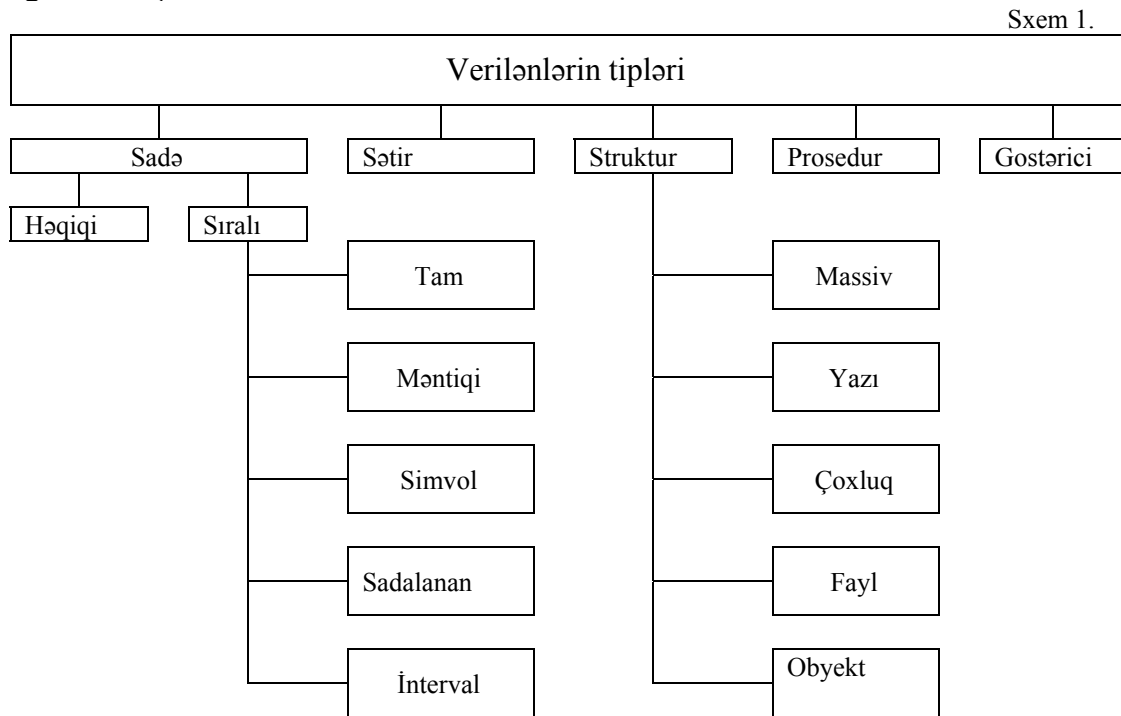
Qeyd edək ki, proqramın istənilən yerində izahedici mətnlərdən ibarət olan şərhlər verilə bilər. Şərhlər vasitəsilə proqramın məqsədi, dəyişənlərin və sabitlərin mənası, proqramın çətin başa düşülən yerləri izah edilir. Yaxşı şərhlə verilən proqram həm oxunaqlı olur, həm də orada baş verən səhvlərin tapılması asanlaşır. Şərhin mətni {} və ya (\* \*) simvolları arasında yazılır. Şərhləri Turbo Pascal dilinin kompilyatoru nəzərə almır. Oudur ki, onlar proqramın yerinə yetirilməsinə təsir göstərmirlər.



## 2.4. Verilənlərin tipləri. Sadə və sətir tiplər

Turbo Pascal dilində verilənlərin çoxsaylı tipi nəzərdə tutulmuşdur. Bunlara standart tiplər deyilir. Standart tiplər əsasında proqramlaşdırıcı yeni tip yarada bilər. Yaradılmış yeni tip proqramın təsvir bölməsində elan olunur.

Turbo Pascal dilində müəyyən edilmiş standart tiplərin təsnifatı sxem 1-də göstərilmişdir.



Sxem 1-dən görüldüyü kimi verilənlərin standart tiplərini şərti olaraq beş qrupa bölürlər:

- Sadə;
- Sətir;
- Struktur;
- Göstərici;
- Prosedur.

Sadə tiplər qrupuna həqiqi və sıralı tiplər daxildir. Sıralı tiplər ona görə *sıralı* adlanır ki, əvvəla, onlarla təyin olunan sabitlərin, dəyişənlərin, ifadələrin qiymətləri sonlu sayda elementlərdən ibarətdir, digər tərəfdən, hər bir qiymətdən əvvəlki və sonrakı qiymətlər mövcuddur. Başqa sözlə, sıralı tipli verilənlərin aldığı qiymətləri nömrələmək mümkündür.

Sıralı tiplərə tam, məntiqi, simvol, sadalanan və interval tipləri aiddir.

Sıralı tiplərdən fərqli olaraq həqiqi tiplərin aldığı qiymətlər tam və kəsr hissələrdən ibarət olan həqiqi ədədlərdir. Hətta ən məhdud diapazonda yerləşən həqiqi ədədləri nömrələmək mümkün deyil.

Turbo Pascal dilində 5 növ tam tip nəzərdə tutulmuşdur. Bu tiplər və onların aldığı qiymətlər oblastı aşağıdakı cədvəldə verilmişdir.

Tip	Qiymətlər oblastı	Ölçüsü
ShortInt	-128...127	1 bayt
Integer	-32768...32767	2 bayt
Logint	-2147483648...2147483647	4 bayt
Byte	0...255	1 bayt
Word	0...65535	2 bayt

Verilənlərin qiymətləri tip üçün müəyyən edilmiş intervaldan kənara çıxarsa, dolub-daşma vəziyyəti qeyd olunmur, lakin bu zaman hesablama düzgün aparılmayacaq.

Tam tipli verilənlərin aldığı qiymətlər onluq və onaltılıq say sistemində verilə bilər. Onaltılıq ədədlər, qarşısında \$ işarəsi olan onaltılıq say sisteminin ədədləridir. Onların dəyişmə diapazonu \$00000000-dan \$FFFFFFF-ə qədərdir.

Tam dəyişənlər üzərində aşağıdakı əməliyyatları aparmaq olar:

- mənimsətmə
- toplama
- çıxma
- vurma
- tam bölmə
- bölmə nəticəsində alınan qalıqın hesablanması

Tam dəyişənlərə yalnız tam ədədlər və tam qiymətli ifadələr mənimsətmək olar.

**Boolean**, **ByteBool**, **WordBool**, **LongBool** standart tiplər məntiqi tip qrupuna aiddir. Məntiqi tipli verilənlər yalnız iki qiymət ala bilər: *true* (doğru) və *false* (yalan). Bu qiymətlər üçün *true* > *false* münasibətləri doğrudur.

Simvol tipli verilənlərin qiymətləri ASCII kod cədvəlinin yalnız bir simvolu ola bilər. ASCII cədvəlində hər bir simvola uyğun olan bir ikilik yazılış var. Bu səbəbdən simvol tipli verilənlər yaddaşda bir bayt tutur. Simvollar onların kodlarının artması sırası ilə nizamlanmışdır, yəni 'A<'B', 'B<'C', və s. Simvolların kodları 0-dan 255 arasında yerləşir. Simvol tipli dəyişənlər aşağıdakı kimi elan olunur.

**Var**

**<identifikator>: char;**

Dəyişənin və sabitin aldığı qiymət – simvollar apastrof daxilində yazılır. Simvolları alternativ olaraq ASCII kodu vasitəsilə də vermək olar. Bu halda kodun qarşısında mütləq # işarəsi yazılmalıdır. Simvol tipli verilənin qiyməti həmçinin idarəedici kodlu xüsusi simvollar ola bilər. Bu simvollar isə ^ işarəsi vasitəsilə yazılır. Məsələn,

**Var c: char;**

**c := ^A;**

Bu yazılış göstərir ki, c simvol dəyişəni ctrl-A qiyməti alır. Simvol tipli verilənlər üzərində müqayisə və mənimsətmə əməliyyatları aparmaq olar.

Turbo Pascal dilində 5 standart həqiqi tiplər: *Real*, *Single*, *Double*, *Extended* *Comp* nəzərdə tutulmuşdur. Bu tiplər təyin etdiyi qiymətlər oblastına, vergüldən sonra ədədlərin sayına və yaddaşdakı həcminə görə fərqlənir.

Tip	Ölçüsü (Bayt)	Qiymətlər oblastı
Single	4	$1.5 \times 10^{-45} - 3.5 \times 10^{38}$
Real	6	$2.9 \times 10^{-39} - 1.7 \times 10^{38}$
Duble	8	$5 \times 10^{-324} - 1.7 \times 10^{308}$
Extended	10	$1.9 \times 10^{-4951} - 1.1 \times 10^{4932}$
Comp	8	$-2^{63} - 1 \dots + 2^{63} + 1$

Qeyd edək ki, həqiqi sabitlər və dəyişənlər üzərində toplama, fərq, hasil və bölmə əməliyyatları yerinə yetirmək olar. Həqiqi dəyişənlər və sabitlər həqiqi qiymətlərlə yanaşı tam qiymətlər də ala bilər. Hesabi əməliyyatlar zamanı aşağıdakı prioritet saxlanılır: əvvəlcə hasil və bölmə, sonra isə vurma və fərq əməliyyatı yerinə yetirilir. Bu prioriteti mötərizələr vasitəsilə dəyişmək olur. Həqiqi verilənlərin aldığı qiymətlər qeyd olunmuş və sürüşgən vergüllü ədədlər kimi kimi təsvir olunur. Məsələn, 0.0015 və 1.5E-3 (yəni  $1,5 \cdot 10^{-3}$ ) yazılışları ekvivalentdir.

Hesab əməlləri yerinə yetirildikdə dolub-daşma vəziyyəti alınarsa, onda proqramda səhv baş verir və uyğun diaqnostik məlumat ekranda əks olunur. Əgər olduqca kiçik ədəd alınarsa, onda bu ədəd başa düşülmür və bu vəziyyət tərtibin pozulması adlanır.

Sadalanən tiplərdə verilənlərin qiymətləri birbaşa sadalanır. Qiymətlər bir-birindən vergül işarəsi ilə ayrılmaqla mötərizə daxilində yazılır. Sadalanən tiplərin ümumi yazılış forması belədir:

**Type**

**<tipin adı>= (W(1), W(2), ..., W(n))**

Bu yazılışda aşağıdakı münasibətlər ödənilir:

- 1) İdentifikator W (i) <>identifikator W(j), əgər i <> j
- 2) İdentifikator W (i) < identifikator W (j), əgər i < j

Məsələn,

**Type janr=('dram', 'poeziya', 'detektiv');**

Bu misalda **janr** tipi aşkar elan olunmuş və dram, poeziya, detektiv qiymətləri alır.

İnterval tipli verilənlər müəyyən diapazonda dəyişən qiymətlər alır və aşağıdakı yazılış formatına malikdir:

**Type <adi>= <diapazon>;**

Diapazonu müəyyən etmək üçün onun ən böyük və ən kiçik qiymətlərini daxil etmək lazımdır.

Məsələn,

```
type
    sıra = 0...100;
```

Qeyd edək ki, sadalanan və interval tipli dəyişəni proqramın təsvir bölməsində birbaşa elan etmək olar. Məsələn,

```
var
    b: 0...100;
    a: ('dram', 'poeziya', 'detektiv');
```

Sətir tipli dəyişənləri elan edən zaman sətirin maksimal uzunluğu, yəni sətirdəki simvolların maksimal sayını göstərmək lazımdır. Bu tipə malik olan dəyişəni **String** işçi sözü vasitəsilə təyin edirlər. **String** sözündən sonra kvadrat mötərizənin içərisində sətirin maksimal uzunluğu yazılır. Sətirin maksimal uzunluğunu göstərən ədəd tam ədəddir. Bu tam ədəd [0; 255] intervalında dəyişir. Məsələn,

```
var
    b: String[18];
```

Sətir tipli dəyişənlərin yaddaşa tutduqları yer onların uzunluğundan bir bayt artıqdır. Sətirə aid olan simvollar 1-dən başlayaraq sətirin uzunluğunu göstərən ədədə qədər indeksləşirlər. Sətirin hər bir simvoluna müraciət etmək üçün bu indeksdən istifadə olunur. Sətir və simvol tipli dəyişənlər eyni bir ifadədə iştirak edə bilirlər.

## 2.5. Operatorlar

Operatorlar Turbo Pascal dilinin əsas elementlərindən biri olub, verilənlər üzərində əməliyyatların yerinə yetirilməsini təmin edir. Operatorlar təyinatına görə aşağıdakı qruplara bölünür:

- Hesabi
- Məntiqi
- Müqayisə
- Birləşdirmə
- Mənimləmə
- Çoxluq
- Tərtib
- Budaqlanma
- Dövr

Bundan əlavə operatorları tətbiq edildiyi operandın sayına görə iki qrupa - unar və binar qruplarına bölürlər. Unar operatorlar bir operanda, binar operatorlar isə iki operanda tətbiq edilir. Binar operatorlar və onların təyinatı aşağıdakı cədvəldə verilmişdir

Operator	Təyinatı
Not	Məntiqi inkar və ya tərtibin inkarı
-	Minus. Dəyişən və ya sabitin işarəsini dəyişir.
In	Elementin çoxluğa daxil olmasını müəyyən edir.
@	Göstərici tip dəyişənə müəyyən qiyməti mənimsətmək üçün istifadə olunur. <sup>1</sup>

Mənimsətmə operatoru adından məlum olduğu kimi dəyişənin adına konkret qiyməti və ya ifadəni mənimsədir. Mənimsətmə operatoru olaraq := işarəsindən istifadə edilir. Mənimsətmə operatoru yerinə yetirildikdə mənimsətmə işarəsindən sağ tərəfdəki ifadənin qiyməti hesablanır və bu qiymət sol tərəfdə duran dəyişənin adını bildirən identifikatora mənimsədir. Qeyd edək ki, dəyişən və ifadə eyni tipə malik olmalıdır. Yalnız, həqiqi dəyişənlərə həqiqi və tam tipli ifadələr və ya qiymətlər mənimsədilə bilər. Bir ölçülü sətir dəyişəninə simvol tipli, simvol dəyişəninə isə bir ölçülü sətir tipli verilənlər və ifadələr mənimsətmək olar. Fayl tipindən başqa bütün tiplər üçün mənimsətmə operatorundan istifadə etmək olar

Hesabi operatorlara aşağıdakılar aiddir:

- + (toplama)
- -(çıxma)
- \* (vurma)
- / (bölmə)
- DIV (tam bölmə)
- MOD (bölmə nəticəsində alınan qalıqın hesablanması)

Hesabi operatorlar ifadədə tam və həqiqi tipli verilənlərlə tətbiq olunur. Bu zaman ifadənin qiymətinin tipi aşağıdakı kimi müəyyən olunur.

Operatorlar	İfadə	Operandların tipi	Nəticənin tipi
+	A+B	real, real integer, integer real, integer	real integer real
-	A-B	real, real integer, integer real, integer	real integer real
*	A*B	real, real integer, integer real, integer	real integer real
/	A/B	real, real integer, integer real, integer	real real real

<sup>1</sup> Bax: Göstəricilər.

Operatorlar	İfadə	Operandların tipi	Nəticənin tipi
Div	A div B	integer, integer	integer
Mod	A mod B	integer, integer	integer

Qeyd edək ki, ifadədə hesabi operatorların yerinə yetirilmə prioriteti aşağıdakı kimidir:

/, \*, **div**, **mod**, +, -

Bu prioriteti dəyişmək üçün “(“ və “)” mötərizələrindən istifadə etmək lazımdır.

