

Osnove programiranja

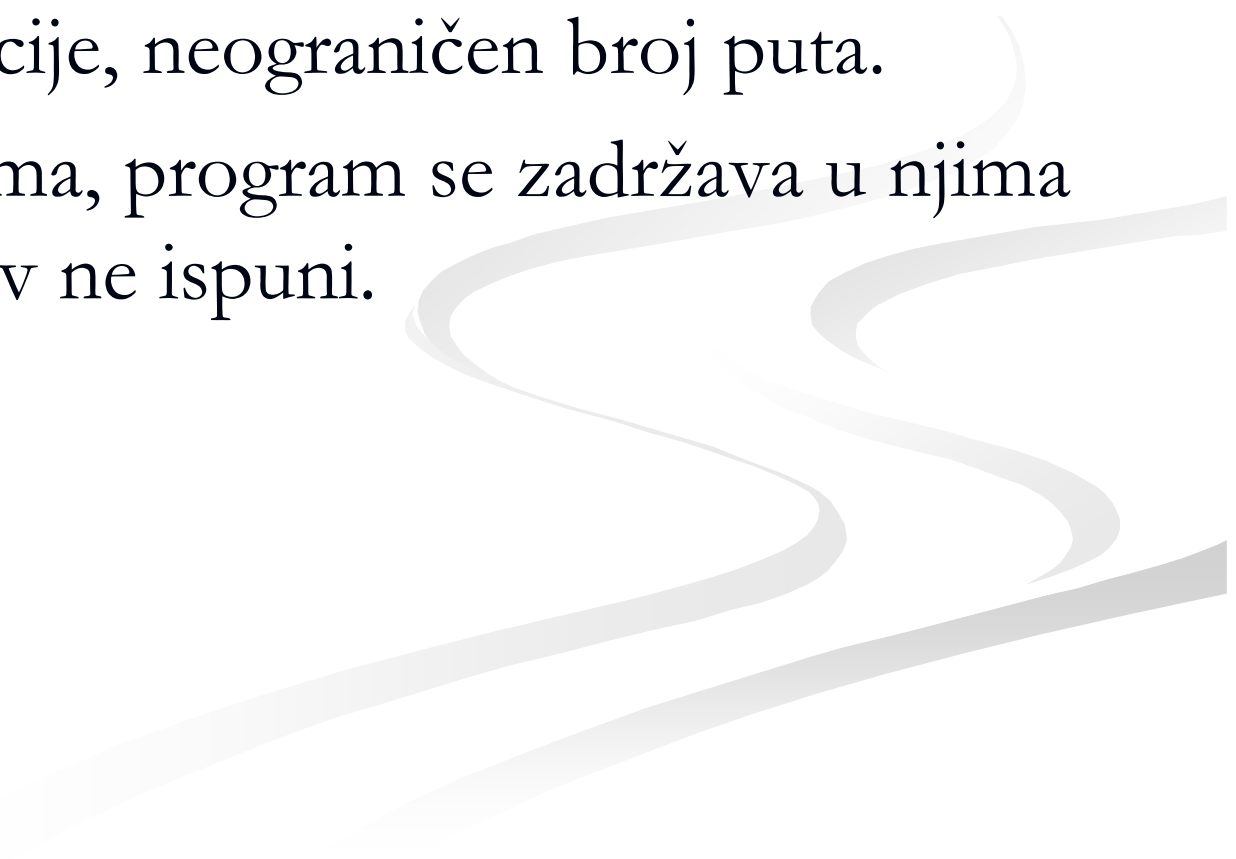
Petlje

The slide features a light blue gradient background. The main title 'Osnove programiranja' is centered in a large, bold, blue serif font. Below it, the word 'Petlje' is centered in a smaller, bold, black serif font. On the right side of the slide, there are several thick, light gray wavy lines that curve downwards and to the right, creating a decorative, flowing effect.

