


Программалаштыру дәресләре

VIII нче дәрес. Pascal'да циклар: while, repeat һәм for

📅 26.05.201826.05.2018 📄 [tatprog](#)



while
repeat
for

Исәнмесез, кадерле укучылар! Менә без сезнең белән циклар өйрәнүгә якынлаштык. Pascal телендә циклар нәрсә ул? Аларны ничек кулланырга? Алар ни өчен кирәк? Бу дәрестә мин нәкъ шул сорауларга җавап бирәм.

Белгәнегезчә, алгоритмнарның өч төре бар: сызыклы, тармаклы һәм циклик алгоритмнар. Без сызыклы һәм тармаклы алгоритмнарны өйрәндөк инде, ә хәзер циклик алгоритмнарны башлыйбыз.

Күпчелек программалау телләрендә кебек, Pascal телендә циклик конструкцияләр өч тибы бар.

Цикл типлары:

1. Алшартлы цикл
2. Артшартлы цикл
3. Параметрлы цикл

Һәр цикл тәннән һәм шарттан тора. Циклның тәне — ниндидер операторлар. Шарты — логик аңлатма, циклның тәнненең кабатлануы шартның кыймәтеннән тора.

Өйдәгез, бер мәсьәлә алыгыз һәм аны төрле цикл типлары ярдәмендә чишигыз.

1 мәсьәлә. Бердән башлап, клавиатурадан кертелгән санга кадәр бөтен саннарны экранга чыгарырга. Мәсәлән, кертелгән саны 5 булса, программа экранга шуны чыгарырга тиеш: 1 2 3 4 5

while, яки алшартлы цикл

while циклында шарт аның тәне алдыннан языла. Шарт true булганда гына цикл кабатлана, false булгач ук — цикл тәмамлана.

while'ның форматы шундый:

```
while <шарт> do <оператор>;
```

Ә әгәр while'ның тәнендә бердән артык оператор бар, аларны оператор жәяләр эчендә язарга кирәк:

```
while <шарт> do begin <операторлар> end;
```

Өйдәгез, бу циклыны мисалында карыйк.

```
program while_example;
```

```
var
```

```
i, N: integer; { үзгәрешлеләр игълан итәбез }  
begin
```

```
i := 1; { i'гә 1не билгелибез }
```

```
readln(N); { Бер сан укыйбыз }
```

```
while i <= N do {i N'нан артык булгач ук, цикл тукталачак (< да язарга мөмкин,  
эмма шул очракта N + 1 язарга туры киләчәк) }
```

```
begin {Оператор жәяләр ачабыз}
```

```
write(i, ' '); {i'ны экранга чыгарабыз}
```

```
inc(i); {i'ны 1'гә арттырабыз}
```

```
end; { оператор жәяләр ябабыз }
```

```
end.
```

repeat, яки артшартлы цикл

While'дан аермалы буларак, repeat циклында шарты аның тәненнән соң тора. Шарт false (!) булганда гына цикл кабатлана, true булгач ук — цикл тәмамлана.

Repeat'ның форматы шундый:

```
repeat
```

```
<операторлар>
```

```
until <шарт>
```

Begin һәм end язарга кирәкми.

Беренче мәсьәләне repeat ярдәмендә чишелеш:

```
program repeat_example;
```

```
var
```

```
i, N: integer;{ үзгәрешлеләр игълан итәбез }
```

```
begin
```

```

i := 1; { i'гә 1не билгелибез }
readln(N); { Бер сан укыйбыз }
repeat

write(i, ' '); {i'ны экранга чыгарабыз}
inc(i); {i'ны 1'гә арттырабыз}
until i = N + 1; {Мәсәлән, i = 11, ә N = 10. Цикл тәмамланачак, чөнки шарт инде true.}

end.

```

for, яки параметрлы цикл

for циклының кабатланулар саны алдан ук билгеләнелә.
for'ның ике варианты була, әйдәгез әле, аларны карыйк:

Беренче вариант:

```
for <санагыч> := <кыйммәт1> to <ахыргы_кыйммәт> do <оператор1>;
```

Һәр итерациядән (кабатланудан) соң санагычның кыйммәте 1гә арта.

<кыйммәт1> — санагычның башлангыч кыйммәте. Аның урынында сан яки үзгәрешленең исеме языла.

<кыйммәт1> ның кыйммәте тән артык булгач та, цикл тәмамлана.

Әгәр циклының тәнәндә бердән артык оператор бар, алар begin һәм end эчендә языла.

<санагыч>, <кыйммәт1> һәм <ахыргы_кыйммәт> — бөтен сан типтагы булырга тиеш.

Еш кына программчылар циклының санагыч үзгәрешлегә i исеме бирәләр.

Икенче вариант

```
for <санагыч> := <кыйммәт2> downto <ахыргы_кыйммәт> do <оператор1>;
```

Беренче варианттан ул шулар белән аерылып тора:

Һәр итерациядән соң санагычның кыйммәте 1гә кими.