Müqayisə operatorları iki operandın<sup>1</sup> qiymətlərini müqayisə edir. Operatorun nəticəsi məntiqi tipli **True** və ya **False** qiyməti olur. Turbo Pascal dilində istifadə edilən müqayisə operatorları aşağıdakı cədvəldə verilmişdir:

Operator	Adı	Misal	Nəticə
=	bərabərdir	A=B	<b>True</b> , əgər A= B, əks halda <i>False</i>
<>	fərqlidir	A<>B	<b>True</b> , əgər A ≠ B, əks halda <b>False</b>
>	böyükdür	A>B	<b>True</b> , əgər A> B, əks halda <b>False</b>
<	kiçikdir	A<B	<b>True</b> , əgər A<B, əks halda <b>False</b>
>=	böyükdür-bərabərdir	A>=B	<b>True</b> , əgər A ≥B, əks halda <b>False</b>
<=	kiçikdir-bərabərdir	A<=B	<b>True</b> , əgər A ≤B, əks halda <b>False</b>

Müqayisə operatorları ifadədə aşağıdakı ardıcılıqla yerinə yetirilir:

=, <>, <, >, <=, >=

Qeyd edək ki, ifadədə həm hesabi operatorlar, həm də müqayisə operatorları iştirak etdikdə, əvvəlcə hesabi operatorlar, sonra isə müqayisə operatorları yerinə yetirilir.

Müqayisə operatorları tam, həqiqi, simvol və sətir verilənlərlə tətbiq olunur. Sətirlərin müqayisəsi soldan sağa doğru üst-üstə düşməyən simvola qədər davam edir. Simvol tipli verilənlərin müqayisəsi onların ASCII cədvəlində yerləşmə kodunun müqayisəsi əsasında yerinə yetirilir. Məsələn, ‘A<’B’, ‘B<’C’, və s.

Məntiqi tip operatorlar ifadədə məntiqi operandlara tətbiq edilir. Onlar aşağıdakı cədvəldə verilmişdir.

Operator	Misal	A-nın qiyməti	B-nın qiyməti	Nəticə	Operatorun adı və riyazi işarəsi
not	not A	true		false	məntiqi inkar ( $\bar{1}$ )
		false		True	

<sup>1</sup> Müqayisə operatorunun tətbiq edildiyi operandlar eyni tipli və ya uyuşan tipli olmalıdır.

Operator	Misal	A-nın qiyməti	B-nın qiyməti	Nəticə	Operatorun adı və riyazi işarəsi
and	A and B	true	true	true	məntiqi və (∨)
		true	false	False	
		false	true	false	
		false	false	False	
or	A or B	true	true	true	məntiqi və ya (∧)
		true	false	true	
		false	true	true	
		false	false	False	
xor	A xor B	true	true	false	məntiqi istisna (⊕)
		true	false	true	
		false	true	true	
		false	false	False	

Cədvəldə verilmiş misallardan göründüyü kimi, **Not** məntiqi inkar operatoru unar tip operatorudur və məntiqi operandın qiymətini dəyişdirir. **And** operatoru hər iki operandın eyni qiymət aldığı halda, **True** qiyməti, **or** operatoru operandların heç olmasa biri **True** qiymət aldığıda, **xor** operatoru isə birinci operandın qiyməti ikinci operandın qiymətindən fərqli olduqda **True** qiyməti alır.

Birləşdirmə operatoru -“+” və ya “&” ifadədə simvol və sətir tipli verilənlərlə tətbiq edilir və nəticə sətir tipli olur<sup>1</sup>.

Tərtib operatorlarına aşağıdakı operatorlar aiddir:

- ◇ **not**-tərtibin inkarı
- ◇ **and**-tərtibin hasili;
- ◇ **or**-tərtibin toplanması;
- ◇ **xor**-tərtibin toplanmasının istisnası;
- ◇ **shl** -sola doğru tərtibin sürüşdürülməsi;
- ◇ **shr** -sağa doğru tərtibin sürüşdürülməsi;

Tərtib operatorları proqram kompilyasiya edilərkən ikilik say sistemə çevrilmiş ədədlər üzərində əməliyyatlar yerinə yetirir. Məsələn, əgər A bayt tipli

<sup>1</sup> Nəticədə alınan sətirin uzunluğu 255-dən çox olmamalıdır.

dəyişəninə 6 qiyməti mənimsədilibsə, onda ikilik say sistemində onun aldığı qiymət 00001010 olacaqdır. **Not** A -nın qiyməti isə 11110100 olacaq.

**xor** operatoru aşağıdakı kimi dəyişənlərin qiymətlərini toplayır:

1 **xor** 1=0

1 **xor** 0=1

0 **xor** 1=1

0 **xor** 0=0

Toplamının nəticəsi onluq ədədə çevrilir.

**k shl N** (sola doğru sürüşdürmə) operatoru k ədədinin ikilik təsvirini N mövqə sola sürüşdürür. Məsələn, shl 17=256. Çünki, 2 ədədinin ikilikdə təsviri 2 bayt yer tutur və 20000000000000010 kimidir. Hər biti 7 mövqə sola sürüşdürsək, 0000001000000000 alırıq. Bu isə 256 onluq ədədidir.

Oxşar olaraq, sağa doğru sürüşdürmə operatoru yerinə yetirilir. Bu halda göstərilən mövqə qədər sağa doğru sürüşdürmə aparılır. Məsələn,

160 **shr** 2=40,

256 **shr** 7=2

İfadədə bu operatorlar aşağıdakı ardıcılıqla yerinə yetirilir:

/ , \* , div , mod , and , or , xor , shl .

Çoxluq operatorları adından məlum olduğu kimi çoxluq tipli verilənlər<sup>1</sup> üzərində əməliyyat aparır. Onların siyahısı və təyinatı aşağıdakı cədvəldə verilmişdir.

Operator	Operatorun adı	Operatorun təyinatı
=	Bərabərdir	Çoxluqların bərabərliyini müəyyən edir. Eyni elementli çoxluqlar üçün nəticə True qiymət olur.
<>	Fərqlidir	Fərqli elementli çoxluqlar üçün nəticə True qiymət olur
>=	Böyükdür-bərabərdir	Əgər A çoxluğu B çoxluğuna daxildirsə onda B >=A ifadəsi True qiyməti alır.
<=	Kiçikdir-bərabərdir	Əgər B çoxluğu A çoxluğuna daxildirsə onda B <=A ifadəsi True qiyməti alır.
+	Birləşmə	İki çoxluğun birləşməsini təmin edir.
*	Kəsişmə	İki çoxluğun kəsişməsini təmin edir.
-	Fərq	İki çoxluğun fərqini təmin edir.
in	Daxildir	Elementin çoxluğa daxil olmasını yoxlayır

## 2.6. Giriş və çıxış prosedurları

Giriş və çıxış prosedurları **System** modulunun standart prosedurlarındandır və verilənlərin giriş və çıxış əməliyyatlarını yerinə yetirir. Giriş əməliyyatı zamanı ilkin verilənlərin daxil edilməsi - verilənlərin klaviaturadan və ya xarici fayldan operativ yaddaşa ötürülməsi prosesi baş verir. Çıxış əməliyyatı zamanı nəticə operativ yaddaşdan xarici daşıyıcıya ötürülür. Xarici daşıyıcı olaraq monitor, printer, faylın yerləşdiyi informasiya daşıyıcısı istifadə oluna bilər. Bu zaman

<sup>1</sup> Bax: Çoxluqlar.



müvafiq olaraq proqramın nəticəsi ekranda əks oluna, kağızda çap oluna bilər, digər fayla yazıla bilər.

Giriş və çıxış əməliyyatlarını yerinə yetirmək üçün istifadə edilən əsas prosedurlar aşağıdakılardır:

**Read, ReadLn, Write, WriteLn.**

Giriş prosedurları **Read, ReadLn** prosedurları aşağıdakı yazılış formatına malikdir:

**Read** (x1, x2, ..., xn) və ya **Read** (FV, x1, x2, ..., xn)

**ReadLn** (x1, x2, ..., xn) və ya **ReadLn** (FV, x1, x2, ..., xn)

Bu yazılışlarda x1, x2, ..., xn dəyişənlər, FV –fayl dəyişənidir. **Read** (FV, x1, x2, ..., xn) və **ReadLn** (FV, x1, x2, ..., xn) prosedurları fayldan dəyişənləri qiymətlərinin oxunmasını<sup>1</sup>, **Read** (x1, x2, ..., xn) **Read** (x1, x2, ..., xn) prosedurları isə klaviatüradan dəyişənlərin qiymətlərinin daxil edilməsini təmin edir.

x1, x2, ..., xn dəyişənlərinin qiymətləri aralarında ən azı bir probel olmaqla klaviatüradan daxil edilir və ekranda göstərilir. **Read** prosedurunda verilmiş dəyişənlərin qiymətlərini daxil etdikdən sonra **Enter** düyməsini sıxmaq lazımdır. Daxil olunan dəyişənlərin qiymətləri onların tipinə uyğun olmadıqda giriş-çıxış xarakterli səhv meydana çıxır.

Bir proqramda bir neçə **Read** prosedurundan istifadə etmək olar. Bu halda əvvəlcə birinci **Read** prosedurunda olan dəyişənlərin qiymətləri klaviatüradaxil edilir, sonra **Enter** düyməsi sıxılır və ikinci **Read** prosedurunda olan dəyişənlərin qiymətləri daxil edilir və **Enter** düyməsi sıxılır və s.

**ReadLn** prosedurunun yerinə yetirilməsi qaydası **Read** operatoruna oxşardır. Fərqli cəhəti ondadır ki, birinci **ReadLn** prosedurunun axıncı dəyişənin qiyməti daxil edildikdən sonra ekranda kursor yeni sətirə keçir və növbəti **ReadLn** prosedurunun dəyişənlərinin qiymətləri yeni sətirdən oxunur.

Çıxış prosedurları – **Write, WriteLn** prosedurları aşağıdakı yazılış formatına malikdir:

**Write** (y1, y2, ..., yn, <format>)

**WriteLn** (y1, y2, ..., yn, <format>)

Bu yazılışlarda y1, y2, ..., yn **Integer, Byte, Real, Char** və s. tipə malik ifadə və dəyişənlərdir. <format> məcburi olmayan parametri çıxış formatını müəyyən edir. Tam, simvol və məntiqi tipli ifadələri və ya dəyişənləri çıxışa verdikdə format parametri olaraq çıxış sahəsinin uzunluğunu – dəyişənin və ya ifadənin qiyməti üçün ayrılan mövqelərin sayını vermək olar. Tam dəyişənin və ya ifadənin qiyməti çıxış sahəsinin sağ hissəsindən başlayaraq yazılır. Məsələn,

**Write** (N: 6);

Əgər N=250 olarsa, onda prosedurun nəticəsi \_\_ \_250 olar.

<sup>1</sup> Fayllardan verilənlərin oxunması dərsləyin Fayl tipli dəyişənlər. Fayllarla əməliyyatlar paraqrafında verilmişdir.

Həqiqi ifadələri və ya dəyişənləri çıxışa vermək üçün format olaraq “sahənin uzunluğu: dəqiqlik” yazılışından istifadə olunur. Bu yazılışda dəqiqlik onluq nöqtədən sonra gələn rəqəmlərin sayını göstərir. Ümumiyyətlə, çıxış prosedurunda sahənin uzunluğunu və dəqiqliyi göstərməmək də olar. Bu halda Turbo Pascal kompilyatoru tam ədədlər üçün 10 mövqe, həqiqi ədədlər üçün isə 20 mövqe ayırır. Həqiqi ədədlər ekpansional formada çıxışa verilir.

**WriteLn** proseduru **Write** proseduruna oxşayır. Fərq ondadır ki, birinci **WriteLn** prosedurunda olan axırncı dəyişənin qiyməti çıxışa verildikdən sonra növbəti sətərə keçid təmin olunur. Növbəti çıxış prosedurundakı dəyişənlərin qiymətləri təzə sətirdən çap olunurlar. Parametrsiz yazılmış **WriteLn** operatoru yalnız təzə sətərə keçməyi təmin edir.

Məsələn,  $y = x^n$  funksiyasını hesablama proqramı aşağıdakı kimi olacaq:

```

Program N_2;
Var
  n, x, y : real;
BEGIN
  Readln(x);
  Readln(n);
  y := exp(n*ln(x));
  Writeln('Cavab y=', y:10:6);
END.

```

## 2.7. Şerti keçid operatoru

Şerti keçid operatorları verilmiş şərtədən asılı olaraq hər hansı operatorun yaxud operatorlar qrupunun yerinə yetirilməsini və ya yerinə yetirilməməsini təmin edir.

Şerti keçid operatoru aşağıdakı yazılış formatına malikdir:

```

if <şert> then <operator_1> else <operator_2>;

```

Burada <şert>-məntiqi ifadədir. Operator yerinə yetirilərkən məntiqi ifadənin qiyməti doğru olarsa, onda <operator\_1>, əks halda <operator\_2>, yerinə yetirilir. Burada <operator\_1> və <operator\_2> tək bir operatorndan və ya **begin** və **end** arasında yerləşən operatorlar qrupundan ibarət ola bilər. **Misal 1.**

```

Program Oper_If;
Var
  a, b : Integer;
  c    : Char;
Begin
  Write('İki eded daxil edin: ');
  Readln(a,b);
  Write('Onlari toplayaq (y/n)? ');
  Readln(c);
  If c='y' Then
  Begin
    a:=a+b;

```

```

Writeln('Netice =', a); End
Else
  Writeln('emeliyyat yerine yetirilmedi');
End.

```

Program iki ədəd üzərində toplama əməliyyatının aparılmasını istifadəçidən “soruşur”. İstifadəçi klaviaturanın “y” düyməsini sıxdıqda toplama əməliyyatı yerinə yetirilir və nəticə monitorda əks olunur. Əks halda, toplama əməliyyatı yerinə yetirilmir və bu haqda məlumat monitorda əks olunur.

Praktikada çox zaman şərti keçid operatorunun qısa yazılış formatından istifadə edilir:

```

if <şərt> then <operator_1>;

```

Bu formada operator yerinə yetirilərkən məntiqi ifadənin qiyməti doğru olarsa, onda <operator\_1>, əks halda, programdakı növbəti operator yerinə yetirilir. Qeyd edək ki, bir şərti keçid operatoru digər şərti keçid operatorunun tərkibinə daxil ola bilər:

```

if <şərt> then
if <şərt> then <operator 1>
else <operator 2>;

```

və ya

```

if <şərt> then <operator 1>
else if <şərt> then <operator 2>;

```

Misal 2:

```

Program ves;
CONST A1='Yüngül';
        A2='Orta';
        A3='Ağır';
        A4='Daha ağır';
var
  V : integer;
BEGIN
Write('Cekini daxil edin V');
ReadLn(v);
  if V<62 then WriteLn(A1)
else
  if (V>62) and (V<75) then WriteLn(A2)
else
  if (V>75) and (V<88) then WriteLn(A3)
else
  WriteLn(A4)
END.

```

Program idmançının kütləsindən asılı olaraq, onun hansı çəki kateqoriyasına aid olmasını müəyyən edir. Belə ki, 62 kq az çəkili idmançı yüngül, 62 -75 kq çəkili idmançı orta, 75-88 kq çəkili idmançı ağır, bundan artıq çəkili idmançı isə daha ağır çəkili idmançı kateqoriyasına aid olunur.

Misal 3:

```

Program example1;

```

```

var x, f:real;
begin
  write('Daxil edin x');
  readln(x);
  if x<-1 then f:= -x-1
else
  if (x>=-1) and (x<0) then f:= x+1
else
  if (x>=0) and (x<1) then f:= -x+1
else
  f:= x+1;
  writeln('F= ',f);
end.

