

# Turbo Pascal программалау тілі

---

**Баймаханова А.Б**

# Мақсаты

Паскаль программалау тілінің алфавитімен таныстыру және алгоритмді программалау тілінде жазу ережелерін түсіндіру.

**Turbo Pascal – жоғарғы деңгейлі программалау жүйесі. Оның тұңғыш нұсқасын 1970 жылы швейцария физигі Niklaus Wirth ұсынған. Автор тілді алғашқы арифметикалық есептеу машинасын жасаған француздың ұлығалымы Блез Паскальдің құрметіне Паскаль деп атап, тіл атауын үнемі бас әріппен жазылуын сұраған.**

**Паскаль тілін 1968-1971 жылдары швейцариялық ғалым Никлаус Вирт қолайлы программалау тілі ретінде ұсынған болатын.**

**Бұл тілдің стандарты кейінірек бекітілді, ол сол кездерде кең таралған**

**АЛГОЛ, ФОРТРАН, БЕЙСИК тілдеріне қарағанда жетілдірілген, жұмыс істеуге ыңғайлы тіл болды.**

**Паскаль тілінде жазылған программаның дұрыстығын компьютерде тексеру және жіберілген қатені түзету оңай.**

**Қазіргі кезде Паскаль тілі кез келген күрделі есептерді шығара алатын, кең таралған стандартты оқу тіліне айналды.**

Кейінен тілдің көптеген нұсқалары жарық көреді. Turbo Pascal – оның тұңғыш жоғарғы деңгейлі программалау жүйесі.

Бейсик тілі сияқты, Паскальды оқып-  
үйренуге жеңіл, түрлі салалық  
информациямен жұмыс істеуде нәтижелі  
болғандықтан, дүние жүзінде көп тараған  
тілдердің бірі.

# Паскаль тілінің негізі элементтері

Программалар белгілі бір мәселені, есепті шешуге арналған. Есеп шығару барысында компьютерге бастапқы мәліметтер енгізіледі, оларды қалай өңделетіндігі көрсетіледі және нәтиже қандай түрде, қандай құрылғыға шығарылатыны айтылады.

Паскаль тілінде программа жеке - жеке жолдардан тұрады. Оларды теру, түзету арнайы мәтіндік редакторлар атқарылады. Программа алдындағы азат жол немесе бос орын саны өз қалауымызша алынады. Бір қатарда бірнеше команданемесе оператор орналаса алады, олар бір - бірімен нүктелі үтір (;) арқылы ажыратылып жазылады, бірақ бір жолда бір ғана оператор тұрғаны дұрыс, ол түзету жеңіл, әрі оқуға ыңғайлы.



Кез келген программаның алғашқы жолы **PROGRAM** сөзінен басталатын оның тақырыбынан тұрады. Одан кейін программаның ішкі объектілерінің сипатталу бөлімі жазылады. Бұл бөлім программадағы айнымалылар, тұрақтылар тәрізді объектілердің жалпы қасиеттерін алдын ала анықтап алуға көмектеседі.

Сипаттау бөлімі бірнеше бөліктерден тұрады, бірақ программаның күрделілігіне байланысты көбінесе ол бір немесе екі ғана бөліктен тұруы мүмкін.

Программаның соңғы және негізгі бөлімі операторлар бөлімі – болып табылады. Орындалатын іс-әрекеттер, командалар осы бөлімде орналасады. Ол **begin** түйінді сөзінен басталып, барлық атқарылатын операторлар (командалар) тізбегі жеке-жеке жолдарға жазылып біткен соңы **end** түйінді сөзімен аяқталады.

# Мәлімет типтері

Мәліметтердің немесе шамалардың типі деп, олардың қабылдай алатын мәндерінің және олармен орындауға болатын амалдардың жиынын анықтауды айтады, яғни тип дегеніміз – шамалардың қабылдайтын мәндеріне берілетін сипаттама.

Мәліметтердің әр типі тек өзіне ғана сәйкес келетін операциялар жиынын орындата алады.

Мысалы: 1 мен 2 мәндері бүтін сандар типіне жатады, оларды қосуға, азайтуға, көбейтуге және бөлуге болады. Ал, «IBM» және «PC» мәндері сөз тіркесі типіне жатады, бұларды біріктіріп жазуға ғана болады («IBM PC»). Қосуға, азайтуға, көбейтуге, бөлуге болмайды. Кез келген тұрақты, айнымалы функция немесе өрнек өзіне тән бір типпен ғана сипатталады.