Sadržaj

- Šta su petlje?
- Petlje `while`
- Petlje `do while`
- Petlje `for`
- Prekidanje petlji: naredbe za skok
 - `break`, `continue`, `goto`, `return`

Definicija

- **Petlje** predstavljaju ponavljanje izvršenja iskaza.
 - Ova tehnika omogućava ponavljanje, jednom napisane operacije, neograničen broj puta.
 - U nekim petljama, program se zadržava u njima sve dok se uslov ne ispuni.
- 
- A decorative graphic consisting of several overlapping, wavy, light gray lines that flow from the bottom left towards the top right, positioned in the lower right quadrant of the slide.

Petlja while

- Petlja **while** ponovo izvršava telo koda sve dok je logički izraz istinit.
- Izraz se ispituje **pre** nego što se izvrši telo petlje (telo petlje ne mora da se izvrši).

```
int i = 0;
```

```
while (i < 3)
```

```
{
```

```
    Console.Write(i+" ");
```

```
    i++;
```

```
}
```

→ 0 1 2

Primer 1

- Izračunati sumu prirodnih brojeva od 1 do 10.

```
int i=1, suma=0;  
while (i <= 10)
```

```
{  
    suma = suma + i;  
    i++;  
}
```

```
    Console.WriteLine("Suma prvih 10 prirodnih  
brojava je: "+ suma);
```

Petlja do-while

- Petlja **do-while** razlikuje se od petlje while samo po tome što se u njoj izraz ispituje **nakon** izvršavanja bloka naredbi (što obezbeđuje da se blok uvek izvrši bar jednom).

```
int i = 0;
```

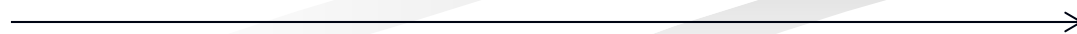
```
do
```

```
{ // Vitičaste zagrade su ovde opcione
```

```
  Console.WriteLine (i++);
```

```
}
```

```
while (i < 3);
```



0

1

2

Primer 2

- Izračunati sumu prvih n prirodnih brojeva.

```
int n, i=1, suma=0;
    Console.WriteLine("Unesite broj n:");
    n = int.Parse(Console.ReadLine());
    do
    {
        suma = suma + i;
        i++;
    }
    while (i <= n);
    Console.WriteLine("Suma prvih {0} prirodnih brojava je {1} ", n,
suma);
```

Petlja for

- Petlja **for** je slična petlji *while* s posebnim odredbama za inicijalizaciju i iteraciju promenljive petlje.
- Petlja for sadrži sledeće tri odredbe:
- `for (int i = 0; i < n; i++)`

Odredba za inicijalizaciju

(brojač) se izvršava pre početka petlje i to **samo jednom** i obično inicijalizuje jednu ili više promenljivih iteracije.

Odredba uslov je logički izraz koji se ispituje **pre** svake **iteracije** petlje.

Odredba za iteraciju

se izvršava **nakon** svake **iteracije** tela programa

Petlja for

1. *Odredba_za_inicijalizaciju* (brojač) – izvršava se pre početka petlje i to samo jednom (može imati više promenljivih).
2. *Odredba_uslov* - logički izraz koji se ispituje pre svake iteracije petlje.
 - Telo programa se izvršava sve dok je taj uslov istinit.
3. *Odredba_za_iteraciju* se izvršava **nakon svake iteracije** tela programa.
 - Obično se koristi za ažuriranje iteracione promenljive.