```

Proqram aşağıdakı funksiyanın hesablanmasını təmin edir.

$$f(x) = \begin{cases} -x-1, & x < -1 \\ x-1, & -1 \leq x < 0 \\ -x+1, & 0 \leq x < 1 \\ x+1, & x \geq 1 \end{cases}$$

## 2.8. Seçim və ya variant operatoru

Seçim və ya variant operatoru şərti keçid operatorunun ümumiləşməsidir. Bu operator yerinə yetirildikdə mümkün variantlardan biri yerinə yetirilir. Operator selektor adlanan ifadədən və seçim üçün istifadə olunan nişanlanmış operatorlar siyahısından ibarətdir. Şərti keçid operatorunda olduğu kimi burada da **else** işçi sözü işləyə bilər. Seçim və ya variant operatorunun yazılış formatı aşağıdakı kimidir:

- **Tam formatda yazılışı**

```

case <selektor-ifadə> of
<nişan 1>: <operator 1;>
< nişan 2>: <operator 2;>
... ..
< nişan n>: <operator n;>
else <operator>
end;

```

- **Qısa formatda yazılışı**

```

case <selektor-ifadə> of
< nişan 1>: <operator 1;>
< nişan 2>: <operator 2;>
... ..
< nişan n>: <operator n;>
end;

```

Burada < nişan 1>, < nişan 2>, ..., < nişan n> selektorun qiymətləridir. Seçim və ya variant operatoru aşağıdakı kimi işləyir. Proqram yerinə yetirildikdə əvvəlcə selektor-ifadənin qiyməti hesablanır, sonra isə selektor-

ifadənin qiymətinə bərabər olan nişan müəyyənləşdirilir və ona uyğun olan operator yerinə yetirilir. Selektorun qiymətinə bərabər olan sabit olmadıqda **else** xidməti sözündən sonra yazılan operator, operatorun qısa yazılışında isə operatorndan sonra gələn operator yerinə yetirilir. Qeyd edək ki, həqiqi və sətir tipinə aid olan dəyişənləri selektor olaraq istifadə etmək olmaz.

**Misal:**

```

Program Aysel_Proqram;
Var
  X, Y : Real;
  C : Char;
BEGIN
  Write('?ki eded daxil edin: ');
  Readln(X,Y);
  Write('Emeliyyat simvolu daxil edin: ');
  Readln(C);
  Case C Of
    '+' : Begin
      X:=X+Y;
      Writeln('Toplama');
      End;
    '-' : Begin
      X:=X-Y;
      Writeln('Ferq');
      End;
    '*' : Begin
      X:=X*Y;
      Writeln('Hasil');
      End;
    '/': Begin
      X:=X/Y;
      Writeln('Qismet');
      End;
  End;
  Writeln('Netice= ',X:12:4);
END.

```

Proqram istifadəçinin klaviaturadan daxil etdiyi "+", "-", "\*" və "/" simvollarına uyğun olaraq X və Y dəyişənlərinin cəmini, fərqi, hasilini və qismətini hesablayır.

## 2.9. Şərtsiz keçid operatoru

Şərtsiz keçid operatoru proqramda idarəni nişanlanmış operatora verir, başqa sözlə operatorların yerinə yetirilmə ardıcılığını dəyişir. Operator aşağıdakı yazılış formatına malikdir:

```
GOTO n;
```

bu yazılışdan nişanlanmış operatorun nişanıdır.

## 2.10. Dövr operatorları

Dövr operatorları proqramda dövrü alqoritmləri təsvir etmək üçün istifadə edilir. Turbo Pascal dilində üç növ dövr operatoru vardır:

- Parametrlı dövr operatoru;
- Sonrakı şərtli dövr operatoru
- İlkin şərtli dövr operatoru

### Parametrlı dövr operatoru.

Dövrlərin sayı məlum olduqda parametrlı dövr operatorundan, əks halda isə sonrakı şərtli dövr operatorundan və ya ilkin şərtli dövr operatorundan istifadə edilir.

Parametrlı dövr operatoru iki formada istifadə olunur:

1. **for** <dövr parametri>:= <n> **to** <m>**do** <operator>;

burada <n> və <m> dövr parametrinin müvafiq olaraq başlanğıc və son qiymətlərini müəyyən edən ifadələrdir. Dövr parametri, <n> və <m> ifadələrinin eyni tipli qiymətlər almalıdır və bu qiymətlər həqiqi tipə aid ola bilməzlər. <operator> dövrün gövdəsi adlanır və sadə - tək bir operatorndan və mürəkkəb operatorndan - **begin** və **end** arasında yerləşən operatorlar qrupundan ibarət ola bilər.

Dövr operatoru yerinə yetirildikdə dövrün parametrinin aldığı cari qiymət parametrin son qiymətilə müqayisə olunur. Cari qiymət son qiymətdən kiçik olduqda dövrün gövdəsi yerinə yetirilir, dövr parametrinin qiyməti vahid qədər artır və bu qiymət yenidən parametrin son qiymətilə müqayisə olunur. Bu proses parametrin cari qiymətinin son qiymətdən böyük olana qədər davam edir. Bundan sonra dövr operatorundan sonra yazılan operator yerinə yetirilir.

**Misal 1.**  $s = \sum_{i=1}^{100} i$  cəmini hesablamalı.

```

Program Turqut_prog1;
Const n=100;
var i,s:integer;
begin
    s:=0;
for i:=1 to n do
    s:=s+i;
writeln ('s=', s);
end.

```

2. **for** <dövr parametri>:= <n> **downto** <m> **do** <operator>;

Bu yazılışda da <n> və <m> dövr parametrinin müvafiq olaraq başlanğıc və son qiymətlərini müəyyən edən adsız sabitlər və ya ifadələrdir, eyni tipli olmalıdır və həqiqi tipə aid ola bilməzlər. <operator> dövrün gövdəsi olub, sadə - tək bir operatorndan və mürəkkəb operatorndan - **begin** və **end** arasında yerləşən operatorlar qrupundan ibarət ola bilər.

Dövr operatoru yerinə yetirildikdə dövr parametrinin aldığı cari qiymət parametrin son qiymətilə müqayisə olunur. Cari qiymət son qiymətdən böyük olduqda dövrün gövdəsi yerinə yetirilir, dövr parametrinin qiyməti vahid qədər azalır və bu qiymət yenidən parametrin son qiymətilə müqayisə olunur. Bu proses parametrin cari qiymətinin son qiymətdən kiçik olana qədər davam edir. Bundan sonra dövr operatorundan sonra yazılan operator yerinə yetirilir.

**Misal 2.** Birdən 100-ə qədər tam ədədləri azalma sırası ilə çap etməli.

```
Program Turqut_proqram_1;
Const n=100;
var i,s:integer;
begin
  s:=0;
  for i:=100 downto 1 do
    writeln (i);
  end.
```

Göründüyü kimi, parametrlı dövr operatorunda dövr parametri ancaq vahid qədər artıb-azala bilər. Bu operatorun çatışmayan cəhətidir. Bu çatışmayan cəhəti sonrakı şərtlı dövr operatoru və ilkin şərtlı dövr operatoru vasitəsilə aradan qaldırmaq mümkündür.

Qeyd etmək lazımdır ki, dövrün gövdəsində dövrün parametrinin qiymətini dəyişmək olmaz. Dövr qurtardıqdan sonra dövrün parametrinin qiyməti son qiymətə bərabər olur. Qeyd edək ki, dövrün vaxtından əvvəl – müəyyən şərt ödənildikdə yerinə yetirilməsini dayandırmaq üçün **break** operatorundan istifadə etmək məsləhətdir. Həmçinin, dövr sona çatmamış dövrün daxilindən **GOTO** n<sup>1</sup> operatorunun vasitəsilə idarəni dövrədən kənara – nişanlanmış digər operatora vermək olar.

### Misal 3

```
Program Turqut_proqram_3;
var n,i:longint;
    s:boolean;
begin
  write ('N=');
  readln (n);
  s:=true;
  for i:=2 to n-1 do
    if n mod i = 0 then begin
      s:=false;
      break;
    end;
  if s=true then writeln ('Sade ededdir')
  else
    writeln ('sade eded deyil');
  end.
```

---

<sup>1</sup> Burada n nişanı bildirir.

Program istifadəçinin daxil etdiyi tam ədədin sadə ədəd olmasını yoxlayır. Programda verilmiş **break** operatoru dövrün yerinə yetirilməsini dayandırır və idarəni sonrakı operatora verir.

Praktikada bir-birinə daxil olan dövr operatorlarından da (mürəkkəb dövrləri təşkil etmək üçün) geniş istifadə olunur.

**Misal 4.**  $h = \pi/100$  addımla  $y = a \sin(x) - \frac{\cos(x)}{a}$  funksiyasının qiymətini

$a = \{1; 1.5; 2; 2.5; 3\}$  qiymətlərində hesablamalı, harada,  $x \in [0, \pi]$ .

```

Program tabl;
var
  y, x, a, dx : real;
  i, j: integer;
Begin
Writeln(' Hesablamalı: y=A*sin(x)-cos(x)/A; ');
Writeln('-----');
Writeln('| X | A=1.0 | A=1.5 | A=2.0 | A=2.5 | A=3.0 |');
Writeln('-----');
  dx := pi/100;
  for i:= 0 to 100 do
    begin
      x:= dx*i;
      Write( x:8:4 );
      for j := 1 to 5 do
        begin
          A := 0.5*(j+1);
          y := A*sin(x)-cos(x)/A;
          Write(y:8:4)
        end;
      writeln ;
    end;
  End.

```

Aşağıdakı cədvəlin köməyi ilə parametrlı dövr operatorundakı dövrün gövdəsinin təkrarlanmasının sayını təyin etmək olar:

Operator	n<m	n=m	n>m
for...to...do	m-h+1	1 dəfə	yerinə yetirilmir
for...downto...do	yerinə yetirilmir	1 dəfə	n-m+1

### 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** <operator> **Until** <şərt>;

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 edən operatorlar təkrar yerinə yetirilir. Əgər məntiqi ifadənin qiyməti *true* olarsa, onda dövr sona çatır və dövr operatorundan sonrakı operator



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.

**Misal 5.**  $y = \sum_{n=1}^{100} \frac{1}{n^2}$  funksiyasını hesablamalı.

```
Program Turqut_proqram_4;
Const
    k=100;
Var
    n:integer;    y:real;
begin
    y:=0;
    n:=1;
repeat
    y:=y+1/(n*n);
    n:=n+1;
until n=k+1;
    Writeln(y);
end.
```

### 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 <şərt> do <dövrün gövdəsi>;
```

Burada <şərt> məntiqi ifadə, <dövrün gövdəsi> isə sadə və ya mürəkkəb operatorudur. 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, <şərt>-in qiyməti yalan (*false*) olarsa, onda dövrün gövdəsini təşkil edən operatorlar bir dəfə də olsun yerinə yetirilmir. Lakin sonrakı şərtli dövr operatorunda dövrün gövdəsi <şərt>-dən 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.

**Misal 6.**

```
Program Turqut_6;
Var
    I : Integer;
    St : String;
BEGIN
    Write('Sozu daxil edin ');
    Readln(St);
```

```

I:=0;
While I < 10 Do
  Begin
    Writeln(St);
    I:=I+1;
  End;
END.

```

Program istifadəçinin daxil etdiyi sözü 10 dəfə təkrar çap edir.

## 2.11. Çoxluqlar

Çoxluq-verilənlərin strukturlanmış tipi olub, müəyyən əlamətə görə qarşılıqlı əlaqəli olan elementlər toplusudur. Çoxluqların elementlərinin sayı sonlu olur və 0-dan 256-ə qədər dəyişə bilər. Elementlərinin sayı sıfır olan çoxluq boş çoxluq adlanır. Çoxluğun elementləri massivlərdən fərqli olaraq indekslənməyib və müəyyən nizamlı düzülüşə malik deyil. Çoxluqda elementlər təkrar oluna bilməz, yəni eyni bir element çoxluqda yalnız bir dəfə iştirak edə bilər.

Çoxluq tipli dəyişənlər operativ yaddaşda 16 bayt yer tutur və aşağıdakı formada elan olunur

```

type
  <tipin adı>=set of <baza tipi>;
var
  <dəyişənin adı>: <tipin adı>;

```

və ya

```

var <dəyişənin adı>: set of <baza tipi>;

```

Məsələn,

```

type
  ByteSet = set of byte;
  CharSet = set of char;
  Digits = set of '0'..'9';
var s: ByteSet;
  m: CharSet;
  n: Digits;
  k: set of 1..3;

```

Programda çoxluq tip sabitlər və çoxluq tip dəyişənin aldığı qiymətlər kvadrat və ya dairəvi mötərizə daxilində göstərilir. Məsələn, [1,2,3,4], ['a', 'b', 'c'], ('a'..'z'). Boş çoxluq [ ] işarə olunur. Çoxluq tipli dəyişənin qiymətlər oblastı baza tipli elementlərdən təşkil olunmuş bütün mümkün olan altçoxluqlar yığıdır. Əgər baza tipi N qiymət alarsa, onda çoxluq tipli dəyişən  $2^N$  sayda müxtəlif qiymətlər alar. Məsələn,

```

var
  T : Set of 1..3;

```

kimi elan olunmuş çoxluq aşağıdakı qiymətlər ala bilər:

```

[], [1], [2], [3], [1,2], [1,3], [2,3], [1,2,3]

```

Çoxluq haqqında riyaziyyatdan məlum olan anlayışlar çoxluq tipli verilənlər üçün də eynidir və onlar üzərində eyni əməliyyatlar aparmaq olar<sup>1</sup>.

Çoxluq tip dəyişənlər üzərində aşağıdakı əməliyyatları yerinə yetirmək mümkündür:

- *mənimsətmə əməliyyatı*;

Mənimsətmə əməliyyatı mənimsətmə operatoru<sup>2</sup> vasitəsilə yerinə yetirilir və çoxluq tip dəyişənə element və ya digər çoxluq tip dəyişən mənimsədilməsini təmin edir.

- müqayisə əməliyyatı;

Müqayisə əməliyyatları =, <>, >=, <= operatorları vasitəsilə yerinə yetirilir və müvafiq olaraq iki çoxluğun ekvivalentliyini<sup>3</sup>, fərqli olmasını<sup>4</sup> və bir çoxluğun digər çoxluğa daxil olmasını<sup>5</sup> müəyyən edir. Məsələn,

```

Program a1;
var
  a,s: set of 0..3;
begin
  a:=[2,1,0,3];
  s:=a;
  writeln (s=a);
  writeln (s<>a);
  writeln (s<=a);
  writeln (s>=a)
end.

```

Verilmiş proqramda a və s çoxluq tipli dəyişənlər elan olunur. a dəyişəninə çoxluğun elementləri mənimsədilir. s dəyişəninə isə a dəyişəni mənimsədilir. s və a çoxluqlarının ekvivalentliyi, fərqli olması,  $s \in a$  və  $a \in s$  şərtləri yoxlanaraq nəticə write xaricətmə proseduru vasitəsilə ekrana göndərilir. s və a çoxluqları ekvivalent olduğundan nəticə aşağıdakı kimi olacaq:

```

True
False
True
True

```

- *birləşmə əməliyyatı*;<sup>6</sup>

Birləşmə əməliyyatı çoxluq üçün “+” operatoru vasitəsilə yerinə yetirilir. Məsələn,

```

Program a2;
var
  s1,s2,s3,s: set of byte;

```

<sup>1</sup> Əməliyyatlar yalnız bütövlükdə çoxluq üzərində aparılır.

<sup>2</sup> Bax: Operatorlar

<sup>3</sup> A və B çoxluqları eyni elementlərdən ibarət olarsa, onda bu çoxluqlar ekvivalentdir.

<sup>4</sup> A və B çoxluqlarının heç olmasa bir elementi fərqlənərsə, bu çoxluqlar fərqli olacaq.

<sup>5</sup> B-nin bütün elementləri A çoxluğuna daxil olduqda B çoxluğu A çoxluğunun altçoxluğudur.

<sup>6</sup> A və B çoxluqlarının birləşməsi (A+B) bu çoxluqların elementlərindən ibarət olan üçüncü bir çoxluqdur.