Паскаль тілінде шамалардың типін көрсету міндетті болып табылады. Сондықтан программа алдына оның сипаттау бөлімінде пайдаланылатын барлық шама атаулары және оның типтері көрсетілуі қажет.

Паскаль тілінде пайдаланылатын барлық типтер скалярлық (қарапайым) және құрылымдық (структуралық) болып үлкен екі топқа бөлінеді. Скалярлық (қарапайым) типке шамалардың стандартты типі және жасанды тип (тағайындалған) жатады.

# Стандартты типтерге

- Бүтін - **INTEGER**;
- Нақты - **REAL**;
- Логикалық - **BOOLEAN**;
- Тіркестік - **(STRING)**;
- Мәтіндік (**TEXT**) тәрізді типтер жатады.



Мәліметтердің қарапайым түрлерін стандартты типтер деп атау қалыптасқан. Стандартты типтердің **real** түрінен өзгелері **реттелген** деп аталады, өйткені оларды бүтін сан арқылы өрнектей отырып өсуі не кемуі бойынша реттеп орналастыруға болады.

**Программалау тілдерінің  
қарапайым операторларына:  
меншіктеу, шартсыз көшу, енгізу-  
шығару операторлары мен бос  
оператор және процедураларды  
орындау (шақыру) операторлары  
жатады.**

Ал күрделі немесе құрама операторларға: **шартты көшу, қайталау(цикл), таңдау, жалғастыру (тіркестендіру) т.б. операторлары жатады.**

Математикадағы қарапайым  
теңдеу тәрізді айнымалыларға  
сандық (символдық та болуы  
мүмкін) мен беру бұл жағдайда  
**меншіктеу операторы** деп  
аталады.

# Көшу операторы

Көшу немесе шартсыз көшу операторы операторлардың рет-ретімен орналасуын бұзып, келесі атқарылуды, яғни белгісі бар операторға көшуді орындайды. Бұл оператор **GO TO** (көшу) түйінді сөзінен басталады да, одан кейін **LABEL** бөлімінде сипатталған белгі атауы жазылады.

# Бос оператор

- Бос оператор ешқандай да әрекеттің орындалмайтынын көрсетеді, ол программа блогынан шықпай-ақ, бірнеше жолды орындамай өткізіп жіберіп, блок соңына бірден көшетін кезде
  - пайдаланылады.

Паскаль тілінде мәлімет енгізілетін READ  
(READ-ағылшынша оқу) операторының  
жалпы түрі (пішімі)  
төмендегідей

- READ (a1, a2,...,an);
- READLN (a1, a2, ..., an);
- READLN;

Мұндағы  $a_1, \dots, a_n$ –айнымалы атаулары, оларды енгізу операторының параметрлері деп те атайды. Алғашқы оператор көрсетілген айнымалылардың мәндері пернелер арқылы араларына бір бос орын қалдыру арқылы енгізілген соң, курсорды келесі қатарға көшірмейді. Ал `readln` (`readline` – «қатарды оқу» деген сөздің қысқартылуы) алдыңғы оператордағыдай пернелерден мәндер енгізілгеннен кейін курсорды келесі қатарға көшіріп жібереді.



# Шығару операторы

- Паскаль тілінде нәтижені экранға шығару үшін WRITE (жазу) операторы (процедурасы деп те айтылады) қолданылады:
- WRITE (a1, a2,... , an);
- Паскаль тілінде параметрсіз шығару операторы WRITELN жаңа жолға өту үшін қолданылады.

# Басқару операторы

- Паскаль тілінің басқару операторларына шартсыз және шартты көшу операторлары, цикл ұйымдастыру операторлары жатады.

Шартты операторлар:

If

Case

**Көпжағдайда аргументтердің әр түрлі мәні бойынша алгоритмнің белгілі бір бөліктерін бірнеше рет қайталауға тура келеді. Осындай процестерді ұйымдастыру үшін – қайталау операторлары қолданылады.**

● Паскаль тілінде қайталау процесін жүзеге асыратын циклдік операторлардың үш түрі бар: арифметикалық цикл – **FOR**, шартты алдын ала тексеретін цикл – **WHILE** және шартты соңынан тексеретін цикл – **REPEAT**.

- Паскаль тілінде қайталау процесін жүзеге
- асыратын
- циклдік операторлардың үш түрі бар.
  - Арифметикалық цикл - FOR
  - Шартын алдын ала тексеретін цикл - WHILE
  - Шартты соңынан тексеретін цикл - REPEAT

Егер шартын алдын ала тексеретін циклдегі операторды неше рет қайталау керек екендігі белгісіз болып, оның тек қайталану шарты берілсе, онда WHILE, REPEAT операторлары пайдаланылады. Ал FOR операторы қайталану саны алдын ала белгілі болған кезде қолданылады.

