

3 – тақырып

Турбо Паскальдағы деректердің негізгі түрлері

- деректердің толық, логикалық, символдық, заттық түрлері
- типтелген константтар, аударылған түрі, диапазон түрі
- жолды түрлері, жол үсті операциялары, стандарттық операциялар және жол функциялары (length, copy, uppercase, pos, insert, delete, str, val)
- жазу түрлері
- with ... do жалғанған операторлар
- варианттық жазу
- көптік түрлері, көптік конструкторы, жиынтыққа операция жасау

Программаның негізгі объектілері айнымалылар мен тұрақтылар болып табылады. Программа құру барысында, программалаушы олардың типтерін алдын-ала анықтауы қажет. Айнымалылар мен тұрақтылардың *типтері* – олардың мәндерін және оларға қолданылатын әр түрлі өндеу амалдарын белгілейтін, - басты көрсеткіші.

Turbo Pascal-да берілгендердің типтерін екі үлкен топтарға жіктеуге болады:

- Қарапайым типтер (скалярлық);
- Құрылымдық типтер (структуралық);

Қарапайым (скалярлық) типтердің өзі стандартты және пайдаланушылар типтері болып бөлінеді. Стандартты типтер: бүтін, нақты, логикалық (бульдік), символдық (литерлік).

Құрылымдық (структуралық) типтер құрамына қарапайым типтер кіреді. Құрылымдық типтерге жолдар, массивтер, жазбалар және файлдар жатады.

Бүтін тип

Паскаль тілінің стандартында бүтін типтің Integer бір ғана түрі анықталса, Turbo Pascal-да стандартты бүтін типтердің бес түрі бар: **shortint**, **integer**, **longint**, **byte**, **word**. Бұл типтердің сипаттамасын 2.1 кестеден көре аласыз.

| Тип | Диапазоны | Форматы | Көлемі (байт) |
|----------|--------------------------|---------|---------------|
| Shortint | -128...127 | Таңбалы | 1 |
| Integer | -32768...32767 | Таңбалы | 2 |
| Longint | -2147483648...2147483647 | Таңбалы | 4 |
| Byte | 0...255 | Таңбалы | 1 |
| Word | 0...65535 | Таңбалы | 2 |

Turbo Pascal-да бүтін типтегі оң және теріс сандар әдеттегідей жазыла береді. Оларға барлық арифметикалық амалдарды қолдануға болады.

Логикалық (бульевтік) тип

Паскальда стандартты логикалық тип – **Boolean** (көлемі – 1 байт). Логикалық типтің бұл атауы, атақты математик Джордж Буль есіміне байланысты **Boolean**(бульевтік) деп аталады. Логикалық типтегі элемент тек екі мәнді қабылдай алады: **true** (ақиқат), **false** (жалған). Turbo Pascal-да логикалық типтің жаңа түрлері анықталған. Олар: **ByteBool** (көлемі – 1 байт), **WordBool** (көлемі – 2 байт), **LongBool** (көлемі – 4 байт). Барлық логикалық типтердің мәні **true** (ақиқат) болғанда, 1 санына сәйкес келеді. Ал, олардың **false** (жалған) мәні 0 санына тең болады.

Логикалық типтегі берілгендер программада шартты тексеруде қолданылады. Логикалық шарт мына қатынас таңбалары: $<$ (кіші), $>$ (үлкен), $=$ (тең), $< >$ (тең емес), $<=$ (кіші немесе тең), $>=$ (үлкен немесе тең) арқылы жазылады. Мысалы, $12 < 30$ нәтиже – true (ақиқат), $12 = 30$ нәтиже false (жалған).

Логикалық типтегі берілгендерге логикалық амалдар қолданамыз. Логикалық амалдар арқылы логикалық өрнектер ұйымдастыруға болады. Логикалық өрнектің нәтижесі әрқашанда екі мәннің: **true** (ақиқат), **false** (жалған) біріне тең болады. Логикалық амалдардың толық тізімін кестеден көре аласыз.

| Амалдар | Іс-әрекет | Өрнек | A | B | Нәтиже |
|---------|---------------------------------------|---------|-------|-------|--------|
| Not | Логикалық “терістеу” | Not A | True | | False |
| | | | False | | True |
| And | Логикалық “және” (конъюнк-ция) | A and B | True | True | True |
| | | | True | False | False |
| | | | False | True | False |
| | | | False | False | False |
| Or | Логикалық “немесе” (дизъюнкция) | A or B | True | True | True |
| | | | True | False | True |
| | | | False | True | True |
| | | | False | False | False |
| Xor | “немесеге” кері | A xor B | True | True | False |
| | | | True | False | True |
| | | | False | True | True |
| | | | False | False | False |

Символдық (литерлік) тип

Символдық (литерлік) тип Char ASCII (American Standart Code for Information Interchange) кодының барлық түріне сәйкес келетін символдарды анықтайды. Char типтегі көрсеткіштердің кодын `ord` функциясымен алуға болады. Осы кодтары арқылы символдық типтегі көрсеткіштерді бір-бірімен салыстыруға болады.

