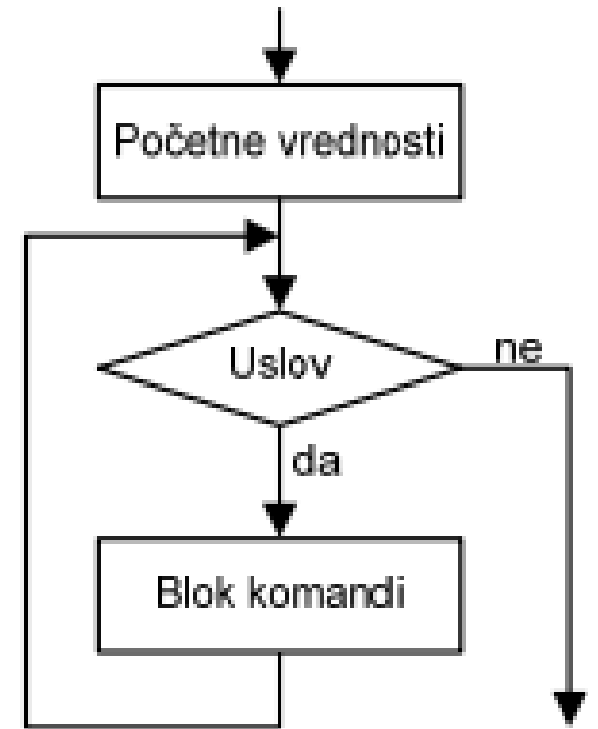


Naredbe ciklusa

**WHILE NAREDBA**  
**(NAREDBA SA PREDUSLOVOM)**

- Da bi se koristila naredba for neophodno je znati tačan broj ponavljanja pre početka izvršavanja petlje. Pored toga, za vreme izvršavanja petlje nije moguće menjati početnu i krajnju vrednost brojačke promenljive. Međutim, veoma često je slučaj da određeni deo programa želimo ponavljati sve dok je neki uslov ispunjen. Na taj način iteracije se mogu ponavljati manji ili veći broj puta u zavisnosti od rezultata izvršavanja naredbi unutar petlje.
- Jedna od naredbi koja omogućava realizaciju ovakvih ciklusa je naredba while. Osnovna karakteristika while petlje je da je to petlja sa preduslovom, što znači da se ispunjenost uslova proverava pre izvršavanja svake iteracije. Posledica toga je da je moguće da se naredbe unutar petlje ne izvrše ni jednom, ukoliko uslov u startu nije ispunjen.

- Naredba `while` omogućava ponavljanje određenog dela programa **sve dok je navedeni uslov ispunjen**.
- Na početku svake iteracije proverava se da li je uslov ispunjen, tj. da li je vrednost logičkog iskaza tačno (`true`), a zatim, ukoliko je jeste, izvršava se jedna naredba ili blok naredbi navedenih unutar komande `while`. Nakon izvršenja naredbi unutar `while` petlje otpočinje nova iteracija i nova provera uslova. U trenutku kada na početku neke iteracije vrednost logičkog izraza postane netačno (`false`), prekida se izvršavanje petlje i nastavlja se izvršavanje ostatka programa.



# Sintaksa

**while** <logicki\_izraz> **do** <naredba>;

Naredba WHILE nema mogućnost automatske dodele nove vrednosti promenljivoj, pa će sintaksa ove naredbe češće biti:

**while** <logicki\_izraz> **do**

begin

<naredba1>;

[naredba2];

[naredba3];

.....

end;

**while** <logicki\_izraz> **do**

begin

<naredba1>;

[naredba2];

[naredba3];