- WHILE
- REPEAT
- FOR

## FOR операторы.

- Циклдегі операторларды қайталау саны алдын ала белгілі болған жағдайда FOR операторы қолданылады. Бұл операторды параметрлі цикл операторы деп те атайды, өйткені қайталау саны функция аргументі (айнымалы) сияқты циклдік параметрі қызметін атқаратын басқару айнымалысы арқылы беріледі. Параметр өзінің соңғы мәніне дейін бірлік қадаммен өзгеру барысында циклге кіретін бір немесе бірнеше операторды қайталап орындау ісін атқарады.

# Оның жазылуы екі түрде болады:

1) Айнымалының 1-ге артуы:

For <айнымалы>:=<S1> to <S2> do <оператор>;

```
for <айнымалы> := <алғашқы мән> to  
    <ең соңғы мән> do begin  
    {цикл денесі}  
end;
```

2) Айнымалының 1-ге кемуі:

For <айнымалы>:=<S1> downto <S2> do <оператор>;

```
for <айнымалы> := <алғашқы мән>  
    downto  
    <ең соңғы мән> do begin  
    {цикл денесі}  
end;
```



## WHILE операторы

Алдын ала берілген шарт бойынша қайталауды жүзеге асыратын бұл оператор WHILE және DO түйінді сөздері қолданылып орындалады. Цикл әрбір орындалар алдында қайталану шартының мәні есептеледі, егер ол ақиқат – true болса, цикл тұлғасы орындалады да, шарт қайта тексеріледі. Қайталану шарты жалған – false болса, циклді орындау доғарылып, while операторынан кейінгі жолдар атқарыла бастайды.

# Ерекшеліктері

```
while <шарт> do begin  
    {цикл денесі}  
end;
```

күрделі шарттарды қолдануға болады:

```
while (a<b) and (b<c) do begin  
    {цикл денесі}  
end;
```

- егер цикл денесінде бір ғана оператор болса, **begin** мен **end** сөздерін жазбай кетуге болады:

```
while a < b do  
    a := a + 1;
```

- циклге кірген сайын шарт есептеліп отырады  
егер шарт циклге кіруде жалған болса, цикл бірде-бір

рет

```
a := 4; b := 6;  
while a > b do  
    a := a - b;
```

егер шарт түбінде жалған болмайтын болса,  
программаның жұмысы дағдарысқа ұшырайды  
(тоқтаусыз орындала берелі)

```
a := 4; b := 6;  
while a < b do  
    d := a + b;
```

## REPEAT операторы.

REPEAT қайталау операторы WHILE операторына ұқсас, айырмашылығы – қайталану шарты цикл соңында тексеріледі, сондықтан ол кем дегенде бір рет орындалатын болады. Екінші өзгешелігі – цикл тұлғасы шарт жалған болғанда қайталанып, ол ақиқат болған кезде циклді орындау доғарылады.

Бұл оператор REPEAT және UNTIL түйінді сөздерін пайдаланып орындалады.

# Ерекшеліктері:

цикл денесі кемінде бір рет орындалады  
`until` ("...болмайынша") сөзінен кейін  
циклдан ШЫҒУДЫҢ шарты қойылады

```
program qq;  
var n: integer;  
begin
```

```
  repeat  
    writeln('Оң сан енгіз');  
    read(n);  
  until n > 0;
```

ШЫҒУДЫҢ шарты

```
    ... { негізгі алгоритм }  
end.
```

- Экранға  $X$  – тің барлық  $1$  – ден  $100$  – ге дейінгі бүтін мәндері үшін  $Y=x^2-50*x-30$  өрнегінің нәтижесін шығару командасы.

Программада  $X$  идентификаторын өрнектің айнымалысы әрі циклдің параметрі ретінде белгілеп,  $1$  – ден  $100$  – ге дейін қайталанатын цикл ұйымдастырамыз. Цикл параметрі  $X$   $1$  – ді меншіктеп, өрнектің  $X = 1$  болғандағы мәнін шығарады. Цикл параметрінің келесі мәні  $2$  – ні меншіктеп, өрнектің  $X = 2$  болғандағы мәнін шығарады, әрі қарай цикл аяқталғанша өрнектің сәйкес мәндері шыға береді. Цикл параметрі  $101$  мәнін қабылдағанда, цикл тоқтайды.