Белгілі бір іс - әрекеттерді атқаратын, басқарушы символдарға 0-ден 31-ге дейінгі код тағайындалған.

- Символдық типті айнымалылар мәні ‘ (апостроф) таңбасына алынады. Мысалы: ‘A’, ‘\$’, ‘3’ т.с.с.

Символдық типтің кеңейтілген түсінігі жолдық тип `string` болып табылады.

Нақты тип

Паскальдің стандартында нақты типтің бір ғана түрі `real` анықталса, Turbo Pascal-да нақты типтің 5 түрі қолданылады: `real`, `single`, `double`, `extended`, `comp`.

| Тип | Диапазоны | Форматы (цифр саны) | Көлемі (байт) |
|----------|--|---------------------|---------------|
| Real | $2,9 \cdot 10^{-39} \dots 1,7 \cdot 10^{38}$ | 11-12 | 6 |
| Single | $1,5 \cdot 10^{-45} \dots 3,4 \cdot 10^{38}$ | 7-8 | 4 |
| double | $5 \cdot 10^{-324} \dots 1,7 \cdot 10^{308}$ | 15-16 | 8 |
| Extended | $3,4 \cdot 10^{-4932} \dots 1,1 \cdot 10^{4932}$ | 19-20 | 10 |
| Comp | $-2^{63} + 1 \dots 2^{63} - 1$ | 19-20 | 8 |

Мұндағы, **single, double, extended, comp** типтерін компьютердің жетілген, соңғы түрлерінде жұмыс істегенде қолданған тиімді. Себебі, бұл нақты типтің түрлерін пайдаланғанда компьютердің қосымша процессоры болғаны жөн. Сол кезде, есептеулер сапасы жоғары, әрі жылдам жүргізіледі. Қарапайым программаларда **real** типі пайдаланылады. Ал, **real** типі бойынша жүргізілетін есептеулерге қосымша процессордың қажеті жоқ.

Нақты типтегі сандарды жазудың екі тәсілі бар: әдеттегідей және экспоненциал. Нақты сандарды Turbo Pascal-да әдеттегідей жазу үшін, санның бүтін бөлігін ажырататын үтір (,) таңбасының орнына нүкте (.) қойылады.

Мысалы: $125,45 \rightarrow 125.45$, $-52,1 \rightarrow -52.1$

Өте үлкен және өте кіші нақты сандарды жазуда 10 санының дәрежесін қолданған ыңғайлы (мысалы: $12 \cdot 10^{-16}$, $3 \cdot 10^8$). Осындай сандарды экспоненциал түрге айналдыру үшін 10-ның орнына E символы және дәреже көрсеткіші жазылады.

Мысалы:

| Математикада жазылуы: | Экспоненциал жазылуы: |
|-----------------------|-----------------------|
| $31 \cdot 10^{14}$ | $31E+14$ |
| $5 \cdot 10^{-21}$ | $5E-21$ |
| 0,001 | $10E-4$ |
| 1,0 | $1E+0$ |

Кез келген нақты типтегі сан компьютер жадысында экспоненциал түрде сақталынады. Программа нәтижесі нақты сан болса, бұл сан экранға экспоненциал түрде шығады.

Жиын – элементтерінің түрі тең және мәндері бір бірімен байланысқан тізім. Байланыстарды тек қана программа құрушы белгілейді. Жиынға кіретін элементтердің саны 0...255 дейін өзгереді. Элементі жоқ жиын *бос* жиын деп аталады. Жиынның элементтерінің саны алдын ала белгіленбейді.

Екі жиын тең деп саналады, егер олардың барлық элементтері тең болса және кездесетін реті сақталмасада болады. Егер бір жиынның элементтері екінші жиында кездессе, онда бір жиын екінші жиынға *енгізіліген*. Бос жиын барлық жиында кездеседі.