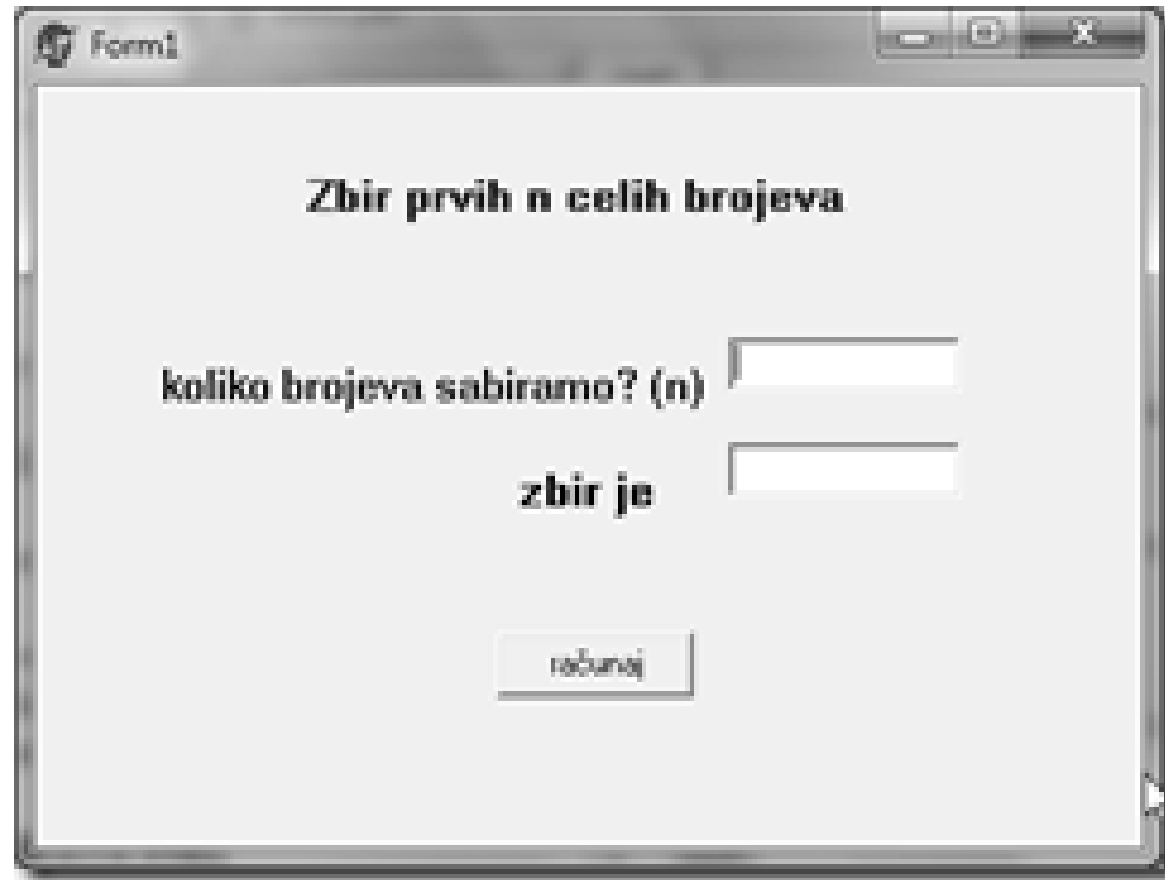
.....

**brojac := brojac + 1;**

end;

**PRIMER 1.** Sastaviti aplikaciju koja će računati zbir prvih  $n$  prirodnih brojeva.

npr. za uneto  $n=3$  računa se  $s=1+2+3$



The image shows a screenshot of a Windows application window. The title bar reads "Form1". The main content area has the text "Zbir prvih n celih brojeva" centered at the top. Below this, there are two input fields. The first is labeled "koliko brojeva sabiramo? (n)" and the second is labeled "zbir je". At the bottom center, there is a button labeled "računaj".

## Procedure TForm1.Button1Click (Sender:Tobject);

Var n,s,i: integer;

BEGIN

n:=StrToInt (edit1.text);

S:=0;

**i:=1;**

**while i <= n do**

**begin**

**S:=S+i;**

**i:=i+1;**

**end;**

edit2.text:=IntToStr(s);

end;

# NAPOMENA:

- While petlja mora biti tako napisana da garantuje da će u konačnom broju iteracija navedeni logički izraz postati netačan (false). Na taj način obezbeđuje se mehanizam izlaska iz petlje nakon konačnog broja iteracija. (pazite da ne zaboravite ručno uvećavanje brojača, npr liniju  $i:=i+1$ )
- Ukoliko logički izraz nikada ne bi dobio vrednost false došlo bi do beskonačnog broja ponavljanja (tzv. mrtva petlja), odnosno do blokade izvršenja ostatka programa.