```

begin
  s1:=[1..4];
  s2:=[2..5];
  s3:=[] ;
  s:=s1+s3;
  writeln (s1=s);
  s:=s1+s2;
  write (s1=s)
end.

```

▪ *kəsişmə əməliyyatı;*

Kəsişmə əməliyyatı çoxluq üçün \* operatoru vasitəsilə yerinə yetirilir.

Məsələn,

```

var
  s1,s2,s3,s: set of byte;
begin
  s1:=[1..4];
  s2:=[2..5];
  s3:=[] ;
  s:=s1*s3;
  writeln (s1=s);
  s:=s1*s2;
  write (s1=s)
end.

```

▪ *çoxluqların fərqi əməliyyatı;*

Fərq əməliyyatı çoxluq üçün "-" operatoru vasitəsilə yerinə yetirilir.

Məsələn,

```

var
  s1,s2,s3,s: set of byte;
begin
  s1:=[1..4];
  s2:=[2..5];
  s3:=[] ;
  s:=s1-s3;
  writeln (s1=s);
  s:=s1-s2;
  write (s1=s)
end.

```

▪ *elementin çoxluğa aid olmasını yoxlamaq əməliyyatı.*

Elementin çoxluğa aid olmasını yoxlamaq əməliyyatı çoxluq üçün "in" operatoru vasitəsilə yerinə yetirilir. Məsələn,

```

var
  s1,s2,a,b,c: set of byte;
begin
  s1:=[1..4];
  s2:=[2..5];
  a:=s1-s2;
  b:=s1*s2;

```

```

c:=s1+s2;
writeln (5 in a, ' ', 5 in b, ' ', 3 in c);
end.

```

Proqramda çoxluqlardan istifadə olunması nəticəsində məsələnin həlli alqoritmi sadələşir, müəyyən hallarda yaddaşa qənaət edilir, kompilyasiya və yerinə yetirilmə vaxtı azalır. Çoxluq tipli dəyişənlərdən istifadə olunmasının çatışmayan cəhəti odur ki, onları giriş və çıxış prosedurlarında istifadə etmək olmaz. Bu hallarda proqramlaşdırıcı özü uyğun giriş-çıxış prosedurlarını yazmalıdır.

Çoxluqların bir növü də sabit çoxluqlardır. Sabit çoxluqlar bir-birindən vergüllə ayrılan və kvadrat mötərizənin arasında yerləşən elementlərdən ibarətdir.

## 2.12. Massivlər

Massivlər-verilənlərin strukturlaşmış növü olub, eyni tipli sonlu sayda elementlərdən ibarətdir. Massivin hər bir elementinə onun elementlər ardıcılığında mövqeyini birqiymətli təyin edən indeks uyğundur. İndeks olaraq həqiqi tip istisna olmaqla ixtiyari sadə tipli dəyişən və ya ifadə iştirak edə bilər<sup>1</sup>. Massivlər aşağıdakı formada elan olunur:

```

array [indeks] of < elementlərin tipi >;

```

Massivin elementinin tipi öz növbəsində massiv ola bilər. Belə massiv ikiölçülü massiv adlanır. İndeksin sayı massivin ölçüsünü müəyyən edir. Belə ki, massivin elanı zamanı bir indeks verilsə, həmin massiv bir ölçülü, iki indeks verilsə iki ölçülü,  $n$  indeks verilsə  $n$  ölçülü massiv adlanır. Massivin ölçüsü kompyuterin operativ yaddaşının tutumu ilə məhdudlaşır. Adətən, birölçülü massivlərdən vektorları, ikiölçülü massivlərdən matrisləri, çoxölçülü massivlərdən isə tenzorları təsvir etmək üçün istifadə olunur. Məsələn, 100 elementdən ibarət B vektorunu aşağıdakı formada elan etmək olar.

```

type
    vek:array [1..100] of integer;
var
    B:vek;
və ya
    var
        B: array [1..100] of integer;

```

Eyni qaydada ikiölçülü massivlər də elan olunur. Məsələn, A matrisi proqramda aşağıdakı formada elan olunur.

<sup>1</sup> İndeksin tipi onun dəyişmə sərhəddini müəyyən edir.

$$A = \begin{pmatrix} 1 & 23 & -45 & 9 & 2 \\ 0 & 45 & -98 & 12 & 4 \\ 8 & 23 & 45 & -90 & 5 \\ 32 & 3 & 34 & 3 & 6 \end{pmatrix}$$

**type**

```
vek1:array [1..5] of integer;
vek2:array [1..4] of vek1;
```

**var**

```
a:vek2;
```

və ya

**var**

```
a: array [1..4,1..5] of integer;
```

Massivin elementlərinə müraciət onun indeksi vasitəsilə yerinə yetirilir. Bunun üçün massivin adını və kvadrat mütərizənin içərisində indeksi göstərmək lazımdır. Məsələn, B[2] yazılışı massivin ikinci, B [10] yazılışı massivin onuncu elementinə, A[2,3] yazılışı A matrisinin ikinci sətirlə üçüncü sütunun kəsişməsində yerləşən elementə<sup>1</sup> müraciəti təmin edir.

Birölçülü massivin elementləri yaddaşda ardıcıl olaraq yerləşirlər. İkiölçülü massivin elementləri isə yaddaşda sağ indeksin artmasına uyğun olaraq ardıcıl yerləşirlər. Məsələn, təyin olunan A massivinin elementlərinin operativ yaddaşda yerləşmə sxemi aşağıdakı formada olacaq:

```
A[1,1], A[1,2], A[1,3], A[1,4], A[1,5], A[2,1], A[2,2],
A[2,3], A[2,4], A[2,5], A[3,1], A[3,2], A[3,3], A[3,4],
A[3,5], A[4,1], A[4,2], A[4,3], A[4,4], A[4,5]
```

Qeyd edək ki, massivlərin elementlərinin sayı indeksin təyin olunduğu intervalı aşmamalıdır. Əks halda, proqram yerinə yetirilərkən səhv aşkar olunacaq.

Massiv elan olunduqdan sonra massivin elementləri təyin edilməlidir, yəni, massivin hər bir elementinə onun baza tipinə uyğun bir qiymət və ya ifadə mənimsədilməlidir. Bu dövr operatorlarının və **Read, Readln** daxiletmə operatorundan istifadə etməklə yerinə yetirilir. Massivin elementlərinin ekrana və ya kağıza çıxışı isə dövr operatorundan istifadə etməklə **Write, Writeln** prosedurları ilə yerinə yetirilir. Məsələn, A və B massivlərinin elementləri aşağıdakı operatorlar vasitəsilə daxil edilə və ekranda əks oluna bilər.

**var**

```
A: array [1..2,1..2] of integer;
```

```
B: array [1..3] of integer;
```

```
i,j:integer;
```

**Begin**

```
for i:=1 to 3 do Readln (B[i]);
```

```
for i:=1 to 2 do
```

<sup>1</sup> Baxılan misalda 45

```

        for j:=1 to 2 do
            Readln (A [i, j]);
        For i:=1 to 3 do writeln (B [i]);
        For i:=1 to 2 do
            begin
                For j:=1 to 2 do
                    Write (A[i, j]:4);
                Writeln;
            End;
        End.

```

Elementləri müəyyən qayda ilə təyin olunan və bir sıra xüsusi tip massivlərin elementləri mənimsətmə operatoru vasitəsilə də daxil oluna bilər. Məsələn, vahid matrisin elementləri aşağıdakı operatorlar vasitəsilə təyin oluna bilər.

```

var
    A: array [1..4,1..4] of integer;
    i,j:integer;
    Begin
        for i:=1 to 4 do
            begin
                Writeln;
                for j:=1 to 4 do
                    begin
                        if i=j then A [i, j]:=1
                        else
                            A [i, j]:=0;
                            Write (A[i, j], ' ');
                    end;
                end;
            end;
        End.

```

Qeyd edək ki, **Readln**, **Read**, **Write**, **Writeln** prosedurları ilə massivin tək bir elementini və ya bir neçə elementini də həm daxil etmək, həm də çıxışa vermək olar. Məsələn,

```

    Read (B [2]); writeln (A[2,3], A[4,5]);

```

### 2.12.1. Massivlər üzərində əməllər

Massivlər üzərində aşağıdakı əməliyyatlar aparmaq olar:

- Massivləri müqayisə etmək. Bu halda müqayisə edilən massivlər eyni strukturlu olmalıdır, yəni indekslərin tipi, elementlərin sayı və tipi eyni olmalıdır. Müqayisə əməliyyatı massivlərin ekvivalentliyini və ya fərqli olmasını təyin edir. Massivin ekvivalentliyi “=” operatoru vasitəsilə təyin edilir. Məsələn,  $A=B$ . A massivin hər bir elementinin qiyməti B massivin uyğun elementinin qiymətinə bərabər olduqda, əməliyyatın nəticəsi true qiyməti alır. Massivin

fərqliliyi isə “ $\diamond$ ” operatoru vasitəsilə təyin edilir. Məsələn,  $A \diamond B$  massivinin heç olmasa bir elementinin qiyməti müqayisə olunan digər massivinin uyğun elementinin qiymətindən fərqli olduqda əməliyyatın nəticəsi true qiyməti alır.

- Massivə digər massivi mənimsətmək. Bu əməliyyat mənimsətmə operatoru “:=” vasitəsilə yerinə yetirilir. Məsələn,  $A:=B$ . Bu halda B massivinin bütün elementinin qiymətləri A massivinin uyğun elementlərinə mənimsədilir. B massivi dəyişməz qalır.

- Ədədi massivlər üzərində hesabi əməliyyatlar. Massivləri toplamaq, qüvvətə yüksəltmək, onların hasilini tapmaq, transponirə etmək olar. Məsələn, matrisin transponirə edilməsi proqramına baxaq

```

Program P_34;
Const n=3;
Var
  i, j, Lb, Rb: Integer;
  a, b : Array[1..n, 1..n] Of Integer;
Begin
  WriteLn( 'A-matrisi' );
  For i:=1 to n do Begin
  For j:=1 to n do Begin
    read (a[i, j] );
    b[j, i]:= a[i, j];
    Write(a[i, j]:5)
  End;
  WriteLn;
End;
  Writeln;
  WriteLn( 'B-matrisi' );
  For i:=1 to n do
  Begin
    For j:=1 to n do Write(b[i, j]:5); WriteLn
  End;
End.

```

Massivlərin nizamlanması. Massivlərin nizamlanması riyaziyyatda ən çox istifadə olunan əməliyyatlardan biridir. Bir neçə nizamlama üsulu işlənmişdir. Aşağıdakı misallarda elementləri təsadüfi ədədlər olan seçmə və yerdəyişmə üsulu ilə matrislərin nizamlama proqramları verilmişdir:

Misal 1.

```

Program P_48;
Var
  a, b: Array[1..100] Of Integer;
  c, d, i, j, k, n, Min: Integer;
Begin
  ReadLn(n);
  ReadLn(c, d);
  WriteLn('Verilen massiv ');
  For i:= 1 To n Do
  Begin

```



```

        a[i]:= Random(d - c) + c;
    Write(a[i]:5)
    End;
WriteLn;
For i:= 1 To n Do
    Begin
        Min:=a[1];
For j:= 1 To n Do
    If a[j]<= min Then
Begin
    Min:=a[j];
    k:= j
    End;
        b[i]:= a[k];
        a[k]:= 32767;

    End;
WriteLn('Nizamlanmis massiv');
For i:= 1 To n Do Write(b[i]:5);
WriteLn;
End.

```

### Misal 2

```

    Program P_49;
    Var a: Array[1..100] Of Integer;
        c, d, i, p, k, n: Integer;
Begin
Randomize;
    ReadLn(n);
ReadLn(c, d);
WriteLn('Verilmis massiv ');
For i:= 1 To n Do
    Begin
        a[i]:= Random(d - c) + c;
        Write(a[i]:5)

    End;
WriteLn;
Repeat
    k:= 0;
1 To n-1 Do
    If a[i] > a[i + 1] Then
        Begin
            p:= a[i];
            a[i + 1];
            a[i + 1]:= p;
            k:= k + 1

        End;
Until k = 0;
WriteLn('Nizamli massiv');
For i:= 1 To n Do Write(a[i]:5);

```

```
WriteLn;
End.
```

## 2.12.2. Sabit massivlər

Sabit massivlər sabitin adını bildirən identifikatordan, iki nöqtədən, əvvəlcə təyin olunmuş hər hansı bir massiv tipindən, bərabərlik işarəsindən və massivin elementlərinin siyahısından ibarətdir. Elementlərin siyahısı dairəvi mötərizə daxilində yazılır. Elementlər bir-birindən vergüllə ayrılır. Sabit massivlərin elementləri fayl<sup>1</sup> və dinamik<sup>2</sup> tipdən başqa istənilən tipə aid ola bilərlər. Sabit massivlərin elementlərinin qiymətləri tək-tək və ya bütün simvolların tam sətri şəklində verilə bilər. Belə ki, aşağıdakı iki yazılış ekvivalentdir:

```
Const
D:array [0..5] of char=('0', '1', '2', '3');
```

və ya

```
Const
D : array [0..5] of char='0123';
```

Çoxölçülü sabit massivlər də oxşar olaraq təyin edilir:

Məsələn,

```
Type
KBAD= array [0..1, 0..1] of integer;
Const
KBAD1: KBAD=((0,1), (2,3));
```

## 2.13. Yazılar

Yazılar verilənlərin strukturlaşmış növü olub, sonlu sayda müxtəlif tipli elementlərdən ibarətdir. Massivlərdən fərqli olaraq yazılarda eyni bir obyektə aid olan müxtəlif tipli verilənlər qruplaşır. Yazı tipinin elan olunması **record** işçi sözü ilə başlayıb **end** xidməti sözü ilə qurtarır. Bunların arasında yazının elementlərinin siyahısı gəlir. Elementlərin siyahısına yazının sahələri də deyilir. Hər bir sahə identifikatordan və onun tipindən ibarətdir. Başqa sözlə yazı tipini elan etmək üçün aşağıdakı yazılış formasından istifadə edilir:

```
type
<tipin adı>=record
<sahə_identifikatoru_1>:<sahənin tipi>;
. . .
<sahə_identifikatoru_n>:<sahənin tipi>;
end;
```

Yazı tipi elan olunduqdan sonra yazı tipli dəyişənlər aşağıdakı yazılış formatı əsasında elan olunur.

<sup>1</sup> Bax: Fayllar

<sup>2</sup> Bax: Göstəricilər

**Var**

```
<yazı_identifikatoru>:<tipin adı>;
```

Məsələn, ədəbiyyatın bibliografik təsvirini əks etdirən `Biblio` adlı yazı tipi və kitab, məqalə adlı dəyişənləri aşağıdakı formada elan etmək olar:

**type**

```
Biblio=record
Avtor:string[50];
Serlohve:string[100];
Nesr_ili:byte;
Nesr_yeri:string[20];
Nesriyyat:string[30];
Cild:byte;
Sehife:byte;
end;
```

**Var**

```
kitab,meqale:Biblio;
```

Yazılar üçün tələb olunan yaddaşın həcmi sahələrin uzunluqlarını toplamaqla müəyyən etmək olar.

Yazının sahələrinə müraciət etmək üçün aşağıdakı yazılış formatından istifadə olunur<sup>1</sup>:

```
<yazı_identifikatoru>.<sahə_identifikatoru>
```

Yazının sahəsinə müraciət etməklə sahələrə qiymətlər mənimsətmək, sahələrin qiymətlərini daxil və xaric etmək, sahələri ifadələrdə istifadə etmək olar. Məsələn, `Kitab` yazısının sahələrinə müraciət etməklə kitabın bibliografik yazısını çap etmək olar:

**Program Bibliografik\_kataloq;**

**type**

```
Biblio=record
Avtor:string[50];
Serlohve:string[100];
Nesr_ili:integer;
Nesr_yeri:string[20];
Nesriyyat:string[30];
Cild:byte;
Sehife:byte;
```

**end;**

**Var**

```
i:byte;
Kitab:Biblio;
Biblio_yazi:string[200];
```

**Begin**

```
Readln(Kitab.avtor);
```

<sup>1</sup> Başqa sözlə yazının və sahə identifikatorları bir-birindən nöqtə ilə ayrılmış şəkildə yazılmalıdır.