<кыйммәт2> ның кыйммәте <ахыргы_кыйммәт> тән ким булгач та, цикл тәмамлана.

Ике мөһим искәрмә!

Санагычның кыйммәте [башлангыч_кыйммәт; ахыргы_кыйммәт] кисемтәдә булганда гына, цикл кабатлана.

Цикл эчендә санагычның кыйммәтен үзгәртсәң ярамый, компилятор шуңа рөхсәт бирми.

Беренче мәсьәләнең for ярдәмендә чишелеше:

```

program for_example;
var

i, N: integer;
begin

read(N);
for i := 1 to N do write(i, ' ');
end.

```

Бу программага, мәсәлән, 10 кертсәң, ул экранга шуны чыгарыр:

1 2 3 4 5 6 7 8 9 10

Үзегез сынап карагыз)

Әлеге код, алдагыларга караганда, уңайлырак һәм кыскарак. Шулаймы? Һәм цикл for гадәти цикл түгел — аның логик шарты юк. Шуңа күрә, программалаштыруда параметрлы циклы «синтаксик шикәр» дип атыйлар. “Синтаксик шикәр” дип аталган синтаксисның үзлекләре яңа мөмкинлекләр бирмиләр, ләкин кеше өчен программалауны уңайлырак итәләр. Ә хәзер, әйдәгез, берничә мәсьәлә чишик.

2 мәсьәлә. Сөзгә ике бөтен сан бирелгән: A һәм B ($A < B$). A һәм B арасында (A һәм B ны да исәпкә алабыз) урнашкан саннарны экранга чыгарыгыз. Бу саннар санын — N да экранга чыгарыгыз.

$A < B$ шарт сәбәпле, for'ның беренче вариантын кулланабыз. A 'дан B 'га кадәр урнашкан саннар саны $N = B - A + 1$ була.

```
program mesele2;
var

A, B, count, i: integer;
begin

read(A, B);
for i := A to B do write(i, ' '); {саннарны арту тәртибендә экранга чыгарабыз}
count := B - A + 1; {шул саннар санын исәплибез}
writeln;
write( 'Саннар саны = ', count);
end.
```

3 мәсьәлә. Бөтен сан N ($N > 0$) бирелгән. Әгәр ул 3'нең дәрәжәсе икән, True әйтергә, юк икән — False.

Шулай итәбез: N өчкә калдыксыз бүленгәндә, аны өчкә бүлүне кабатлыйбыз. Аннары, әгәр $N = 1$ икән — ул 3'нең дәрәжәсе булган; Әгәр $N \neq 1$ — N саны 3'нең дәрәжәсе булмаган. Әлеге мәсьәләне чишү өчен div, mod һәм логик аңлатмалар ничек эшлэгәннән белергә кирәк.

```
program mesele3;
var

N: integer;
begin

readln(N);
while N mod 3 = 0 do N := N div 3; {N өчкә калдыксыз бүленгәндә, аны өчкә бүлүне
кабатлыйбыз}
writeln(N = 1); {логик аңлатма}
end.
```

4 мәсьәлә. Сөзгә ике бөтен сан бирелә: A һәм B , ($A < B$).

$[A, B]$ кисемтәдә ничә гади сан бар? Программа бу кисемтәдә урнашкан гади саннар санын исәпләргә һәм экранга чыгарырга тиеш.

Ә нәрсә ул гади сан? — дип сорарсыз сез. Гади сан ул — үзенә һәм 1'гә бүленә торган сан.

Махсус очраклар бар: 0 һәм 1 — гади саннар түгел, 2 — гади сан.

```
program mesele4;
var

A, B, javap, san, buluce: integer;
gadi: boolean;
begin

read(A, B);
javap := 0; {гади саннар саны башта 0}
for san := A to B do {A дан башлап B га кадәр саннарны тикшерәбез}
begin

if san <= 1 then {махсус очрак}

gadi := False
else if san = 2 then {махсус очрак}

gadi := True
else if san 2 then

{бүлүчеләр эзлибез. Беләсезме ни өчен без кв.тамырга гына кадәр? Чөнки бер
теорема бар, аның буенча безгә шул кадәр тикшерергә житә}
for buluce := 2 to trunc(sqrt(san)) + 1 do

if san mod buluce = 0 then {әгәр сан үзеннән һәм 1 башка санга бүленә икән, ул
гади түгел}
begin

gadi := False;
break;{эчке for'дан чыгабыз}
end;

if gadi then javap := javap + 1; {әгәр сан гади икән, җавапны 1гә арттырабыз}

end;
write('Бирелгән кисемтәдә гади саннар саны = ', javap);

end.
```

Создайте бесплатный сайт или блог на WordPress.com. (https://wordpress.com/?ref=footer_website) Тема: Big Brother, автор: WordPress.com (<http://automattic.com>).