

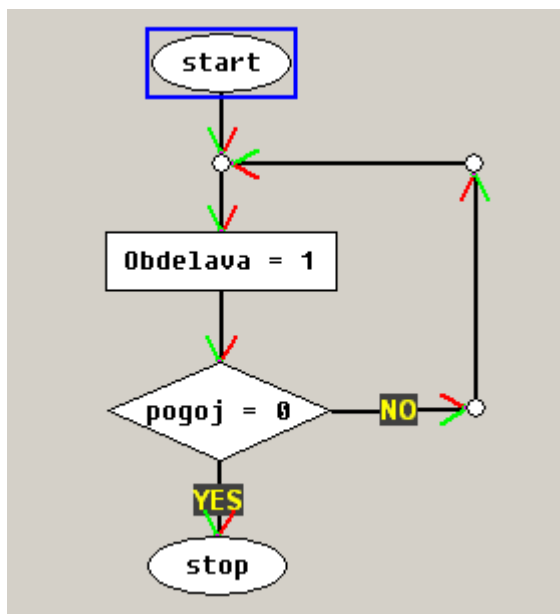
KRMILNI STAVKI

Krmilni stavki v pascalu so:
REPEAT UNTIL stavek,
WHILE DO stavek in
FOR stavek.

REPEAT-UNTIL STAVEK

Repeat until stavku pravimo zaradi ponavljanja tudi zanka.

Grafično prikazano:



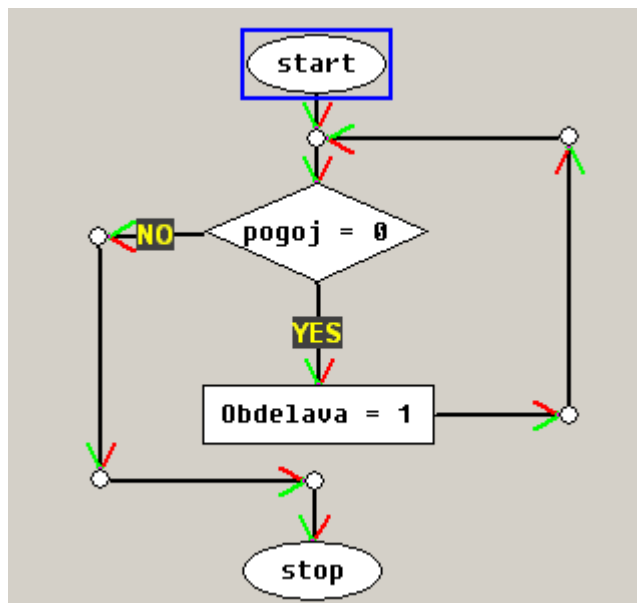
Pascalska oblika:

```
begin  
  i := 0;  
  repeat  
    write(i:2);  
    i := i + 5;  
  until i > 20;  
end.
```

Stavek obdelava se izvede najmanj enkrat. Iz zanke pademo takrat, ko je pogoj izpolnjen.

WHILE-DO ZANKA

Grafično prikazano:



Pascalska oblika:

```
begin
i := 0;
while i <= 20 do
begin
write(i:4);
i := i + 5;
end;
end.
```

Repeat unzil zanko lahko vedno nadomestimo z WHILE DO zanko.

Primer enake zanke:

```
I := 0;
repeat
write(i:4);
i := i + 5;
until i > 20;
```

```
I := 0;
while i <= 20 do
begin
write(i:4);
i := i + 5;
end;
```

Zgornji dve zanki sta popolnoma enaki. Razlika je samo v tem, da je ena zanka z REPEAT UNTIL stavkom, druga pa z WHILE DO stavkom. Sam problem nam določa, za katero zanko se odločimo. Za while do zanko se navadno odločamo takrat, ko imamo problem, kateri ne potrebuje, da se zanka vsaj enkrat izvede (primer branja iz datoteke – če ni podatkov ne beremo), ko pa imamo problem, ko hočemo, da se zanka izvede enkrat ali pa večkrat, takrat uporabimo pa repeat until zanko (primer izpis menija – meni vedno izpišemo, čeprav se odločimo, da program zaključimo).

FOR STAVEK

For zanka je enostavni krmilni stavek.

ZGLED: imamo interval števil od 1 do 10.

Z for zanko zapišemo ta interval na naslednji način:

```
FOR i := 1 TO 10 DO
```

Pravilo zapisa for zanke:

rezervirana beseda	spremenljivka		začetna vrednost	rezervirana končna beseda vrednost	rezervirana beseda
FOR	i	:=	x	TO y	DO
				DOWNTO	

Zgledi:

```
FOR i := 0 to 1000 do
```

```
for i := 1 to 10 do  
begin  
  for j := 2*i downto 1 do  
    write(j);  
  writeln;  
end;
```

```
for ch := 'A' to 'Z' do  
  write(ch:2);
```

```
for b := false to true do  
  write(ord(b):4);
```

Program za izpisovanje vrednosti kotne funkcije sinus na intervalu od 1 stopinja do 90 stopinj.

Program sinus;

```
uses crt;
```

```
var
```

```
  r:real;
```

```
  i,x:integer;
```

```
begin
```

```
  clrscr;
```

```
  for i := 1 to 5 do
```

```
  begin
```

```
    write('    x        sin(x)');
```

```
    writeln;
```

```
  end;
```

```
  r := pi/180;
```

```
  for x := 1 to 90 do
```

```
    write(x:6,sin(x*r):10:5);
```

```
  readln;
```

```
end.
```

Continue in break ukaza

Continue in break nam predčasno ustavita izvajanje zanke. Ukaza delujeta enako v vseh treh zankah, ki jih programski jezik pascal pozna. Poglejmo si razliko na primeru.

Primer:

```
for i := 1 to 10 do
begin
  if i = 5 then
    continue;
  write(i, ' ');
end;
```

v tem primeru nam program izpiše: 1 2 3 4 6 7 8 9 10

Ko uporabimo stavek continue nam program preskoči izvajanje zanke in nadaljuje z izvajanjem naslednje iteracije zanke.

```
for i := 1 to 10 do
begin
  if i = 5 then
    break;
  writeln(i, ' ');
end;
```

v drugem primeru nam program izpiše: 1 2 3 4

Z uporabo break stavka nam program permanentno prekine izvajanje zanke. Program se nadaljuje z stavki za zanko.

NALOGE

1. Izdelaj program, ki z uporabo for zanke izdela križec na sredini zaslona
2. Izdelaj program, ki z uporabo for zanke izdela kvadrat na sredini zaslona
3. Izdelaj program, ki z uporabo for zanke izdela krog na sredini zaslona
4. Izdelaj programe, ki z uporabo repeat until, while do zanke izdela križec, kvadrat in krog na sredini zaslona.