Мысалдар,

[] – бос жиын;

[2,3,5,7,11] – 2,3,5,7,11 бүтін сандардан құрылған жиын;

['a','c','d','f'] – a,c,d,f символдарынан құрылған жиын;

[1, k] – 1 және бүтін k санның ағымды мәнінен құрылған жиын. Егер k=1, онда жиын бір саннан тұрады.

Егер жиынның элементтерінің мәні өспелі диапазонда жатса, онда оларды қысқаша түрде жазамыз:

[1,2..100] = [1..100] – 1...100 диапазонның бүтін сандары;

[k..2*k] – k...2k диапазонның бүтін сандары.

['a'..'d', 'f'..'h', 'k'] – a,b,c,d,f,g,h,k латын әріптерінен тұрады.

Жазба дегеніміз - өрістер деп аталатын, әр түрлі типтерінен тұратын мәліметтердің құрылымы. Осындай ақпараттарды ПР тілінде сипаттау үшін **RECORD** типі қолданылады.

«Жазба» типі **Type** мәліметтерді сипаттау бөлімінде сипатталуы керек. Жалпы программадағы жазылу түрі:

Атауы=**record**

Өріс1:тип1;

Өріс2:тип2;

...

ӨрісN:типN;

End;

Мұндағы: *Атауы* – жазба-типiнiң аты;

ӨрісN және *типN* - жазбадағы N өрісiнiң типi

және атауы;

Мысал:

Type

Person=record

Fam:string[20];

Name:string[15];

Day:integer;

Month:integer;

Year:integer;

Address:string[50];

End;

Date=record;

Day:integer;

Month:integer;

Year:integer;

End;

Типті сипаттағаннан кейін жазба-айнымалыны көрсетуге болады:

Var

Student:person;

Birthday:date;

Программада жазба-айнымалысының өрісін пайдалану үшін, айнымалының атын және өрістің атын көрсету керек. Мысалы, келесі инструкция

Write(person.fam, ' ',person.name);

орындалғаннан кейін экранға *person* жазба-айнымалысындағы *fam* және *name* өрістерінің мазмұнын шығарады.

WITH инструкциясы

WITH инструкциясы программада өрістердің атауын жазба-айнымалысында көрсетпей-ақ қолдана беруге мүмкіндік береді.

Мысал: Жазба-айнымалысына мәліметтерді клавиатурадан енгізеді.

```
Program esep;
```

```
Var
```

```
Person=record
```

```
    Fam:string[20];
```

```
    Name:string[15];
```

```
    Address:string[50];
```

```
End;
```

```
Begin
```

```
Writeln('Мәліметтерді енгізіңіз');
```

```
With person do
```

```
Begin
```

```
Write('Фамилия:');
```

```
Readln(fam);
```

```
Write('Аты:');
```

```
Readln(Name);
```

```
Writeln('Адрес:');
```

```
Readln(address);
```

```
End;
```

```
End.
```


Жолдармен жұмыс

Турбо Паскальда символдар тізбегінен құрылған тұрақтылар (мәтіндер) және мәндері осы тұрақтылардан тұратын айнымалылармен жұмыс істеу де мүмкін, тек әр тұрақтының ұзындығы (мәтіндегі символдар саны) 255-тен аспаса болғаны. Оларды жолдық тұрақты, жолдық айнымалы не қысқаша жол деп атайды. Мысалы:

```
ai := 'Ахмет', stud := 'Біз студенттеріміз'
```

меншіктеу операторларының оң жағындағы мәтіндер жолдық тұрақтылар; `ai`, `stud` — жолдық айнымалылар (идентификаторлар). Жолдың типі оң жағына ашу, жабу тік жақшалары қойылып, олардың ішіне мәнін жаққа енгізу үшін таңдап алынған бүтін сан енгізілген `STRING` (`string` — жол) қызметші сөзі арқылы анықталады. Оны программаның `TYPE` не `VAR` бөлімінде сипаттауға болады.

Мысалы:

```
type jol1 = string[5]; jol2 = string[17]; var al : jol1;  
stud : jol2;
```

```
var al : string [5]; stud : string[17].
```

Бұлардағы `string[5]`, `string[17]` — жолдық типтер; `jol1`, `jol2` — осы типтерге берілген атаулар; `al`, `stud` — жолдық айнымалылар. Бірінші

түрдегі сипатталудың мағынасы мынадай: `al`-дің типі — `jol1`, ал `jol1` — `string[5]` типінің атауы, т.с.с.

Ұзындығын енгізбей, жолдық айнымалыны `al: string` түрінде сипаттау да мүмкін. Бұл кезде `al` үшін жадтан 255 символдық орын қалдырылады.

