



**NAREDBE CIKLUSA**

**REPEAT...UNTIL**

# Naredba ciklusa sa POSTUSLOVOM

- ▶ Za razliku od while petlje, naredba repeat omogućava realizaciju ciklusa sa postuslovom.
- ▶ Ciklus sa postuslovom podrazumeva to da se uslov za dalje izvršavanje petlje proverava na kraju svake iteracije. Na ovaj način naredbe unutar petlje će biti izvršene bar jednom, bez obzira na to da li je uslov bio ispunjen pre ulaska u petlju.

# Ponavljaj... dok se uslov ne ispuni

- ▶ Naredba `repeat` omogućava ponavljanje određenog dela programa sve dok navedeni uslov nije ispunjen.
- ▶ To praktično znači da je (za razliku od `while` ciklusa) navedeni uslov u stvari "uslov za izlazak iz petlje".
- ▶ U slučaju `repeat` petlje, izvršava se jedna naredba ili blok naredbi navedenih unutar komande `repeat`.
- ▶ Nakon izvršenja ovih naredbi vrši se provera uslova navedenog iza rezervisane reči `until`.
- ▶ Ukoliko je vrednost logičkog izraza netačno (`false`), kreće se u izvršavanje sledeće iteracije. U suprotnom prekida se izvršavanje petlje i nastavlja se izvršavanje ostatka programa.

# SINTAKSA

## repeat

<naredba\_1>;

[naredba\_2];

...

**[naredba\_n]** \*\*\* bez znaka ;

**until** <logicki\_izraz>;

Primetimo i to da u slučaju repeat naredbe, rezervisane reči repeat i until istovremeno predstavljaju i graničnike petlje, tako da je upotreba rezervisanih reči **begin** i **end** za označavanje bloka naredbi **nepotrebna**.

# NAPOMENA:

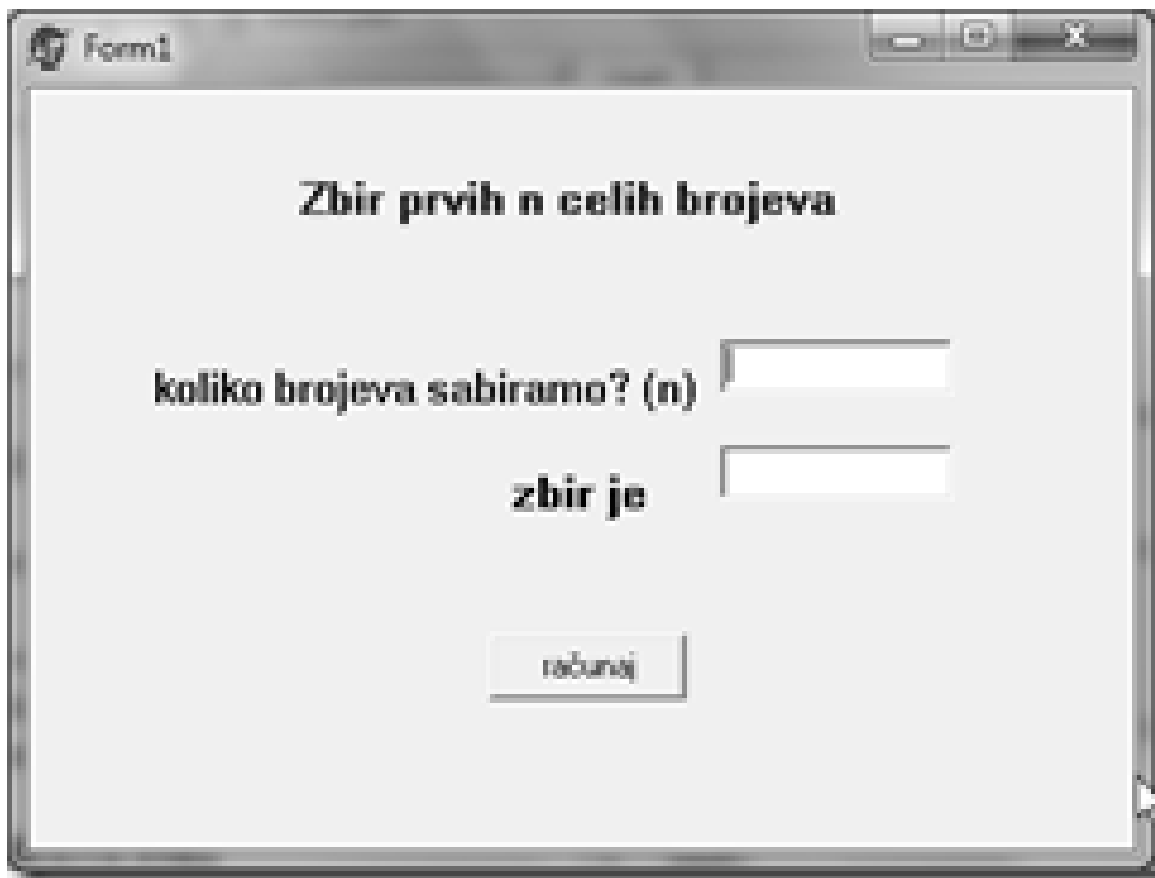
- ▶ Repeat petlja mora biti tako napisana da garantuje da će u konačnom broju iteracija navedeni logički izraz postati tačan (true).
- ▶ Na taj način obezbeđuje se mehanizam izlaska iz petlje nakon konačnog broja iteracija. Ukoliko logički izraz nikada ne bi dobio vrednost true došlo bi do beskonačnog broja ponavljanja (tzv. **MRTVA PETLJA**), odnosno do blokade izvršenja ostatka programa.

# ZADACI

## PRIMER 1

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

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



Form1

**Zbir prvih  $n$  celih brojeva**

koliko brojeva sabiramo? (n)

zbir je

računaj

## REŠENJE 1

```
Procedure TForm1.Button1Click (Sender:Tobject);  
Var n,s,i: integer;  
BEGIN  
  n:=StrToInt (edit1.text);  
  S:=0;  
  i:=1;  
  REPEAT  
    S:=S+i;  
    i:=i+1  
    UNTIL i > n;  
  edit2.text:=IntToStr(s);  
end;
```

Ovaj deo se razlikuje u odnosu na rešnje sa WHILE naredbom.

Uslov se proverava na kraju ciklusa.



## PRIMER 2

Sastaviti aplikaciju koja će računati zbir prvih  $n$  NEPARNIH prirodnih brojeva.

npr. za uneto  $n=7$  računa se  $s=1+3+5+7$

The screenshot shows a Windows application window titled "Form1". The window contains the following elements:

- A title bar with standard Windows window controls (minimize, maximize, close).
- A title "Zbir prvih n NEPARNIH prirodnih brojeva" centered at the top.
- A label "Unesite N:" followed by a text input field labeled "Edit1".
- A button labeled "Računaj" to the right of the input field.
- A label "Zbir je" followed by a text label field labeled "Label4".
- A button labeled "Izlaz" to the right of the label field.

## REŠENJE 2

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

```
  Var n,s,i:integer;
```

```
BEGIN
```

```
  n:=StrToInt (edit1.text);
```

```
  S:=0;
```

```
  i :=1;
```

```
  Repeat
```

```
    if i mod 2 =1 then S:=S+i;
```

```
    i := i + 1
```

```
  until i > n;
```

```
  Label4.Caption:=IntToStr(s);
```

```
END;
```

## PRIMER 3

Sastaviti aplikaciju koja će računati zbir prirodnih brojeva u intervalu  $[A,B]$ .

11

The screenshot shows a Windows application window titled "Form1". The window contains a grid of controls. At the top, there is a title bar with standard Windows window controls (minimize, maximize, close). Below the title bar, the text "Aplikacija koja računa zbir brojeva u intervalu [A,B]" is displayed. The main area of the window contains the following elements:

- A label "Unesite A:" followed by a text box labeled "Edit1".
- A label "Unesite B:" followed by a text box labeled "Edit2".
- A button labeled "Računaj" (Calculate).
- A button labeled "Izlaz" (Exit).
- A label "Zbir je" (Sum is) followed by a label "Label5".

## REŠENJE 3

12

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

Var A,B,s,i :integer;

**BEGIN**

A:=StrToInt (edit1.text);

B:=StrToInt (edit2.text);

S:=0;

**i := A;**

**repeat**

**S:=S+i;**

**i := i+1**

**until i < B ;**

Label5.Caption:=IntToStr(s);

**END;**

## PRIMER 4

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

13

The screenshot shows a Windows application window titled "Form1". The window contains a simple user interface for a calculation. It features three input fields and one button. The first input field is labeled "Granica-G" and contains the value "5". Below it is a button labeled "Racunaj". The second input field is labeled "Suma" and contains the value "6". The third input field is labeled "broj i=" and contains the value "3". The window has a standard Windows title bar with minimize, maximize, and close buttons.

## REŠENJE 4

```
procedure TForm1.Button1Click(Sender: TObject);
  var G, I, S : integer;

begin
  G:=StrToInt(Edit1.Text);
  S:=0;
  i:=0;
  REPEAT
    S:=S+i;
    i:=i+1
  UNTIL S>G;
  Edit2.Text:=IntToStr(S);
  Edit3.Text:=IntToStr(i);
end;
```

# Domaći zadatak

(zadatke rešiti koristeći REPEAT naredbu)

**ZADATAK 1:** Kreiraj aplikaciju koja računa sumu prvih N prirodnih brojeva koji su deljivi sa 4

**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!

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