```

    Readln(Kitab.Serlohve);
    Readln(Kitab.Nesr_ili);
    Biblio_yazi:= Kitab.avtor+' '+Kitab.Serlohve;
    Writeln (Biblio_yazi);

```

End.

Praktikada eyni tipli yazılardan istifadə etdikdə elementləri yazılar olan massivlərdən istifadə etmək əlverişli olur. Məsələn, 10 ədəd kitabın biblioqrafik təsvirini daxil etmək və çap etmək tələb olunarsa elementləri Biblio yazı tipi olan massivdən istifadə etmək olar.

```

Program Biblioqrafik_kataloq;
type
Biblio=record
Avtor:string[50];
Serlohve:string[100];
Nesr_ili:integer;
Nesr_yeri:string[20];
Nesriyyat:string[30];
Cild:byte;
Sehife:byte;
end;
Var
    i:byte;
    Book:array[1..10] of Biblio;
    Bib_yazi:array[1..10] of string[200];
Begin
for i:=1 to 10 do begin
    Readln(Book[i].avtor);
    Readln(Book[i].Serlohve);
    Readln(Book[i].Nesr_ili);
Bib_yazi[i]:=          book[i].avtor+' '+book[i].Serlohve;
    Writeln (Bib_yazi[i]);
End;End.

```

### 2.13.1. Birləşdirmə operatoru

Yazı identifikatoru və sahə identifikatoru çoxsimvollar olduqda yazı sahələrinə müraciəti təmin edən çox uzun düzəltmə adlar yaranır və yazı identifikatoru bir neçə dəfə təkrar olunur. Buna müvafiq olaraq proqram mətni xeyli böyüyür. Proqramın mətnini azaltmaq və onu kompakt şəkildə yazmaq üçün birləşdirmə operatorundan istifadə olunur. Bu operatorun yazılış formatı aşağıdakı kimidir:

```
with <yazı növlü> do <operator>;
```

Birləşdirmə operatorunda yazı tipli dəyişənin adını bir dəfə göstərməklə onun sahələrinin adları ilə adi dəyişən kimi işləmək olar. Məsələn, kitabın biblioqrafik təsvirini aşağıdakı proqram vasitəsilə də daxil etmək və çap etmək olar.

```
Program Biblioqrafik_kataloq;
```

```

type
  Biblio=record
Avtor:string[50];
Serlohve:string[100];
Nesr_ili:integer;
Nesr_yeri:string[20];
Nesriyyat:string[30];
Cild:byte;
Sehife:byte;
end;
Var
  Kitab: Biblio;
  Biblio_yazi:string[200];
Begin
With kitab do
Begin
  Avtor:='Qurbanov Azad Isa o.';
  Serlohve:='Informatika';
  Nesr_ili:=2009;
Biblio_yazi:= avtor+' '+Serlohve;
End;
  Writeln (Biblio_yazi);
End.

```

Turbo Pascal dilində bir-birinə daxil olan yazılardan da istifadə etmək olar. Bu halda yazının sahəsi yenə də yazı ola bilər və uyğun birləşdirmə operatoru aşağıdakı kimi yazılır:

```
with R1, R2, ..., Rn do S;
```

burada  $R_1, R_2, \dots, R_n$  yazı növlü dəyişənlərdir.  $S$  isə Turbo Pascal dilinin operatorlarıdır. Əgər  $R_1, R_2, \dots, R_n$  dəyişənlərində eyni adlı olan varsa, onda onlardan axırıncısı  $S$  operatorunda olan uyğun tipə birləşdirilir.

## 2.13.2. Variantla olan yazılar

Variant operatorundan istifadə etməklə bir neçə yazını bir yazı şəklində birləşdirmək olar. Belə yazıya birləşmiş yazılar və ya variantla olan yazılar deyilir.

Variantla olan yazılar iki hissədən ibarət olur:

- ümumi hissə;
- variantla olan hissə.

Ümumi hissədə yazının təsvir etdiyi obyektlər müəyyənləşdirilir. Variantla olan hissə **Case** operatoru vasitəsilə təyin edilir. Bu hissədə xüsusi sahə olan əlamət sahəsinin köməyi ilə verilmiş anda hansı variantın seçilməsi müəyyən edilir.

Variantla olan yazıların ümumi yazılış formatı aşağıdakı kimidir:

**Type**

```

<tip_identifikatoru>=record
<ümumi hissə>;

```

```

case<əlamət sahəsi>: <tipin adı> of
<l-ci sabit>: (sahə,...:tip);
... ..
<n-ci sabit>: (sahə,...:tip)
end;

```

Məsələn, Fiqur adlı birləşmiş yazı yaradaq. Onun təsviri aşağıdakı kimi

olar:

**type**

```

koordinat=record
absis, ordinat:real;
end;

```

**var**

```

forma=(nöqtə, çevrə, düz_xətt);
figur=record
case forma of
nöqtə:(vəziyyəti: koordinat);
çevrə:(mərkəz: koordinat, r:real)
end;

```

Figur yazısının elementlərinin aşağıdakı kimi çıxışa vermək olar:

**With** figur **do**

```

    case forma of
nöqtə: write (nöqtə.Absis, nöqtə.Ordinat);
        düz_xətt: write (A,B,C);
çevrə: write (mərkəz.Absis, mərkəz.Ordinat,r)
    end;

```

### 2.13.3. Sabit yazılar

Sabit yazıların təyini sabitin adından, iki nöqtədən, əvvəlcə təyin olunmuş yazı tipindən, bərabərlik işarəsindən və dairəvi mötərizənin içərisində yazılan sabitlərin qiymətlərindən ibarətdir. Sabitlərin qiymətləri bir-birindən nöqtə-vergüllə ayrılır. Məsələn,

**Const**

```

<yazı identifikatoru>:=(<sahə identifikatoru>:<qiymət>)

```

Məsələn,

**type**

```

Point = record
x,y: real;

```

**end;**

```

Vector = array[0..1] of Point;

```

**end;**

**const**

```

Origin: Point = (x: 0.0; y: 0.0);

```

```

Line: Vector = ((x: -3.1; y: 1.5), (x: 5.8; y: 3.0));

```

## 2.14. Prosedur və funksiyalar

Hər hansı bir məqsədə çatmağı reallaşdıran operatorlar qrupunun proqramın bir neçə yerində dəyişməz olaraq təkrarlanması məsələsi praktikada tez-tez qarşıya çıxır. Proqramda funksiya və prosedurlardan istifadə etməklə bu problemi evvektif şəkildə həll etmək olar.

Prosedur – proqramda müstəqil və proqram daxilində unikal ada malik olan proqram bloku olub, müəyyən əməliyyatların yerinə yetirilməsini təmin edir. Prosedurların adından istifadə etməklə ona proqramın müxtəlif yerlərindən müraciət etmək olar. Prosedurun adı ifadədə iştirak edə bilməz.

Prosedurlar iki qrupa bölünür:

1. İstifadəçi tərəfindən yaradılan;
2. Modullarda verilən.

Prosedurlar başlıqdan və gövdədən ibarətdir. Prosedurun başlığı **Procedure** sözündən, prosedurun adından və dairəvi mötərizənin içərisində yazılan formal parametrlərdən ibarətdir. Hər bir formal parametrin tipi göstərilməlidir. Prosedurun gövdəsi proqramın quruluşuna oxşardır. Başqa sözlə prosedurun yazılış formatı aşağıdakı kimidir:

```
Procedure <adı> (<parametrlər>);
    <Təsvir etmə bölməsi>
begin
    <operatorlar bölməsi>
End;
```

Əgər zərurət yoxdursa, onda prosedurun başlığında formal parametrlərin siyahısı verilmir.

Proqramda prosedura müraciət etmək üçün prosedurun adını və dairəvi mötərizənin içərisində faktiki parametrlərin siyahısını vermək lazımdır. Faktiki parametrlər bir-birindən vergüllə ayrılmalıdırlar. Prosedura müraciət olunan zaman faktiki və formal parametrlər arasında qarşılıqlı birqiymətli əlaqə yaranır. Faktiki parametrlər vasitəsilə lazım olan dəyişənlərin qiymətləri prosedura ötürülür və ya prosedurda alınan nəticələr əsas proqrama qaytarılır. Faktiki və formal parametrlərin sayı, növü və yazılma ardıcılıqları bir-birinə uyğun olmalıdır. Bu uyğunluq pozularsa, onda proqramda kəsilmə halı baş verəcək. Prosedurda 3 növ formal parametrdən istifadə olunur: parametr - qiymət, parametr - dəyişən, parametr – sabit. Prosedurda parametr - qiymətə əsas proqramdan faktiki parametrlərin qiyməti ötürülür. Proqram yerinə yetirildikdə faktiki parametrin qiyməti hesablanaraq operativ yaddaşa yazılır. Prosedura müraciət zamanı isə operativ yaddaşdan onun surəti prosedura ötürülür və bu qiymət formal parametərə mənimsədir. Bu halda faktiki parametr parametr-qiymətinin tipinə uyğun ifadə də ola bilər.

Əgər prosedurdan əsas proqrama hər hansı dəyişənin qiymətini ötürmək tələb olunursa, onda həmin dəyişən prosedurun başlığında formal parametr – parametr - dəyişən kimi **var** sözü ilə birlikdə təsvir olunmalıdır. Bu halda faktiki

parametrlər ifadə ola bilməzlər. Məsələn,  $y = x^n$  funksiyasını hesablama proqramına baxaq.

```

program XN;
var
n:integer;
x,y:real;
procedure step1(m:integer; x:real;var y:real);
var
i:integer;
begin
y:=1;
for i:=1 to m do
y:=y*x;
end;
begin
        writeln (' x,n -i daxil edin') ;
readln (x,n);
If n=0 then y:=1
else
if n>0 then step1(n,x,y)
else
step1(n,1/x,y);
writeln (x:4:2,' ustu ',n:3,' dââîî ',y:4:2);
end.

```

Proqram yerinə yetirildikdə monitora  $x$  və  $n$  qiymətlərinin daxil edilməsi haqqında məlumat əks olunur. İstifadəçi  $x$  və  $n$ -in qiymətlərini daxil etdikdən sonra  $n$ -in qiyməti yoxlanılır.  $n = 0$  olduqda  $y = 0$  olur.  $n > 0$  olduqda  $y = x^n$  funksiyasını hesablamaq üçün **step1** proseduruna müraciət olunur. Bu zaman prosedurun  $m$ ,  $x$  formal parametrlərinin qiyməti  $n, x$  faktiki parametrlərin qiymətləri ilə əvəz olunur, hesablama aparılır və nəticə -  $y$  əsas proqrama qaytarılır. Bundan sonra çıxış proseduru vasitəsilə nəticə operativ yaddaşdan monitora göndərilir.

Parametr – sabit adından məlum olduğu kimi əsas proqramdan prosedura hər hansı sabitin ötürülməsini təmin edir. Parametr-sabit prosedurun başlığında **const** işçi sözü ilə birlikdə təsvir olunmalıdır.

Eyni bir proqramda bir neçə prosedur ola bilər. Bir neçə prosedur bir-birinin daxilində yerləşə bilər. Bu prosedurlara müraciət etmə qaydası yuxarıdan aşağıya doğru, yəni xarici prosedurdan daxili prosedura doğru olur.

Proqramlaşdırıcı tərəfindən təyin olunan funksiya başlıqdan və funksiyanın gövdəsindən ibarətdir. Başlıq **Function** sözündən, funksiyanın adından, dairəvi mötərizənin içərisində formal parametrlərin siyahısından və funksiyanın qiymətinin tipindən ibarət ola bilər. Prosedurlarda olduğu kimi funksiyalarda da formal parametrlər zəruri olduğu halda yazılır.

Yazılış formatı aşağıdakı kimidir:

```

Function <adı>(formal parametrlər): <tipi >;
<Təsvir etmə bölməsi>

```



```
begin
<operatorlar bölməsi>
end;
```

Funksiyaya müraciət etmə qaydası aşağıdakı kimidir:

<funksiyanın adı> (faktiki parametrlər);

Prosedurlarda olduğu kimi funksiyalarda da faktiki və formal parametrlərin sayı, tipi və yerləşmə ardıcılıqları bir-birinə uyğun olmalıdır. Məsələn,  $\frac{n!}{m!(n-m)!}$  qiymətini hesablamaq üçün k!-ın hesablanması funksiya altprogramından istifadə etmək məsləhətdir.

```
Program kombizon;
Var
n,m:integer;
c:real;

function F(k:integer):longint;
var y:longint;
    i:integer;
begin
    y:=1;
    for i:=1 to k do
        y:=y*i;
    F:=y
end;
begin
    readln (n,m);
    c:=f(n) / (f(m) - f(n-m));
    write(c);
end.
```

Funksiyanı prosedurdan fərqləndirən cəhətlər aşağıdakılardır:

- funksiyanın başlığında onun qiymətinin tipi göstərilir.
- prosedurlarda bir neçə dəyişənin qiyməti müraciət edən proqrama qaytarıldığı halda funksiyada ancaq bir qiymət-funksiyanın qiyməti müraciət edən proqrama qaytarılır.
- funksiyanın başlığında göstərilən tip funksiyanın hesablamadan alınan qiymətinin tipinə uyğun olmalıdır.
- funksiyanın gövdəsində ən azı bir mənimləmə operatoru olmalıdır ki, onun vasitəsilə funksiyanın adına funksiyanın qiyməti – ifadə mənimlənilir.

## 2.15. Prosedur tipli dəyişənlər

Turbo Pascal-da klassik Pascal dilindən fərqli olaraq prosedur tipli dəyişənlər təyin edilmişdir. Məhz onun vasitəsilə funksiya və prosedurların adları digər

prosedur və funksiyalarda formal parametr kimi istifadə olunur. Prosedur tipi aşağıdakı kimi elan olunur:

**Type**

```
<identifikator>=procedure(<formal parametrlər>);  
    <Təsvir etmə bölməsi>
```

və ya

**Type**

```
<identifikator>=function(<formalparametrlər>):  
<tip>;
```

Qeyd edək ki, prosedur tipindən istifadə etmək üçün aşağıdakı qaydalara əməl etmək zəruridir:

- Proqramın səhvsiz yerinə yetirilməsi üçün {\$F+} kompilyator açarı verilməlidir;
- Prosedur tipinin təyin etdiyi prosedur və funksiyalar standart tipli olmamalıdır;
- Prosedur tipinin təyin etdiyi prosedur və funksiyalar digər prosedur və ya funksiyanın təsvir etmə bölməsində elan olunmamalıdır.

Məsələn, vurma cədvəlinin yaradılması proqramına baxaq.

**Program P\_22;**

**Uses Crt;**

**Type f = Function(x,y: Integer): Integer;**

**Var a,b: Integer;**

```
    {$f+} Function Mult(x,y: Integer): Integer;           Begin  
Mult:= x * y;  
end;  
    {$f-}  
    Procedure Print(a,b: Integer; Oper: f);  
    Var i, j: Integer;                                   Begin  
    For i:=1 To a Do  
        Begin  
            For j:= 1 To b Do Write(Oper(i, j):5);  
            WriteLn  
        End;  
    WriteLn  
    End;  
  
    Begin  
    Write('Setir ve sutunların sayını daxil edin');  
    ReadLn(a, b);  
    Print(a, b, Mult);  
    End.
```

## 2.16. Rekursiyalar

Əgər prosedur və ya funksiyanın gövdəsində cari prosedura və ya dövrü olaraq funksiya müraciət olunarsa, onda həmin prosedura və ya funksiya

rekursiv prosedur və ya funksiya deyilir<sup>1</sup>. Qeyd edək ki, bir çox riyazi funksiyaları rekursiv olaraq təyin etmək mümkündür. Məsələn  $y = x^m$  funksiyasının hesablanması rekursiv funksiya vasitəsilə yerinə yetirilə bilər.

