

2. Upravljanje tokom izvršenja

Sadržaj

- Uslovni izrazi
- Petlje
- Nizovi primitivnog tipa

Strukture unutar programa

- Sekvenca
 - niz instrukcija koje se izvršavaju jedna za drugom
- Selekcija
 - omogućava da se od više sekvenci koje se nalaze u različitim granama izvrši jedna
 - izbor sekvence koja će se izvršiti zavisi od ispunjenosti postavljenog uslova
- Iteracija
 - Sekvenca koja se može izvršiti više puta
 - Uslov izvršavanja može biti
 - Dok je zadovoljen postavljeni uslov
 - Dok se postavljeni uslov ne zadovolji

Primer sekvence

- Bezuslovno se izvršavaju sve instrukcije u nizu

```
int broj = 5;  
System.out.println("Broj ima vrednost " + broj);  
broj = broj + 3;  
System.out.println("Broj ima vrednost " + broj);
```

Instrukcije `if` i `if-else`

- Ako je uslov zadovoljen, izvršava se sekvenca

```
if (<logickiUslov>
    sekvenca1;
```

- Ako je uslov zadovoljen, izvršava se jedna sekvenca, u suprotnom se izvršava druga

```
if (<logickiUslov>
    sekvenca1;
else
    sekvenca2;
```

Instrukcija `if-else` sa više uslova grananja

```
if (<logickiUslov1>
    sekvenca1;
else if (<logickiUslov2>)
    sekvenca2;
else if (<logickiUslov3>)
    ...
else
    sekvencaN;
```

Primer selekcije

```
int broj = 4;
...
if (broj % 2 == 0)
    System.out.println("Broj " + broj + " je paran.");
else
    System.out.println("Broj " + broj + " je neparan.");
int drugiBroj = broj + 3;
System.out.println("Drugi broj je " + drugiBroj);
```

Petlja `while`

- Izvršava se sve dok je zadovoljen uslov (ima vrednost `true`)

```
while (<logickiUslov>)  
    sekvenca;
```

- Ukoliko uslov pri ulasku u petlju nije zadovoljen (ima vrednost `false`), neće se izvršiti nijedna iteracija

Primer iteracije pomoću `while` petlje

```
int broj = 4;
while (broj < 10) {
    broj = broj + 1;
    System.out.println("broj = " + broj);
}
```

- Ukoliko sekvenca koja se nalazi u selekciji ili iteraciji sadrži više od jedne instrukcije, navodi se kao blok koda (unutar vitičastih zagrada)

Petlja `do-while`

- Izvršava se sve dok je zadovoljen uslov (ima vrednost `true`)

```
do  
    sekvenca;  
while (<logickiUslov>;
```

- Izvršiće se bar jedna iteracija, bez obzira na vrednost uslova

Primer iteracije pomoću `do-while` petlje

```
int prviBroj = 4;
int drugiBroj = 9;
do {
    prviBroj = prviBroj + 1;
    drugiBroj = drugiBroj - 1;
} while (prviBroj > 8);
```

- Izvršiće se jedna iteracija jer uslov nije zadovoljen, ali se proverava na kraju iteracije

Petlja for

- Sadrži:
 - instrukciju kojom se inicijalizuje petlja
 - uslov koji određuje da li će se nastaviti izvršenje
 - instrukciju koja se izvršava nakon svake izvršene iteracije

```
for (<preIni>; <logickiUslov>; <postIt>)  
    sekvenca;
```

Primer iteracije pomoću `for` petlje

```
int broj = 3;
for (int i = 0; i < 5; i++) {
    broj = broj + 3;
    System.out.println("iteracija " + (i+1));
    System.out.println("broj = " + broj);
}
```

Komande `continue` i `break`

- Ukoliko se želi prekinuti tekuća iteracija i nastaviti izvršenje petlje, koristi se komanda `continue`
- Ukoliko se želi prekinuti tekuća iteracija i izaći iz petlje, koristi se komanda `break`
- Po pravilu, ove komande se navode u nekom `if` izrazu

```
for (int i = 0; i < x; i++) {  
    // deo petlje koji se uvek izvrsava  
    if (<izraz>  
        continue;  
    // deo petlje koji treba da se izvrši samo ako je izraz false  
}
```

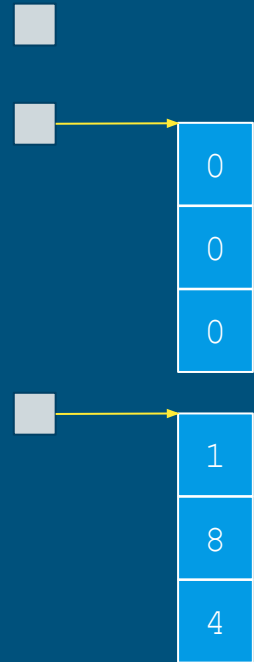
Nizovi primitiva

- Kolekcija promenljivih istog tipa
- Svaki element niza sadrži jednu vrednost
- Dužina niza je fiksna i određuje se prilikom kreiranja

[0]	1
[1]	8
[2]	4

Primer niza

- