

## Naredbe uslovnog grananja

Rešavanje većine problema često zahteva izvršavanje određenog dela programa u zavisnosti od ispunjenosti nekog uslova. Iz tog razloga neophodno je postojanje razgranatih struktura, koje se u programskom jeziku Pascal ostvaruju korišćenjem naredbe uslovnog grananja **if** i naredbe višestrukog grananja **case**.

### Naredba uslovnog grananja – IF

Naredba uslovnog grananja **if** omogućava izvršavanje određenog dela programa u zavisnosti od toga da li je navedeni uslov ispunjen ili ne. Ukoliko je uslov ispunjen, tj. ukoliko je vrednost logičkog iskaza **tačno (true)**, onda se izvršava jedna naredba ili blok naredbi navedenih unutar komande **if**. Ukoliko je vrednost logičkog izraza **netačno (false)**, može se izvršiti neka druga naredba, odnosno blok naredbi.

#### Sintaksa

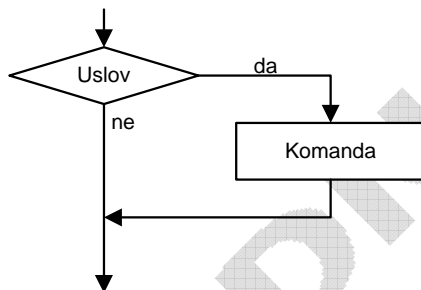
```
if <logicki_izraz> then <naredba1> [ else <naredba2> ];
```

Korišćenjem ovakve **if – then – else** konstrukcije moguće je formirati više oblika uslovnog grananja, kao što su: jednostruko, dvostruko i višestruko grananje.

#### Jednostruko grananje

Kod jednostrukog grananja računar ispituje vrednost logičkog izraza i ukoliko je ova vrednost **tačno (true)**, izvršava se naredba, odnosno blok naredbi iza rezervisane reči **then**. Ukoliko je vrednost izraza **netačno (false)**, ova naredba se ne izvršava.

Na slici je prikazan šematski prikaz ovakvog grananja:



Slika ### Šematski prikaz jednostrukog grananja

Pascal kod koji bi odgovarao ovakvoj vrsti grananja bi bio:

```
if logicki_izraz then naredba;
```

ili

```
if logicki_izraz then  
  begin  
    naredba_1;  
    naredba_2;  
    ...  
    naredba_n;  
  end;
```

### Primer

Napisati program koji na osnovu broja utrošenih kilovata električne energije  $P$  i cene po kilovatu  $C$ , izračunava ukupnu naknadu koju potrošač treba da plati  $T$ . Ukoliko je naknada veća od 1000 dinara, račun se uvećava za 10%, kako bi se stimulisala štednja električne energije.

```

program Struja;
const  LIMIT=1000.0;  {Limit iznad koga se uvecava naknada}
       PENALI=10.0;  {Procenat za koji se uvecava naknada}
var    P,C,T:real;
begin
  writeln('Unesite broj utrosenih kilovata P:');
  readln(P);
  writeln('Unesite cenu po kilovatu C:');
  readln(C);

  T:=P*C;

  if T>LIMIT then T:=T+T*PENALI/100.0;

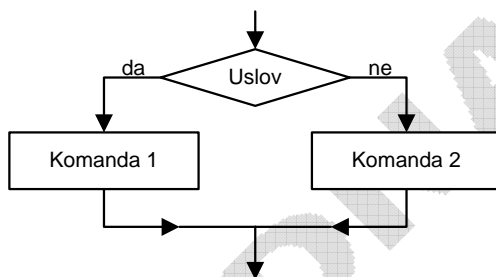
  writeln('Vas racun za elektricnu energiju je ',T,' dinara.');
```

end.

### Dvostruko grananje

U slučaju dvostrukog grananja, ukoliko je logički izraz tačan izvršava se naredba ili blok naredbi iza rezervisane reči **then**, a ako logički izraz nije tačan izvršava se naredba, odnosno blok naredbi, koji se navode iza rezervisane reči **else**.