```

Var
  m: integer;
  x,y:real;
Function XN (X:real;N:integer):real;
begin
if N=0 then XN:=1
else
XN:=X*XN(X,N-1);
end;
begin
  readln (m);
  readln(x);
y:=XN(x,m);
  Write('y=',y);
end.

```

Proqramdan görüldüyü kimi,  $y = x^m$  funksiyasının hesablanması üçün **XN** funksiyasında m qədər özünə müraciət olunur.

### 2.16.1. Forward direktivi

Praktikada ola bilər ki, birinci altproqramdan, müraciət anına qədər hələ təyin olunmayan ikinci altproqrama müraciət olunsun. Belə vəziyyət əsasən vasitəçi rekursiyalarda – prosedurlar və funksiyalar qarşılıqlı olaraq bir-birinə müraciət etdikdə baş verir. Bu problemi həll etmək üçün uyğun prosedur və funksiyaları **Forward** direktivi ilə əvvəlcədən elan etmək lazımdır. Bu məqsədlə prosedur və ya funksiyanın başlığından sonra **Forward** sözü yazılır. Məsələn,

```

var
  n,l:integer;
procedure Turqut(i: integer; var l:integer); forward;
procedure Azad(i: integer; var l:integer);
begin
  Turqut(i-1,l);
end;
procedure Turqut(i: integer; var l:integer);
begin
  l:=i*5;
end;
begin
  read(n);
  Azad(n,l);
  write(l)
end.

```

<sup>1</sup> Başqa sözlə funksiya və ya prosedurun özündən özünə müraciət olunduqda onda həmin prosedura və ya funksiya rekursiv prosedur və ya funksiya deyilir.

Proqram yerinə yetirildikdə əsas proqramdan Azad proseduruna müraciət olunur. Azad prosedurundan isə öz növbəsindən Turqut proseduruna müraciət olunur. Turqut proseduru Azad prosedurundan sonra gəldiyinə görə o, əvvəlcədən **Forward** direktivilə elan olunmuşdur.

## 2.17. Fayl tipli dəyişənlər. Fayllarla əməliyyatlar

Böyük həcmli verilənlərlə işlədikdə fayl tipli verilənlərdən istifadə etmək məqsədəuyğundur. Əks halda hər dəfə ilkin verilənləri klaviaturadan daxil etmək lazım gəlir. Bu isə çox yorucu iş olub, xeyli vaxt itkisi tələb edir. Fayl tipli verilənlər konkret informasiya daşıyıcısında (məsələn, bərk disklərdə, disketlərdə, CD, DVD disklərdə, fleş yaddaşında və s.) proqram və ya istifadəçi tərəfindən fayl şəklində saxlanılır. *Fayl-mətn və ikilik formatda verilənləri saxlamaq üçün ada malik xarici yaddaş sahəsidir.* Hər bir fayl unikal ünvana malik olur. Fayla verilənlər klaviatura vasitəsilə istifadəçi tərəfindən və ya proqram yerinə yetirildikdə əməliyyatların nəticəsi olaraq avtomatik daxil oluna bilər. Bu səbəbdən faylları giriş və çıxış faylı kimi fərqləndirirlər. Proqram giriş faylından ilkin verilənləri alır, çıxış faylına isə yerinə yetirdiyi əməliyyatın son nəticəsini yazır.

Turbo Pascalda fayllarla işləmək üçün fayl tipli dəyişəndən istifadə olunur. Fayl tipli dəyişən ifadələrdə istifadə oluna bilməz və onun vasitəsilə fayla müraciət olunur. Faylda saxlanılan verilənlərin tipinə uyğun olaraq proqramda fayl dəyişəni fərqli elan olunur. Belə ki, mətn fayllarına uyğun fayl dəyişəni aşağıdakı kimi elan olunur:

**Var**

**<faylın adı>: Text;**

İkilik formatlı fayllar öz növbəsində tipli və tipsiz növ fayllara bölünür<sup>1</sup> və hər bir növ fərqli formada elan edilir. Tipsiz fayllara uyğun fayl dəyişəni aşağıdakı kimi elan olunur:

**Var**

**<faylın adı>: file;**

Tipli fayllara uyğun fayl dəyişəni isə aşağıdakı kimi elan olunur:

**Var**

**<fayl adı>: file of <elementlərin tipi>;**

Proqramda fayl tipli dəyişən elan olunduqdan sonra onunla informasiya daşıyıcısında yerləşən faylın ünvanı arasında birqiymətli əlaqə yaradılmalıdır. Bunun üçün standart **Assign (<fayl tipli dəyişən>, <faylın ünvanı>)** prosedurundan istifadə olunur. Bu prosedur yerinə yetirilən andan başlayaraq fayl tipli dəyişən üzərində aparılan bütün əməliyyatlar, prosedurda ünvanı göstərilmiş fayl üzərində aparılan əməliyyatlarla ekvivalentdir. Məsələn,

<sup>1</sup> Tipli fayllar eynitipli verilənləri, tipsiz fayllar sadəcə olaraq baytlar toplusu olub ixtiyari tipli verilənləri saxlayır.

**Assign (f, 'c:\azad\aysel.txt')**

bu yazılışda f – fayl dəyişəni, 'c:\azad\aysel.txt' isə müraciət olunan faylın ünvanıdır.

Faylla işləmək üçün faylı açmaq lazımdır. İnformasiya daşıyıcısında artıq mövcud olan faylı açmaq üçün **Reset(<fayl tipli dəyişən>, <Recsize>)** prosedurundan istifadə olunur. Bu yazılışda **<Recsize>** parametri məcburi olmayan parametr olub, tipsiz fayllar üçün oxunan blokun baytlarla ölçüsünü bildirir. Bu parametr verilmədikdə kompilyator avtomatik olaraq blokun həcmnin 128 bayta bərabər olduğunu qəbul edir. Qeyd edək ki, prosedurda göstərilən ada malik fayl olmadıqda, proqramın yerinə yetirilməsi dayanır və kompilyator giriş-çixış xarakterli səhv haqqında məlumat verir.

Məsələn,

**Reset (f) ;**

Turbo Pascal dilində nəzərdə tutulmuş digər **Rewrite (<fayl dəyişəni>, <Recsize>)** prosedur isə ünvanı **Assign** prosedurunda elan olunan yeni fayl yaradaraq onu açır. Bu yazılışda da **<Recsize>** parametri məcburi olmayan parametr olub, tipsiz fayllar üçün oxunan blokun baytlarla ölçüsünü bildirir. Eyni qaydada, bu parametr verilmədikdə kompilyator avtomatik olaraq blokun həcmnin 128 bayta bərabər olduğunu qəbul edir. Fayl artıq mövcud olduqda, prosedur icra olunduqda həmin fayl pozulur. Qeyd edək ki, **Reset(<fayl tipli dəyişən>, <Recsize>)** proseduru verilənlərin fayldan yalnız oxunmasını, **Rewrite (<fayl tipli dəyişən>, <Recsize>)** proseduru isə verilənlərin fayldan həm oxunmasını, həm də fayla verilənlərin yazılmasını təmin edir.

Mətn fayllarının sonuna yeni verilənləri əlavə etmək üçün faylın açılması **Append (<fayl dəyişəni>)** prosedur vasitəsilə təmin olunur.

Faylı açıqdan sonra fayldan verilənləri oxumaq, fayla verilənlər yazmaq, əlavə etmək, verilənləri redaktə etmək olar.

Fayldan verilənləri oxumaq üçün **Read(<fayl tipli dəyişən>, <m1>, <m2>, ...)** və **Readln(<fayl tipli dəyişən>, <m1>, <m2>, ...)** prosedurlarından istifadə etmək lazımdır. Bu yazılışlarda **<m1>, <m2>, ...** dəyişənlərinin qiymətləri fayldan oxunur. **Read(<fayl tipli dəyişən>, <m1>, <m2>, ...)** proseduru verilənlərin ardıcıl oxunmasını təmin edir. **Readln (<fayl tipli dəyişən>, <m1>, <m2>, ...)** proseduru yalnız mətn faylından verilənlərin yeni sətirə keçməklə oxunmasını təmin edir.

Fayla əməliyyatların nəticəsini yazmaq üçün **Write (<fayl tipli dəyişən>, <n1>, <n2>, ...)**, və ya **Writeln (<fayl tipli dəyişən>, <n1>, <n2>, ...)** prosedurdan istifadə etmək lazımdır. Bu yazılışda **<n1>, <n2>, ...** dəyişənlərinin qiymətləri fayla yazılır. **Writeln (<fayl tipli dəyişən>, <n1>, <n2>, ...)** proseduru yalnız mətn faylına verilənlərin yazılmasını təmin edir.

Mətn fayllarında verilənlərin oxunması ardıcıl yolla həyata keçirilir. Bu zaman kursurun cari mövqeyini təyin etmək üçün aşağıdakı standart funksiyalar nəzərdə tutulmuşdur:

1.) **Eof** (<fayl tipli dəyişən>) funksiyası verilənlərin oxunması və ya yazılması prosesində faylın sonuna çatdıqda **True** qiymət, əks halda isə **False** qiymət alır.

2.) **Eoln** (<fayl tipli dəyişən>) funksiyası verilənlərin oxunması və ya yazılması prosesində sətirin sonuna çatdıqda **True** qiymət, əks halda isə **False** qiymət alır.

Adətən, **eof** funksiyası **While...end** dövr operatorunun daxilində istifadə olunur və mətn faylının bütün sətirlərinin oxunması üçün, **eoln** funksiyası isə sətirdən bütün simvolların oxunması üçün istifadə olunur. Faylla əməliyyat yerinə yetirildikdən sonra o, bağlanılmalıdır. Bunun üçün standart **Close** (<fayl tipli dəyişən>) proseduru nəzərdə tutulmuşdur. Prosedur faylı bağlayır, yəni fayl tipli dəyişənin təyin etdiyi informasiya daşıyıcısında olan faylla əlaqəni kəsir. Məsələn, **BIBLO** adlı faylı **c:\** diskində təşkil etmək lazımdır.

```

Program Biblio;
Type
Books=record
Nomer:integer;
Autor:string [46];
Title: string [60];
ISBN: string [16];
End;
Var
BooksFile: text;
Books1 : Books;
Begin
Assign (Booksfile,'c:\Biblo.txt');
Rewrite (Booksfile);

readln (Books1.Nomer);
writeln(BooksFile,Books1.Nomer);
readln (Books1.Autor);
writeln (BooksFile, Books1.Autor);
readln (Books1.Title);
writeln (BooksFile,Books1.Title);
readln (Books1.ISBN);
writeln (BooksFile,Books1.ISBN);
Close (BooksFile);
End.

```

Qeyd edək ki, **Read** və **Write** prosedurları vasitəsilə tipsiz fayllara verilənlər 128 baytdan ibarət olan bloklarla yazılır və oxunur. Tipsiz fayllardan verilənlərin daha sürətli oxunması üçün əlavə olaraq **BlockWrite**(var f:file; var buf; count:Word; result:Word), verilənləri daha sürətlə yazmaq üçün isə **BlockWrite**(var f:file; var buf; count:Word; result: Word) prosedurlarından istifadə olunur. Buf - parametri müvafiq olaraq oxunan və yazılan verilənə uyğun ixtiyari dəyişən, count – oxunan və yazılan blokların sayını, result – oxunan və ya yazılan yazıların sayını bildirən parametrlərdir.

Məsələn,

```

Program Turqut;
Var

```

```

    OldFile, NewFile : file;
    V : array [1..128,1..200] of byte;
    Result : integer;
    OldName, NewName : string [14];
Begin
    writeln ('İlkin faylın adını verin');
    readln (OldName);
Assign (OldFile, OldName);
    Reset (OldFile);
    writeln ('Yeni faylın adını verməli:');
    readln (NewName);
Assign (NewFile, NewName);
Rewrite (NewFile);
    Repeat
        BlockRead (OldFile, V, 200, Result);
        BlockWrite (NewFile, V, Result);
Until Result=0
    Close (OldFile);
    Close (NewFile)
End.

```

Bu proqram bir faylın digər fayla sürətlə köçürülməsini təmin edir.

Mətn faylı istisna olmaqla tipli və tipsiz fayllara birbaşa üsulla da müraciət etmək olar. Bu isə diskdə saxlanılan informasiyadan operativ olaraq istifadə edilməsini təmin edir.

Birbaşa müraciəti təşkil etmək üçün `seek(F, n)` standart prosedurundan istifadə edilir. Bu prosedurda `n` arqumenti `F` fayl dəyişənilə təyin olunan faylın elementinin sıra nömrəsini təyin edir. `n` tam qiymətlər alır və bu ədəd faylın elementlərinin sayından çox olmamalıdır. Faylın elementləri 0-dan başlayaraq nömrələnir. Prosedur yerinə yetirildikdə cursor faylın mövqeyini `n`-ci elementə doğru yerini dəyişir və həmin elementin oxunması, redaktəsi icra oluna bilər, `n`-ci mövqedən fayla yeni verilən yazıla bilər. Praktikada birbaşa müraciət üçün bu prosedurdan başqa daha 2 standart funksiya istifadə olunur:

1. `FilePos(F)`. Funksiya cari anda `F` faylında cursorun yerləşdiyi elementinin nömrəsini təyin edir.
2. `FileSize(F)`. Funksiya `F` faylının ölçüsünün -faylın elementlərinin sayını təyin edir. Adətən bu funksiya faylda elementin olmasını yoxlamaq üçün istifadə edilir. Əgər `FileSize(F)=0` isə onda fayl boşdur, yəni onda heç bir element yoxdur, əks halda faylda elementlər vardır.

Məsələn ,

```

type tf=file of integer;
procedure sortpuz(name:string);
var f:tf;
    i,nkz:longint;
    r1,r2:integer;
    flag:boolean;
begin
    assign(f,name);
    reset(f);
    nkz:=filesize(f);

```