- **PRAGRAM Tendeu 1;**
- **VAR X:Byte; Y:Longint;**
- **BEGIN**
- **For X:= 1 To 100 Do**
- **Begin**
- **Y:= SQR ( X ) + 50\*X-30;**
- **Writeln ('Y=', Y );**
- **End;**
- **Readln END.**

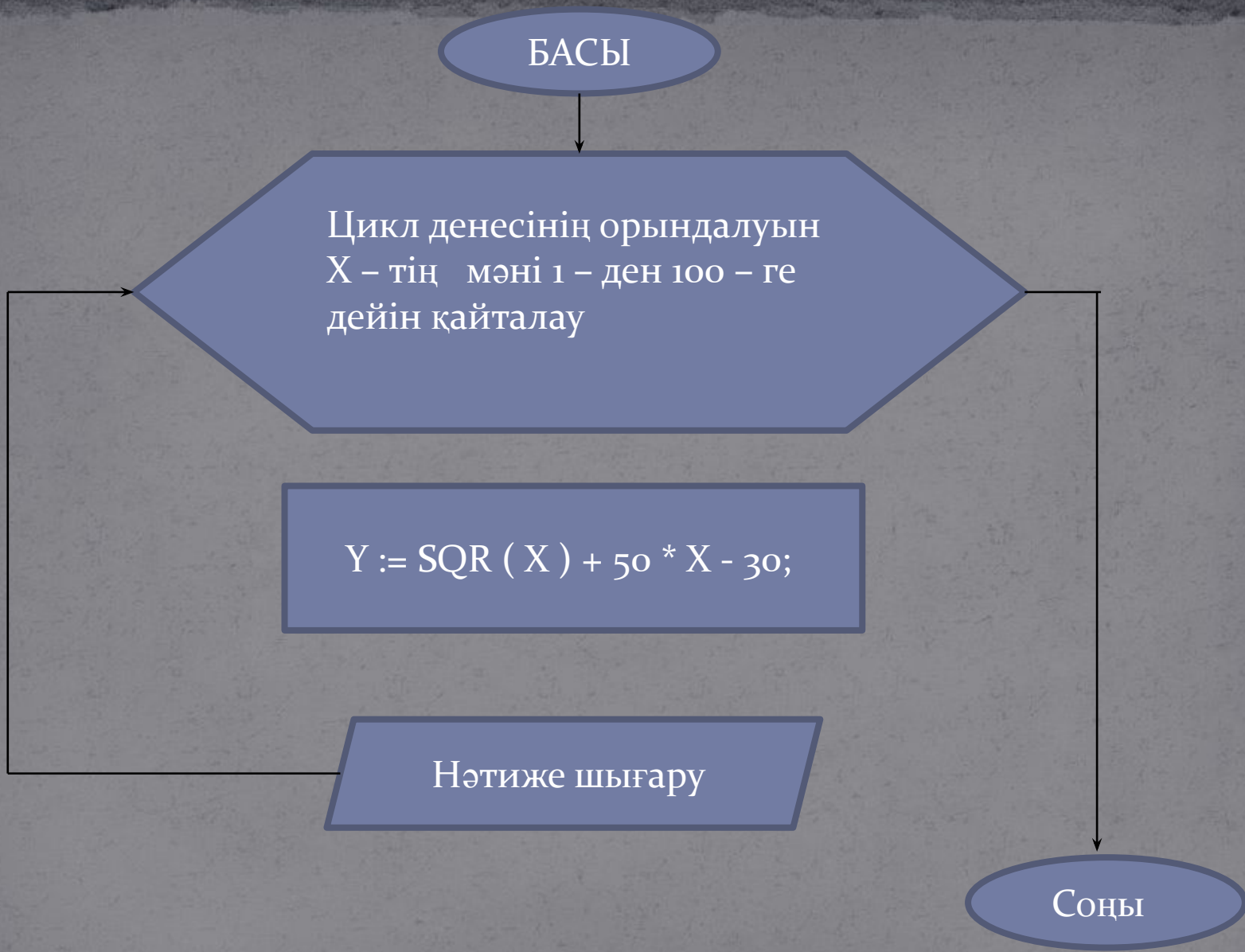
БАСЫ

Цикл денесінің орындалуын  
X – тің мәні 1 – ден 100 – ге  
дейін қайталау

$Y := \text{SQR}(X) + 50 * X - 30;$

Нәтиже шығару

Соңы





Көңіл бөлгендеріңізге  
рахмет