Šematski prikaz koji opisuje ovakvu vrstu grananja prikazan je na slici:



Slika ### Šematski prikaz dvostrukog grananja

Pascal kod koji bi odgovarao dvostrukom grananju bi bio:

```

if logički_izraz then naredba_1
else naredba_2;

ili

if logički_izraz then
begin
  naredba_11;
  naredba_12;
  ...
  naredba_1n;
end
else
begin
  naredba_21;
```

```

naredba_22;
...
naredba_2m;
end;

```

### Primer

Napisati program koji za  $x$  dato na ulazu izračunava vrednost funkcija  $f(x)$  i  $g(x)$  koje imaju oblik:

$$f(x) = \begin{cases} -1, & \text{ako je } x < 0 \\ 2x+3, & \text{ako je } x \geq 0 \end{cases} \quad \text{i} \quad g(x) = \begin{cases} \ln(-x), & \text{ako je } x < 0 \\ e^x, & \text{ako je } x \geq 0 \end{cases}$$

```

program Funkcije;
var x,f,g:real;
begin
  writeln('Unesite x:');
  readln(x);

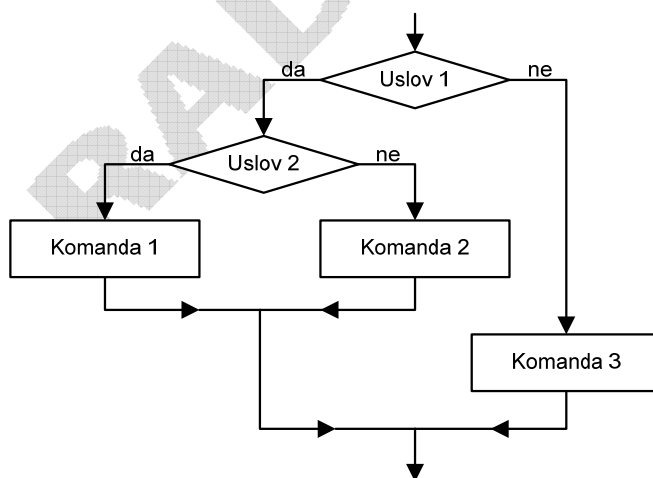
  if x<0 then
    begin
      f:=-1.0;
      g:=ln(-x);
    end
  else
    begin
      f:=2.0*x+3.0;
      g:=exp(x);
    end;

  writeln('f(x)=',f);
  writeln('g(x)=',g);
end.

```

### Višestruko grananje

Imajući u vidu da se unutar **if-then-else** konstrukcije mogu naći bilo koja naredba ili blok naredbi, zaključujemo da to mogu biti i druge **if** naredbe. Korišćenjem takvih konstrukcija možemo ostvariti višestruko grananje, kao što je to prikazano na slici:



Slika ### Šematski prikaz višestrukog grananja

U ovom slučaju program prvo ispituje da li je prvi uslov ispunjen. Ukoliko *Uslov 1* nije ispunjen, program izvršava *Komandu 3*. Međutim, ako je *Uslov 1* ispunjen, program ispituje *Uslov 2* i u zavisnosti od toga da li je on ispunjen ili nije, program izvršava *Komandu 1* ili *Komandu 2*. Na ovaj način dobili smo algoritamsku strukturu sa trostrukim grananjem, koja može izvršavati neku od tri zadate komande u zavisnosti od ispunjenosti pojedinih uslova. Na sličan način, kombinacijom **if** i **else** naredbi, možemo dobiti proizvoljan broj uslovnih grananja.