## PRIMER 2. Kreirati aplikaciju koja sabira sve parne brojeve u interalu od 1 do n.

npr. za uneto  $n=10$  računa se  $s=2+4+6+8+10$

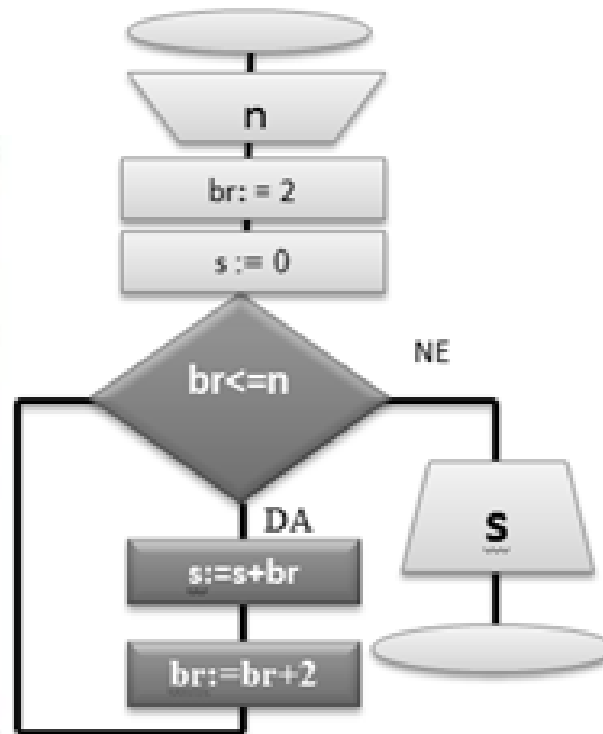
### REŠENJE 1

Form1

Aplikacija sabira sve parne brojeve u interalu od 1 do n

n

zbir





```
procedure TForm1.Button1Click(Sender: TObject);  
var n,br,s:integer;  
begin  
  n:=strtoint(edit1.text);  
  br:=2;  
  s:=0;  
  while br<=n do  
  begin  
    s:=s+ br;  
    br:=br+2  
  end;  
  edit2.text:=inttostr(s);  
end;
```

## REŠENJE 2

```
Procedure TForm1.Button1Click (Sender:Tobject);
Var n,s,i: integer;
BEGIN
  n:=StrToInt (edit1.text);
  S:=0;
  i:=1;
  while i <= n do
    begin
      if i mod 2 = 0 then S:=S+i;
      i:=i+1;
    end;
  edit2.text:=IntToStr(s);
end;
```

10

### PRIMER 2

Kreirati aplikaciju koja sabira sve parne brojeve u intervalu od 1 do n.

# PRIMER 3

Sabirati redom prirodne brojeve dok suma  $S$  ne dostigne granicu  $G$ . Prikazati sumu i broj koji smo poslednji dodali.

11

The screenshot shows a Windows application window titled "Form1". Inside the window, there is a user interface for a simple calculator. At the top left, there is a label "Granica-G" in blue text, followed by a text input box containing the number "5". Below this is a button labeled "Racunaj" with a dotted border. Further down, there is a label "Suma" in blue text, followed by a text input box containing the number "6". At the bottom, there is a label "broj i=" in blue text, followed by a text input box containing the number "3". The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

## REŠENJE

```
procedure TForm1.Button1Click(Sender: TObject);
  var G,i:integer;
      S:real;
begin
  G:=StrToInt(Edit1.Text);
  S:=0;
  i:=0;
  While S<=G do
    Begin
      S:=S+i;
      i:=i+1;
    end;
  Edit2.Text:=FloatToStr(S);
  Edit3.Text:=Inttostr(i);
end;
```

12

### PRIMER 3

Sabirati redom prirodne brojeve dok suma S ne dostigne granicu G. Prikazati sumu i broj koji smo poslednji dodali.

# Domaći zadatak

(zadatke rešiti koristeći While naredbu)

**ZADATAK 1:** Kreiraj aplikaciju koja računa sumu prvih N prirodnih brojeva koji su deljivi sa 3

**ZADATAK 2:** Kreiraj aplikaciju koja računa sumu prvih N prirodnih brojeva koji na mestu jedinica imaju cifre 0 ili 5

**ZADATAK 3:** Sastavi aplikaciju koja će izračunati **zbir recipročnih vrednosti** brojeva u intervalu od a do b

**ZADATAK 4:** Sastavi aplikaciju koja računa n!

$$n! = n * (n-1) * (n-2) * \dots * 1$$