Турбо Паскальда жолдармен түрлі әрекеттер жасап, өндеуге ар-налған мынадай стандартты функциялар бар:

LENGTH, '+' * не CONCAT, COPY, POS

length — ұзындық; '+' — плюс таңбасы, *concat* — біріктіру, жалғас-тыру; *copy* - көшірме, *pos*- (position) - позиция, тұрған орны).

* Length (стр) - жолдың ұзындығын *есептейді* (*стр* (*string*) — жолдық айнымалының не оның мәнінің орнына пайдаланылған сим-волдар тізбегі, *string* — жол). Мысалы:

length (al)=5; length (stud)=17; length ('мен')=3

(мұндағы al, stud — жоғарғы мысалда тандап алынған жолдық типті айнымалылар ар.).

* Concat (стрі, стр2) не стрі + стр2 — стрингі мен стринг2 жол- дарын біріктіреді. Мысалы:

concat ('Ахмет', 'ұлы') → Ахметұлы; 'Ахмет'+ 'ұлы' → Ахметұлы

(мұндағы → белгісі — "орындалу нәтижесі" сойлемінің орнына пай-даланылған белгі).

* Сору (стр, ш, п) — жолдың m символынан бастап p символдық үзіндісін кесіп (ойып) алады. Мысалы:

сору ('Ахметұлы', 3, 5) → метұл; сору ("Ахметұлы1", 1, 1) → А.

* Pos (стрі, стр2) — стринг2 жолынан стрингі үзіндісін іздейді де, жолдың басынан есептегенде үзінді басталған орынның немірін анықтайды, үзінді табылмаса, оны 0-ге тең деп нәтмжелейді.

Жолға қатысты мынадай процедуралар да бар:

$Val(s, x, c)$ — ондық саннан тұратын жолдық тұрақтыны санға айналдыру. Мұндағы s — негізгі жол, x — сандық айнымалы, c — бүтін айнымалы. S -ті санға айналдыру мүмкін болмаса, c нөл емес мөнді қабылдайды. Мысалы, $s='438'$ болса, $val(s, x, c)$ процедурасының орындалу нәтижесі — $x=438$; $c=0$.

$Str(x, s)$ — *val-ға кері* процедура, яғни x -тің сандық мәнін жолдық тұрақтыға айналдырады. Мысалы, $x=435$ болса, $Str(x, s)$ процедурасының орындалу нәтижесі — $s='435'$.

Машинаның символдарды салыстыруына болатынын пайдаланып, адамдардың аты-жөндерін алфавит бойынша компьютер арқылы реттеп, қағазға басып шығару қиын емес.

Мысал. Қазақ алфавитінің өріптерімен жазылған адамдардың аты-жөндерін бірінші әріптері бойынша реттеу программасын құру керек.

```
Program retteu;
const nl = 100;
type msl = string[25]; ms2 = array[1..nl] of msl;
var a : ms2; i, k, j : integer; bl : string[30]; begin
bl:=АӘБВГҒДЕЖЗИЙКҚД\ШҢдаіТРСТУҰҮ^^
L=Length(bl);
write ('Адам саны (n<=nl), n=?'); readln(n);
for k := 1 to n do begin
write (k, '-адамның фамшшысы-аты-'); readln (a[k]) end;
for k := 1 to 13 do write ('-'); writeln;
for j := 1 to L do for k := 1 to n do
if copy(a[k],1,1) = copy(bl, j, 1)then writeln (a[k]);end.
```

Қолданылған әдебиеттер тізімі

- 1) Марко Кэнту. Delphi 5 для профессионалов. –СПб.:Питер, 2001.
- 2) Бабушкина И. А., Окулов С.М. Практикум по объектно-ориентированному программированию. М.: БИНОМ, Лаборатория знаний, 2004. – 366 бет.: ил.
- 3) Хомоненко А.Д. и др. Delphi 7. – СПб.: БХВ-Петербург, 2004.- 1216 бет:ил.
- 4) Фаронов В.В. Delphi 5: Учебный курс.-М.: Нолидж, 2001.- 605 бет.: ил.
- 5) Фаронов А.В. TURBO PASCAL /учебник// Изд. «Питер» М.-2001.
- 6) Культин Н. Turbo Pascal в задачах и примерах. - СПб.: БХВ-Петербург, 2001.-256 бет: ил.
- 7) Матаев С. Delphi 7. Бағдарлама құру негіздері: Оқу құралы. Қарағанды, 2005. – 271 б.
- 8) Н.Культин. Основы программирования в Delphi7. – СПб.: БХВ-Петербург, 2003.