Ekvivalentnost koda

```
for (izraz1; izraz2; izraz3)
```

```
  izraz1;
```

```
  while (izraz2)
```

```
  {
```

```
    naredba
```

```
  izraz3;
```

```
  }
```

Primer 3

- Šta ispisuje sledeći kôd?

```
for (int i = 0; i < 3; i++)  
    Console.WriteLine (i);
```

- Napisati program za ispis velikih slova od A do Z.

```
char slovo;  
for (slovo = 'A'; slovo <= 'Z';slovo++)  
{  
    Console.WriteLine(slovo);  
}
```

Zadatak

- Napisati program koji ispisuje prvih 10 Fibonačijevih brojeva (gde je svaki broj zbir dva prethodna):

```
int i = 0, f1 = 1, f2 = 1;
while(i < 10)
{
    Console.WriteLine(f1);
    int f3 = f1 + f2;
    f1 = f2;
    f2 = f3;
    i++;
}
```

Petlja for

- Svaki od tri dela naredbe **for** može se izostaviti.

```
for (;;)
    Console.WriteLine ("prekini me");
```

```
for (int i=0; i<10;)
    //Console.WriteLine (i);
    Console.WriteLine (i++);
```

```
for (int i = 0; ; i++)
{
    Console.WriteLine(i);
    //Console.ReadLine();
}
```

Petlja foreach

- Naredba **foreach** prolazi (iterira) kroz svaki element u nabrojivom objektu.
- U jeziku C# i .NET Frameworku, većina tipova koji predstavljaju skup ili listu elemenata su nabrojivi.
 - Primer: znakovni niz.

```
class Program
{
    enum Dani
    {
        Ponedeljak, Utorak, Sreda, Cetvrtak,
        Petak, Subota, Nedelja
    }
    static void Main(string[] args)
    {
        foreach(Dani dan in
        Enum.GetValues(typeof(Dani)))
            Console.WriteLine(dan);
            //Console.WriteLine((int)dan);
            Console.ReadKey();
    }
}
```

Prekidanje petlji

- Finija kontrola nad kodom koji se izvršava u petlji obezbeđuje se pomoću četiri komande:
 - **break** - trenutno završava petlju
 - **continue** - trenutno zaustavlja ciklus petlje (izvršavanje se nastavlja od sledećeg ciklusa)
 - **goto** - dozvoljava iskakanje iz petlje na obeleženo mesto u kodu (nije preporučljivo ako želite da vam kod bude čitljiv)
 - **return** - iskače iz petlje i iz funkcije koja je sadrži.

Naredbe za skok - naredba break

- Komanda **break** jednostavno izlazi iz petlje i nastavlja izvršenje prvog reda koda posle

```
int i = 1;
while (i <=10)
{
if (i == 6)
break;
Console.WriteLine("{0}", i++);
}
```



1
2
3
4
5

Naredbe za skok – naredba `continue`

- Naredba **continue** zaustavlja samo trenutni ciklus, a ne celu petlju.
- Naredna petlja preskače broj 5:

```
int i;  
for (i = 1; i <= 10; i++)  
{  
    if (i == 5)  
        continue;  
    Console.WriteLine(i);  
}
```

1
2
3
4
6
7
8
9
10

Napravi razliku

```
int i;
for (i=1;i<=10;i++)
{
if ((i % 2) == 0)
continue;
Console.WriteLine (i) ;
}
```

```
int i;
for (i=1;i<=10;i++)
{
if ((i % 2) == 0)
break;
Console.WriteLine (i) ;
}
```

Pitanje

- Šta nije u redu sa sledećim kodom?

```
int i;  
for (i=1;i<=10;i++)  
{  
if ((i % 2) = 0)  
continue;  
Console.WriteLine (i) ;  
}
```

Zadatak

- Koju vrednost će imati promenljiva a nakon izvršavanja sledećeg C# koda?

```
int a = 10, b = 2;  
for (int i = 0; i < 3; i++)  
{  
    a = a + b + i;  
}
```

Zadatak

- Šta će biti prikazano na ekranu posle izvršenja sledećeg C# koda?

```
int a = 20, b = 10, c = 40;
do
{
    c = c - b / 2;
}
while (c > a);
if (c % 2 == 0)
    c = c * 2;
else
    c = c / 2;
Console.WriteLine(c);
```