### Primer

Napisati program koji na osnovu broja bodova osvojenih na ispitu, određuje ocenu prema sledećoj tabeli:

Broj bodova	Ocena
55-64	6
65-74	7
75-84	8
85-94	9
95-100	10

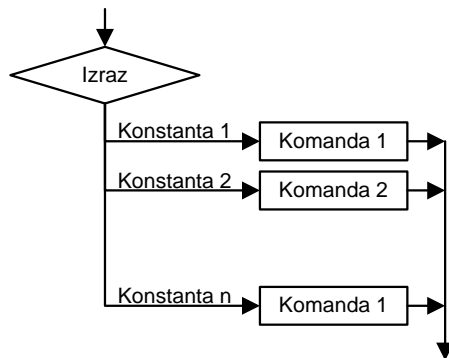
```
program Ocena;
var bodovi,ocena:integer;
begin
  writeln('Unesite broj bodova:');
  readln(bodovi);

  if bodovi>=95 then ocena:=10
  else
    if bodovi>=85 then ocena:=9
    else
      if bodovi>=75 then ocena:=8
      else
        if bodovi>=65 then ocena:=7
        else
          if bodovi>=55 then ocena:=6
          else ocena:=5;

  writeln('Student je dobio ocenu ',ocena);
end.
```

### Naredba višestrukog grananja – CASE

Iako se višestruko grananje može ostvariti korišćenjem naredbe **if**, program pisan na takav način je često veoma nepregledan. Da bi se to izbeglo, u programskom jeziku Pascal postoji naredba višestrukog grananja **case**, koja omogućava grananje programa u zavisnosti od vrednosti nekog izraza, kao što je to prikazano na slici:



Slika ### Šematski prikaz višestrukog grananja pomoću naredbe case

Iza naredbe **case** navodi se izraz koji može biti bilo kog nabrojivog tipa (char, boolean, integer, skup ili intervalni tip). Nakon izraza sledi rezervisana reč **of** iza koje se nabrajaju moguće vrednosti izraza i odgovarajuće naredbe ili blokovi naredbi.

### Sintaksa

```
case <izraz> of
  < konstanta_1: naredba_1; >
  [ konstanta_2: naredba_2; ]
  ...
  [ konstanta_n: naredba_n; ]
end;
```

Napomenimo da umesto jedne konstante može biti navedena i lista više konstanti odvojenih zarezom. U tom slučaju naredba se izvršava ukoliko je navedeni izraz jednak bilo kojoj od nabrojanih konstanti.

**Case** naredba funkcioniše tako što se prvo izračunava vrednost izraza, a zatim traži konstanta koja je jednaka vrednosti izraza. Ukoliko takva konstanta postoji, izvršava se naredba navedena iza konstante. Ako takve konstante nema, onda se ili nastavlja izvršavanje programa zanemarujući naredbu **case** ili program prijavljuje grešku, u zavisnosti od vrste kompajlera. Turbo Pascal omogućava korišćenje rezervisane reči **else** unutar naredbe **case** iza koje se navodi naredba koja će se izvršiti u slučaju da vrednost izraza nije jednaka ni jednoj od navedenih konstanti.

### Primer 1

Napisati program koji na osnovu ocene od 1 do 5, određuje opisnu ocenu koja može biti "nedovoljan", "dovoljan", "dobar", "vrlo dobar" ili "odličan".

```
program Ocene;
var ocena:integer;
begin
  writeln('Unesite ocenu:');
  readln(ocena);

  write('Ucenik je ');
  case ocena of
    1: writeln('nedovoljan');
    2: writeln('dovoljan');
    3: writeln('dobar');
    4: writeln('vrlo dobar');
    5: writeln('odlican');
  end;
end.
```

**Primer 2**

Napisati program koji određuje broj dana u zadatom mesecu.

```
program Dani;
var  mesec:integer;
     prestupna:char;
begin
  writeln('Unesite redni broj meseca:');
  readln(mesec);

  case mesec of
    1,3,5,7,8,10,12: writeln('Mesec ima 31 dan. ');
    4,6,9,11:      writeln('Mesec ima 30 dana. ');
    2:
      begin
        write('Da li je godina prestupna (D/N)? ');
        readln(prestupna);
        if (prestupna='D') or (prestupna='d') then
          writeln('Mesec ima 29 dana. ')
        else
          writeln('Mesec ima 28 dana. ')
        end;
      else writeln('Pogresan broj meseca. ');
    end;
  end.
end.
```

NAPOMENA: Prethodni primer neće raditi na kompajlerima koji ne podržavaju rezervisanu reč **else** unutar naredbe **case**.