```

repeat
  flag:=false;
  for i:=0 to nkz-2 do
    begin
      seek(f,i);
      read(f,r1,r2);
      if r2>r1 then
        begin
          seek(f,i);
          write(f,r2,r1);
          flag:=true
        end
      end;
    nkz:=nkz-1
  until not flag;
  close(f);
end;
begin
sortpuz('c:\azad.txt')
end.

```

Proqram tam ədədlərdən ibarət faylda nizamlama əməliyyatı aparır.

## 2.18. Modullar

Turbo Pascal-da proqramın həcminə müəyyən məhdudiyyət vardır. Belə ki, kompilyator yalnız o proqramları translyasiya edə bilər ki, bu əməliyyat zamanı maşın kodunun həcmi 64 Kbayt-dan çox olmasın. Bu səbəbdən böyük həcmli proqramları bir neçə proqram fraqmentlərinin - modulların toplusu şəklində yaratmaq lazım gəlir. *Modul müstəqil kompilyasiya olunan proqram vahididir.* Modulun mətni “.pas” genişlənməsinə malik faylda, kompilyasiya olduqda isə o, “.tpu” genişlənməsinə malik xüsusi faylda saxlanılır. Əsas proqramdan **uses** işçi sözü vasitəsilə modulun həm mətninə və həm də onun kompilyasiya olmuş faylına müraciət olunur. Qeyd edək ki, modulun mətni və ya onun kompilyasiya olmuş faylı əsas proqramın yerləşdiyi cari kataloqda və ya istifadəçi tərəfindən, Turbo Pascal proqram mühitinin **Options** → **Directories** əmrini yerinə yetirdikdə açılan *Directories* dialoq pəncərəsinin (şəkl. 1) *Unit directories* sahəsində müəyyən olunan digər kataloqda yerləşə bilər.

Modul vasitəsilə həmçinin tətbiqi proqram kitabxanası yaradılır. Bu kitabxananın prosedur və funksiyalarına digər proqramlardan müraciət etməklə bir sıra məsələləri, o cümlədən, riyazi, statistik məsələləri həll etmək, kompyuter qrafikasının elementlərindən istifadə etmək olar.

Modul aşağıdakı struktura malikdir:

```

UNIT <ad>; {Modulun başlığı}
INTERFACE
<intrfeys hissə>
IMPLEMENTATION
<icra olunan hissə>
BEGIN

```



```
<insalyasiya hissəsi>
```

```
END.
```

Bu yazılışda <ad> -modulun adını bildirir. İnterfeys hissədə digər proqramlardan və modullardan müraciət olan elementlər – funksiyalar, prosedurlar, dəyişənlər, sabitlər, istifadəçi tipləri elan olunur. İcra olunan hissədə yalnız cari modulda istifadə olunan elementlər elan olunur və interfeys hissədə elan olunan prosedur və funksiyaların proqram kodları yazılır. İnsalyasiya sahəsində **begin** və **end** işçi sözləri arasında modulun icra olunan operatorları yerləşir. Modulda icra olunan operator olmadıqda onda **begin** sözü istifadə olunmur. Məsələn, aşağıda verilmiş **Esas** proqramda a və b kompleks ədədlərinin cəmini, fərqini, hasilini, qismətini hesablayan proqramda kompleks ədədlər üzərində hesablama əməliyyatlarını aparmağı təmin edən **Cmplx** modulundan istifadə edilmişdir. Modulun **AddC** proseduru ixtiyari iki kompleks ədədin cəmini, **SubC** proseduru fərqini, **MulC** proseduru hasilini, **DivC** proseduru isə cəmini hesablayır.

```
UNIT Cmplx;
INTERFACE
type complex = record
re, im:real
end;
Procedure AddC (x, y: complex; var z : complex) ;
Procedure SubC (x, y: complex; var z : complex) ;
Procedure MulC (x, y: complex; var z : complex) ;
Procedure DivC (x, y: complex; var z : complex) ;
const
c : complex = (re :0.1; im : -1);
IMPLEMENTATION
Procedure AddC; begin
z.re := x.re + y.re; z . im := x.im + y. im
end {AddC};
Procedure SubC;
begin
z.re := x.re - y. re ;
z.im := x.im - y.im
end {SubC};
Procedure MulC;
begin
z.re := x.re*y.re - x.im*y. im;
z.im := x.re*y.im + x.im*y.re
end {MulC};
Procedure DivC;
var
zz : real;
begin
zz := sqr(y.re) + sqr(y.im);
z. re := (x.re * y.re + x.im * y.im) / zz;
z.im := (x.re * y.im - x.im * y.re) / zz
end {DivC};
end.
Program Esas;
```

```

Uses Cmplx;
var
a, b, c : complex;
begin
a.re := 1; a.im := 1;
b.re := 1; b.im := 2;
AddC(a, b, c);
WriteLn('a+b=', c.re:5:1, c.im:5:1, 'i') ;
SubC(a, b, c) ;
WriteLn('a-b=', c.re:5:1, c.im:5:1, 'i') ;
MulC(a, b, c);
WriteLn('a*b=', c.re:5:1, c.im:5:1, 'i') ;
DivC(a, b, c);
WriteLn('a/b=', c.re:5:1, c.im:5:1, 'i') ;
end.

```

Turbo Pascal dilində bir sıra standart modullar- System, DOS, WinDos, Crt, WinCrt, Printer, WinPrn, Overlay, Strings Graph, Turbo3, Graph3, WinTypes, WinProcs, Win31, WinAPI nəzərdə tutulmuşdur.

## 2.19. Standart prosedur və funksiyalar

Standart prosedur və funksiyalar avtomatik qoşulan System moduluna aiddir Tətbiq olunma sahələrinə görə onları aşağıdakı kimi təsnifləşdirmək olar:

1. İdarəetmə prosedurları
2. Dinamik paylanma prosedurları
3. Çevirmə funksiyaları
4. Hesabi funksiyalar
5. Sıra prosedurları və funksiyaları
6. Sətir prosedurları və funksiyaları
7. Digər prosedur və funksiyalar

Aşağıdakı cədvəllərdə hər bir təsnifat qrupuna uyğun standart prosedur və funksiyaların siyahısı verilmişdir.

### İdarəetmə prosedurları.

Prosedurlar	Yerinə yetirdiyi əməliyyat
Continue	Dövr operatorunda idarəni dövrün sonuna ötürür.
Exit	Cari proqramdan avtomatik çıxışı təmin edir.
Halt (ExitCode:Word) :Word	Proqramın yerinə yetirilməsini dayandırır məcburi olmayan ExitCode:parametri proqramın dayanma kodunu müəyyən edir.
Break	Dövr operatorundan vaxtından əvvəl çıxışı təmin edir.

### Çevirmə funksiyaları

Funksiya	Təyinatı
Chr(X: Byte) :char	ASCII kodu x arqumentilə verilmiş simvolu təyin

Funksiya	Təyinatı
	edir.
Ord (X) : LongInt	X arqumentilə verilmiş ixtiyari sıralı tipin sıra nömrəsini təyin edir.
Round (R:Real) : LongInt	R dəyişənin qiymətini ən yaxın tam ədədə qədər yuvarlaqlaşdırır.
Trunc (R:Real) : LongInt	R həqiqi tipli dəyişənin qiymətinin kəsr hissəsini ataraq onu tam ədədə çevirir.

### Hesabi funksiyalar

Funksiya	Təyinatı
Abs (R; Real) : Real	x – arqumentinin modulunu hesablayır.
ArcTan (R:Real) : Real	x – arqumentinin arktangensini hesablayır.
Cos (R:Real) : Real	x – arqumentinin kosinusunu hesablayır.
Exp (R; Real) : Real	x – arqumentinin eksponensial qiymətini hesablayır.
Sin (R:Real) : Real	x – arqumentinin sinusunu hesablayır.
Sqr (R:Real) : Real	x – arqumentinin kvadratını hesablayır.
Sqrt (R:Real) : Real	x – arqumentinin kvadrat kökünü hesablayır.
Frac (R; Real) : Real	x-arqumentinin kəsr hissəsini müəyyən edir.
Ln (R:Real) : Real	x – arqumentinin natural loqarifmini hesablayır
Pi	$\pi$ <sup>1</sup> ədədini təyin edir.

### Sıra funksiyaları

Funksiya	Təyinatı
Odd (X) : Boolean	Byte tipli arqumentin qiymətinin cüt ədəd olmasını yoxlayır. Nəticə true və ya false olur.
Pred (X)	Ixtiyari sıralı tipli arqumentin aldığı qiymətdən əvvəlki qiyməti təyin edir.
Succ (X)	Ixtiyari sıralı tipli arqumentin aldığı qiymətdən sonrakı qiyməti təyin edir.

### Sıra prosedurları

Prosedurlar	Təyinatı
Dec (var X; DX: LongInt)	x arqumentinin qiymətini dx qədər, əgər dx parametri verilməyibsə, bir vahid azaldır.
Inc (var X; DX: LongInt)	x arqumentinin qiymətini dx qədər, əgər dx parametri verilməyibsə, bir vahid artırır.

### Sətir tipli prosedurlar<sup>2</sup>

<sup>1</sup>  $\pi = 3.141592653897932385$

<sup>2</sup> Sətir tipli prosedur və funksiyalar sətir tipli verilənlər üzərində əməliyyat aparmağı təmin edir.

<b>Prosedurlar</b>	<b>Təyinatı</b>
Delete (var S: String; Index, Count: Integer)	s sətindən Index mövqeyindən başlayaraq count sayda simvolu ləğv edir. Əgər count s sətirinin uzunluğundan böyükdürsə, onda s sətirində heç bir simvol silinmir. Əgər “index+count”-in qiyməti s-in uzunluğundan böyükdürsə, onda sətirin bütün simvolları axıra qədər silinir.
Insert (SubS: String; var S: String; Index: Integer)	s sətirinə onun index nömrəli simvolundan başlayaraq subs sətirini əlavə edir.
Str(X [: width [: Decimals]]; var S: String)	x arqumentinin ədədi qiymətini s sətir tipinə çevirir.
Val(S; String; var X; var Code: Integer)	s - arqumentinin sətir tipli qiymətini x – ədədi tipli qiymətə çevirir. Code –parametri çevirmə zamanı səhvin kodunu müəyyən edir.

### Sətir tipli funksiyalar

<b>Prosedurlar</b>	<b>Təyinatı</b>
Concat(S1, S2, ..., SN): String	Arqumentlərlə təyin olunan sətirlərin birləşməsini təyin edir. Alınan sətirin uzunluğu 255-dən böyük olmamalıdır.
Copy(S: String; Index, Count: Integer): String	S sətindən index mövqeyindən başlayaraq count uzunluqlu sətir ayırır.
Length(S: String): Byte	S sətirinin uzunluğunu müəyyən edir.
Pos(SubS, S: String): Byte	Sbs sətirinin s sətirində hansı mövqedən yerləşdiyini müəyyən edir.
UpCase(c)	c arqumentilə təyin olunan simvolun yuxarı registrində yazılışını müəyyən edir.

### Fayl tipli funksiyalar<sup>1</sup>

<b>Prosedurlar və funksiyalar</b>	<b>Təyinatı</b>
ChDir(S: String)	Prosedur S – ünvanlı kataloqu cari kataloq kimi müəyyən edir.
Erase (var F)	Prosedur F faylını ləğv edir.
GetDir(D:Byte;var S:String)	Prosedur verilmiş D diskində S adlı kataloqu cari edir.

<sup>1</sup> Digər funksiyalar § 2.19 Fayl tipli dəyişənlər. Fayllarla əməliyyatlar paragrafında verilmişdir.

Prosedurlar və funksiyalar	Təyinatı
MkDir(S:String)	Prosedur cari kataloqda S altkataloqu yaradır.
Rename(var F,s)	Prosedur F faylının ünvanını dəyişdirir. Yeni ünvan s sətirində verilir.
Rmdir(S:String)	Prosedur boş S altkataloquunu ləğv edir.
FileSize(var F): LongInt	Funksiya tipli və tipsiz faylların ölçüsünü təyin edir.

### Digər prosedurlar

Prosedurlar	Yerinə yetirdiyi əməliyyat
Exclude (var S: set of T; I: T)	I elementini S çoxluğundan ləğv edir.
Include (var S: set of T; I: T)	I elementini S çoxluğuna əlavə edir.
Randomize	Cari sistem vaxtına uyğun təsadüfi ədədlər generatorunu aktiv edir.

### Digər funksiyalar

Funksiyanın adı	Yerinə yetirdiyi əməliyyat
Hi(X: Word): Byte	X arqumentinin böyük baytını müəyyən edir.
High(X)	X arqumentinin maksimal qiymətini müəyyən edir.
Lo(X: Word): Byte	X arqumentinin kiçik baytını müəyyən edir.
Low(X)	X arqumentinin maksimal qiymətini müəyyən edir.
Random (Range)	Təsadüfi ədədi müəyyən edir. Range parametri verilmədikdə [0;1] intervalında yerləşən, parametr verildikdə isə [0;Range-1] intervalında yerləşən təsadüfi ədədləri müəyyən edir.
SizeOf(X):	x arqumentinin baytını müəyyən edir.

Dinamik paylanma prosedur və funksiyaları dərsliyin Gostəricilər paraqrafında verilmişdir.

## 2.20. Turbo Pascal dilinin qrafiki imkanları

Turbo Pascal dilinin qrafiki imkanları GRAPH modulunun standart prosedurları vasitəsilə müvafiq qrafiki drayver yükəndikdə yerinə yetirilir. Qrafiki drayver video adapter qurğusunun qrafiki rejimlərini idarə edir. Qrafiki rejimlər ilk rənglərin sayı və piksellərin<sup>1</sup> sayı ilə fərqlənir.

Qrafiki iş rejimini müəyyən etmək üçün **uses** işçi sözü vasitəsilə **GRAPH** modulunu qoşmaq, sonra isə InitGraph(var Driver, Mode: Integer; Path: String)

<sup>1</sup> Piksel kompyuter qrafikasında ölçü vahididir.

prosedurunu yerinə yetirmək lazımdır. Qeyd edək ki, prosedur Path ünvanında yerləşən və Driver parametrlə təyin olunan drayveri operativ yaddaşa yükləyərək video adapteri Mode parametrlə təyin olunan iş rejiminə keçirir. Əgər proqram ixtiyari video adapter üçün nəzərdə tutulubsa, onda qrafiki iş rejimini aşağıdakı formada müəyyən etmək olar.

