

Uvod u programiranje

dr Ninoslava Savić

Kontrola programskog toka

Iteracija

Prekid programskog toka

for petlja

Sintaksa for petlje:

```
for (inicijalizacija; uslov; inkrement) {
    naredba;
}
```

Semantika for petlje:

1. Vrší se inicijalizacija kontrolne promenljive petlje
2. Proverava se uslov (izlazni kriterijum)
3. Ako je uslov *true*:
 1. izvršava se naredba (telo petlje)
 2. povećava se kontrolna promenljiva (inkrement)
 3. ponavlja se od koraka 2.
4. Ako je uslov *false*, prelazi se na sledeću naredbu iza petlje

2

for petlja, ugnježdjena for petlja

```
for ( int i = 0; i < n; i++ ) {
    // i dobija vrednosti od 0 do n-1
    // telo petlje će se izvršiti n puta
}
```

■ Ugnježdjena for petlja

```
for ( int i = 0; i < m; i++ ) {
    for ( int j = 0; j < n; j++ ){
```

```
/* telo petlje će se izvršiti m x n puta,
   za svaku pojedinačnu vrednost kontrolne promenljive i
   spoljašnje petlje unutrašnja petlja se izvrši kompletno
   tj. j uzme sve vrednosti od 0 do n-1 */
}
```

3

for petlja, ugnježdjena for petlja

Primer:

```
// može i ovako
```

```
int i;
for( i = 0; i < 10; i++ ){
    System.out.println(i);
}
```

```
/* ovde i nije lokalna promenljiva petlje,
   koja je definisana SAMO u telu petlje,
   već je vidljiva i izvan petlje */
```

4

Primeri - for petlja

- Napisati program koji ispisuje sve parne brojeve od 1 do 30

```
public class parniDo30{
    public static void main(String[] args){
        System.out.println
            ("Parni brojevi od 1 do 30 su:");
        for (int i = 2; i <= 30; i += 2){
            System.out.println(i);
        }
    }
}
```

5

while petlja

- Sintaksa **while** petlje

```
while(uslov){
    naredba; // blok naredba
}
```

- Semantika **while** petlje

- Proverava se uslov
 - Ako je uslov **true**, izvršava se naredba (telo petlje)
 - Ako je uslov **false**, prelazi se na sledeću naredbu posle **while**
- Moguće je da se **ni jednom** ne izvrši telo **while** petlje ako je uslov odmah **false!**

6

Primer while

```
class WhileDemo {
    public static void main(String[] args){
        int broj = 1;
        while (broj < 11) {
            System.out.print(broj + ",");
            broj++;
        }
    }
}
```

Šta će biti ispisano na kraju programa?

7

Primer while

```
class WhileDemo {
    public static void main(String[] args){
        int broj = 1;
        while (broj < 11) {
            System.out.print(broj + ",");
            broj++;
        }
    }
}
```

1,2,3,...,10

8

do while petlja

■ Sintaksa **do while** petlje

```
do {
    naredba;          // blok naredba
} while(uslov);
```

■ Semantika **do while** petlje

1. Izvršava se naredba (telo petlje)
2. Proverava se uslov
3. Ako je uslov **true**, izvršava se naredba
4. Ako je uslov **false**, prelazi se na sledeću naredbu posle do while

■ Telo **do while** petlje se izvršava **bar jednom!**

9

Primer: do while

```
class DoWhileDemo {
    public static void main(String[] args){
        int br = 1;
        do {
            System.out.print(br + ",");
            br++;
        } while (br <= 11);
    }
}
Šta će biti ispisano na kraju programa?
```

10

Primer: do while

```
class DoWhileDemo {
    public static void main(String[] args){
        int br = 1;
        do {
            System.out.print(br + ",");
            br++;
        } while (br <= 11);
    }
}
```

1,2,3,...,10,11

11

for - primer

Pitanje: Šta ispisuje ovaj program?

```
class ForDemo {
    public static void main(String[] args){
        for(int i=1; i<11; i++){
            System.out.println(i);
        }
    }
}
```

12

for - primer

Pitanje: Šta radi (ispisuje) ovaj program?

```
class ForDemo {
    public static void main(String[] args){
        for(int i=1; i<11; i++){
            System.out.println(i);
        }
    }
}
```

Odgovor: Ispisuje
1
2
3
4
5
6
7
8
9
10

13

break, continue

■ break

- izaziva bezuslovni prekid `while`, `do while` ili `for` petlje, kao i `switch` naredbe i
- skok na prvu sledeću naredbu

■ continue

- se koristiti samo u naredbama ciklusa (`while`, `do while` ili `for`)
- izaziva bezuslovni prekid tekuće iteracije i prelazak na sledeću iteraciju u petlji

14

break naredba

Sintaksa:

```
{ ..... // telo petlje
  if (uslov)
    break;
  .....
}
```

// sledeća naredba

Semantika:

- `break` izaziva bezuslovni prekid tekuće iteracije i prelazak na sledeću naredbu u programu.
- može se koristiti samo u naredbama ciklusa (`while`, `do while`, `for`) i u `switch` naredbi.

15

break naredba

Primer:

- Deo programa ispisuje redom sve elemente niza celih brojeva A dužine 100, dok ne nađe na element koji je veći od 80.

Tada nakon ispisa prekida ovu petlju i prelazi na sledeću naredbu.

```
.....
for (int i = 0; i < 100; i++) {
    if ( A[ i ] > 80 ) break;
    System.out.println( a[ i ] );
}
....
// sledeća naredba
```

16

continue naredba

Sintaksa:

```
// telo petlje
{ .....
  if (uslov)
    continue;
  .....
}
```

Semantika:

- `continue` izaziva bezuslovni prekid tekuće iteracije i prelazak na sledeću iteraciju u petlji.
- može se koristiti samo u naredbama ciklusa (`while`, `do while` ili `for`).

17

continue naredba

Primer: Dat je niz `a` koji sadrži 100 celih brojeva. Ispisati sve elemente niza koji su jednaki 0 koristeći naredbu `continue`.

// deo programa sa petljom

```
for ( int i = 0; i < 100; i++ ) {
  if ( a[i] != 0 ) continue;
  System.out.println
    ( "a[" + i + "]" = " + a[i]);
}
```

Pitanje: Koliko puta se izvršava `if` naredba, a koliko puta metod `println()`?

18

continue naredba

// deo programa sa petljom

```
for ( int i = 0; i < 100; i++ ) {
  if ( a[i] != 0 ) continue;
  System.out.println
    ( "a[" + i + "]" = " + a[i]);
}
```

Pitanje: Koliko puta se izvršava `if` naredba, a koliko puta `println()`?

Odgovor:

If će se izvršiti 100 puta, a ispis (metod `println()`) onoliko puta koliko u nizu bude 0 elemenata.

19

Pretraživanje niza – osnovni algoritam

- U datom nizu koji sadrži 100 celih brojeva naći prvu pojavu zadatog broja.

```
..... // nedostaje deklaracija niza a
int index, clan = 7;
// clan je promenljiva za broj koji se traži (7)
for (int i = 0; i < 100; i++) {
  if ( a[i] == clan ) {
    index = i;
    break;
  }
} // program nakon break-a nastavlja u ovoj tački
System.out.print("Traženi element - " + clan +
  " se javlja prvi put na " + index + " . mestu");
```

Pitanje: Čemu služi promenljiva `index` i koja vrednost će po izlasku iz petlje biti u njoj?

20

Pretraživanje niza – osnovni algoritam

Odgovor:

Promenljiva `index` pamti poziciju elementa niza koji se traži.