```
var DriverVar, ModeVar: integer;
begin
  DriverVar:=Detect;
  InitGraph(DriverVar, ModeVar, '\TP\GRAPH');
```

Qrafiki rejim müəyyən olunduqdan sonra Graph modulunun aşağıdakı prosedur və funksiyaları vasitəsilə qrafiki obyektlər təmin olunur.

### Prosedurlar

Prosedurlar	Təyinatı
SetColor(a: word);	Rəsmi rəngini müəyyən edir.
SetBkColor(a: word);	Fonun rəngini müəyyən edir.
SetFillStyle(b, a: word);	Qapalı oblast üçün rənglənmənin stilini və rəngini müəyyən edir, b-arqumenti stilin, a arqumenti rəngi təyin edir.
SetLineStyle(a, b, c: word);	Xəttin stilini, rəngini və qalınlığını müəyyən edir.
SetTextStyle(a, b, c: word);	Mətn üçün şrifti, stili və ölçüsünü müəyyən edir.
ClearDevice;	Monitoru təmizləyir .
SetViewport(x1, y1, x2, y2: integer, Clip: boolean);	Qrafiki düzbucaqlı pəncərə müəyyən edir. x1, y1, x2, y2 müvafiq olaraq pəncərənin sol yuxarı və sağ aşağı təpələrinin kordinatlarını təyin edir.
ClearViewport	Cari pəncərəni təmizləyir.
PutPixel(a, b, c: integer);	(a,b) kordinatlı c rəngli nöqtəni monitorda əks etdirir.
Line(x1, y1, x2, y2: integer);	(x1, y1) nöqtəsindən (x2,y2) nöqtəsinə yönəlmiş düz xətt çəkir.
Rectangle(x1, y1, x2, y2: integer);	Diaqonalı (x1, y1) nöqtəsindən (x2,y2) nöqtəsinə yönəlmiş düzbucaqlı çəkir.
Bar(x1, y1, x2, y2: integer);	Diaqonalı (x1, y1) nöqtəsindən (x2,y2) nöqtəsinə yönəlmiş müəyyən edilmiş rənglə və stillə rənglənmiş düzbucaqlı çəkir.
Bar3D(x1, y1, x2, y2, d: integer; a: boolean);	Paralelepiped çəkir. x1, y1, x2, y2 arqumentləri ön üzün yuxarı və aşağı təpə nöqtələrini kordinatları, d: yan tilinin qalınlığını, a: yuxarı tilin əks olunması

Prosedurlar	Təyinatı
	rejimini müəyyən edir.
Circle(x, y, r: word);	Mərkəzi (x, y) nöqtəsində olan r radiuslu çevrəni çəkir.
Arc (X,Y:Integer; U1, U2, R:Word)	Cari rəngli qövs çəkir. X, Y – qövsün mərkəzinin kordinatları, U1 – qövsün ilk nöqtəsinə qədər bucaq, U2 – qövsün son nöqtəsinə qədər bucaq R – qövsün radiusunu müəyyən edir.
Ellipse (X, Y: Integer; U1, U2, XR, YR: Word)	Ellips tipli qövs çəkir. X, Y – qövsün mərkəzinin kordinatları, U1 – qövsün ilk nöqtəsinə qədər bucaq, U2 – qövsün son nöqtəsinə qədər bucaq, XR, YR – üfüqi və şaquli yarımoxu müəyyən edir.
FillEllipse(x, y, Rx, Ry: integer);	Rəngli ellips çəkir.
MoveTo(x, y: integer);	Cari kursoru (x, y) nöqtəsinə sürüşdürür.
MoveRel(x, y: integer);	Cari kursoru üfüqi istiqamətdə x qədər, şaquli istiqamətdə isə y qədər sürüşdürür.
OutText(text: string);	Monitorda sətiri əks etdirir.
OutTextxy(x, y: integer; text: string);	Monitorda (x, y) nöqtəsindən başlayaraq sətir əks olunur.

### Funksiyalar

Funksiyalar	Təyinatı
GetBkColor: Word	Fonun rənginin kodunu müəyyən edir.
GetColor: Word	Cari rəngi müəyyən edir.
GetDriverName; String	Cari drayverin adını təyin edir.
GetGraphMode: Integer	Cari qrafiki rejimi təyin edir.
GetMaxColor; Word	SetColor prosedurunda parametrlər kimi istifadə edilən rəng kodunun maksimal qiymətini təyin edir.
GetMaxMode: Integer	Cari drayverin maksimal rejiminin nömrəsini təyin edir.
GetMaxX: Integer	Qrafiki rejimdə monitorun üfüqi istiqamətdə maksimal koordinatını təyin edir.
GetMaxY; Integer	Qrafiki rejimdə monitorun şaquli istiqamətdə maksimal koordinatını təyin edir.
GetModeName (ModeNum: Word): String	Cari anda aktiv olan qrafiki rejimin adını təyin edir.
GetPaletteSize: Integer	Rəng palitrasının ölçüsünü təyin edir.
GetPixel (X, Y; Integer) : Word	(X,Y) kordinatlı pikselin rəngini təyin edir.

<b>Funksiyalar</b>	<b>Təyinatı</b>
GetX: Integer	Kursorun absisini müəyyən edir.
GetY: Integer	Kursorun ordinatını müəyyən edir.
TextHeight (TextStr: String) : Word	Sətrin hündürlüyünü piksellə təyin edir.
TextWidth (TextStr: String): Word	Sətrin enini piksellə təyin edir.

Turbo Pascal dilində rənglər aşağıdakı rənglər və rənglənmə stilləri nəzərdə tutulmuşdur.

### Rənglər cədvəli

<b>Rəng</b>	<b>Kod</b>	<b>Rəng</b>	<b>Kod</b>
Black (qara)	0	DarkGray (tünd qonur)	8
Blue (göy)	1	LightBlue (mavi)	9
Green (yaşıl)	2	LightGreen (tünd yaşıl)	10
Gyan (firuzəyi)	3	LightGyan (tünd firuzı)	11
Red (qırmızı)	4	LightRed (tünd qırmızı)	12
Magenta (narıncı)	5	LightMagenta (tünd narıncı)	13
Brown (boz)	6	Yellow (sarı)	14
LightGray (açıq qonur)	7	White (ağ)	15

### Rəngləmə şablonu

<b>Rəngləmə şablonu</b>	<b>Kod</b>
EmptyFill	0
SolidFill	1
LineFill	2
LtSlashFill	3
SlashFill	4
BkSlashFill	5
LtBkSlashFill	6
HatchFill	7
XhatchFill	8
InterleaveFill	9
WideDotFill	10
CloseDotFill	11
UserFill	12

Məsələn,

```
uses graph; crt;
Const
```



```

PrD:real=80;
PATH='';
Var
W,H,gd,gm,i,N,X,Y.R:integer;
Angle:real;
procedure WH (var W,H:integer);
begin
W:=( GetMaxX+1);
H:=( GetMaxY+1);
end;
Function Xscr (X:integer):integer;
Begin Xscr:=X + W div 2 end;
Function Yscr (Y:integer):integer;
Begin Yscr:=H div 2 - Y end;
BEGIN
repeat
write ( ` Duzgun Coxbucaqlinin tereflerinin sayini
daxil edin);
readln(N);
until N>2;
gd:=DETECT;
initgraph (gd,gm,path);
WH(W,H);
R:=round(PrD*0.01/*H/2);
Angle :=2*pi/N;
MoveTo (Xscr(R), Yscr(0));
SetColor (Yellow);
For i:=1 to N do begin
X:=round (R*cos(Angle*i));
Y:= round (R*sin(Angle*i));
LineTo (Xscr(X), Yscr(Y));
End;
Repeat until keypressed;
CloseGraph;
End.

```

Proqram çevrənin daxilinə düzgün çoxbucaqlının çəkilməsini təmin edir.

## 2.21. Göstəricilər

Göstərici dəyişənin operativ yaddaşda mövqeyini bildirən dəyişəndir. Başqa sözlə göstərici verilənin saxlandığı operativ yaddaş sahəsinin ilk baytının ünvanını bildirir. Proqramda göstəricidən operativ yaddaşın dinamik paylanması üçün istifadə olunur. Belə ki, proqram kompilyasiya olduqda hər bir statik verilən üçün onun tipinə uyğun yaddaşda müəyyən yer ayrılır və ayrılmış yaddaşın ümumi maksimum 64 kбайt ola bilər. Bu səbəbdən böyük həcmli massivlərlə işlədikdə yaddaş çatışmamazlığı meydana çıxma bilər. Dinamik paylanma zamanı isə verilən operativ yaddaşın bütün oblastında saxlanıla bilər. Bu zaman dəyişənlər və sabitlər

üçün yaddaş kompilyasiya zamanı deyil, proqram yerinə yetirildikdə ayrılır və sonradan yaddaş azad oluna bilər. *Proqramın yerinə yetirildiyi zaman yaradılan və ləğv olunan dəyişənlər dinamik dəyişənlər adlanır.* Göstəricilərə aşağıdakı münasibət operatorları “<” və “=” tətbiq edilə bilər. İki göstərici eyni bir veriləni təyin etdikdə bərabər hesab olunur.

Göstərici təyin etdiyi dəyişənə uyğun olaraq tipli və tipsiz ola bilər. Tipli göstərici konkret tipli dəyişənin saxlandığı yaddaş sahəsinin ünvanını bildirir. Tipli dəyişəni elan etmək üçün “^” simvolundan istifadə olunur. Məsələn,

```
Type BytePtr = ^Byte;
      WordPtr = ^Word;
```

**Var**

```
A: BytePtr;
B: WordPtr;
C: ^integer;
```

Göstəricinin təyin etdiyi dəyişən istinad dəyişəni adlanır. İstinad dəyişəninə qiymət mənimsətmək üçün @ unar operatorundan istifadə olunur. @ operatoru istinad dəyişənin ünvanını bildirir. Həmçinin istinad dəyişəninə qiymətinə göstərici vasitəsilə müraciət etmək olar. Bu zaman göstərici dəyişənin identifikatorundan sonra ^ işarəsi qoyulmalıdır. Məsələn, aşağıda verilən proqramda k-dəyişəninə qiymətini iki üsulla ekrana vermək olar.

**Type**

```
      WordPtr=^Word;
```

**Var**

```
      p: WordPtr;
      k:Word;
      i:integer;
```

**begin**

```
      k:=3;
```

**For** i:=1 **to** 5 **do**

```
      k:=k+i;
```

```
      p:=@k;
```

```
      write (k, ' ', p^)
```

**end.**

Proqramın nəticəsi aşağıdakı kimi olacaq.

```
18 18
```

Qeyd edək ki, Turbo Pascal-da xüsusi göstərici **Nil** nəzərdə tutulmuşdur. O, heç bir yaddaş ünvanını bildirmir və ixtiyari tip istinad dəyişənilə birlikdə istifadə oluna bilər.

Turbo Pascal-da proqramın yerinə yetirilməsi zamanı tipi dəyişilən dəyişənlərin yaddaşda ünvanını müəyyən etmək üçün tipsiz göstəricidən istifadə olunur. Tipsiz göstərici **Pointer** işçi sözü ilə elan olunur. Proqramda qeyri-aşkar şəkildə tipsiz göstəriciyə tipli göstərici mənimsədilə bilər. Tipli göstəriciyə tipsiz göstərici isə yalnız aşkar şəkildə mənimsədilə bilər. Məsələn,

**Type**

```
      preal=^real;
```

**var**

```

p: pointer;
pr: ^integer;
a:preal;
begin
  p:=pr;
a: =preal(p);
pr^:=3.14;

```

Dinamik yaddaş və göstəricilərlə işləmək üçün aşağıdakı standart prosedurlar nəzərdə tutulmuşdur:

<b>Prosedur</b>	<b>Təyinatı</b>
Dispose(var P: Pointer)	Dinamik dəyişəni ləğv edir.
FreeMem(var P: Pointer; Size: Word)	Dinamik dəyişənin tutduğu yaddaşı təmizləyir
GetMem(var P: Pointer; Size: Word)	Konkret ölçülü dinamik yaddaş yaradır və göstərici dəyişənin ünvanına daxil edilir.
New(var P: Pointer)	Yeni dinamik dəyişən yaradır və ona göstərici təyin edir.

**Ә Д Ә В І У У А Т**

1. Информатика и информационные технологии : учеб. пособие для студентов / Ю.Д.Романова [и др.] ; под ред.Ю.Д.Романовой М. : Эксмо, 2008
2. С.Д. Шапорев. Информатика. Теоретический курс и практические занятия : учеб.для студентов вузов / С.Д.Шапорев СПб. : БХВ-Петербург, 2008
3. В.А. Острейковский Информатика : учеб.для студентов вузов М. : Высш. шк., 2007
4. С.В.Симонович Общая информатика : [универс.курс] СПб. : Питер, 2007
5. А.Н.Степанов Информатика : Учеб.пособие для студентов вузов / СПб.: Питер, 2007
6. С.Е.Столяр,А.А.Владыкин М Информатика : представление данных и алгоритмы /: БИНОМ. Лаб. знаний
7. Г. А.Сырецкий Информатика: фундам.курс : учеб.для студентов вузов : Т.2. Информационные технологии и системы / СПб.: БХВ-Петербург, 2007
8. М.Сухарев Turbo Pascal 7.0: теория и практика программирования / СПб. : Наука и техника, 2007
9. В.В.Фаронов Turbo Pascal : Учеб.пособие для студентов вузов / СПб.: Питер, 2007
10. В.В.Фаронов Turbo Pascal 7.0 : нач.курс:учеб.пособие для студентов вузов / М. : КНОРУС, 2007
11. Ю.С.Ковтанюк Программирование на Turbo Pascal / М. : Эксмо, 2008
12. Н.Б. Культин Программирование в Turbo Pascal 7.0 и Delphi СПб. : БХВ-Петербург, 2008
13. Ю.А.Шпак Turbo Pascal М. : Эксмо, 2008
14. М.П.Малыхина Программирование на языке высокого уровня Turbo Pascal : учеб.пособие для студентов СПб. : БХВ-Петербург, 2006
15. О.А.Меженный Turbo Pascal М. : Диалектика, 2006
16. С.В.Мельников Delphi и Turbo Pascal на занимательных примерах / СПб. : БХВ-Петербург, 2006
17. А.Н.Моргун Справочник по Turbo Pascal для студентов М. : Диалектика, 2006

18. В.В.Потопахин Turbo Pascal : Решение слож.задач СПб. : БХВ-Петербург, 2006
19. Фаронов В.В. Turbo Pascal 7.0.Начальный курс : Учеб.пособие для студентов вузов М. : КНОРУС, 2006
20. А.А.Борисенко Локальная сеть М. : Эксмо, 2006
21. А.И.Ватаманюк Беспроводная сеть своими руками / СПб. : Питер, 2006
22. Д.Н.Колисниченко Сделай сам компьютерную сеть : Монтаж,настройка,обслуживание СПб. : Наука и техника, 2006
23. А.А.Борисенко Локальная сеть М. : Эксмо, 2007
24. Джон Буравчик Локальная сеть без проблем М. : Лучшие кн., 2005
25. А.О.Даниленков,Ю.В.Васильев Локальная сеть своими руками : 100% самоучитель / М. : Технолоджи-3000, 2008
26. И.Я.Минаев Локальная сеть своими руками : 100% самоучитель М. : Технолоджи-3000, 2004
27. А.В. Поляк-Брагинский Локальная сеть. Самое необходимое СПб. : БХВ-Петербург, 2009

## MÜNDƏRICAT

<b>Ön soz</b> .....	3
<b>Giriş</b> .....	5
<b>FƏSİL 1 Kompüterlərin texniki-proqram təminatı</b> .....	7
1.1 Kompüterlər və onların inkişaf tarixi .....	9
1.2 Kompüterlərin təsnifatı.....	15
1.3 Masaüstü fərdi kompüterin tərkib hissələri .....	22
1.4 Alqoritm anlayışı, alqoritmin əsas xassələri, tipləri və təsvir üsulları.....	44
1.5 Kompüterlərin proqram təminatı.....	50
1.6 Alqoritmik dillər.....	57
1.7 Kompüter şəbəkələri.....	60
1.7.1 Şəbəkə komponentləri.....	62
1.7.2 Şəbəkə topologiyası.....	66
1.7.3 Şəbəkənin məntiqi arxitekturası.....	71
1.7.4 Naqilsiz şəbəkələr.....	72
1.7.5 Şəbəkə arxitekturaları.....	74
<b>FƏSİL 2 Turbo Pascal alqoritmik dili</b> .....	78
2.1. Dilin əsas elementləri .....	80
2.2 Proqramın strukturu .....	90
2.3 Verilənlərin tipləri. Sadə və sətir tiplər .....	90
2.4 Operatorlar .....	95
2.5 Giriş və çıxış prosedurları .....	101
2.6 Şərti keçid operatoru.....	103
2.7 Seçim və ya variant operatoru .....	106
2.8 Şərtsiz keçid operatoru .....	108
2.9 Dövr operatorları.....	108
2.10 Çoxluqlar.....	114
2.11 Massivlər.....	119
2.11.1 Massivlər üzərində əməllər.....	122
2.11.2 Sabit massivlər .....	125
2.12 Yazılar .....	126

---

2.12.1 Birləşdirmə operatoru.....	129
2.12.2 Variantla olan yazılar.....	130
2.12.3 Sabit yazılar.....	131
2.13 Prosedurlar .....	132
2.14 Funksiyalar .....	135
2.15 Prosedur tipli dəyişənlər.....	136
2.16 Rekursiyalar .....	138
2.16.1 Forward operatoru.....	139
2.17 Fayl tipli dəyişənlər. Fayllarla əməliyyatlar.....	140
2.18 Modullar .....	146
2.19 Standart prosedur və funksiyalar.....	149
2.20 Turbo Pascal dilinin qtafiki imkanları.....	155
2.21 Göstəricilər.....	160
<b>Ədəbiyyat</b> .....	164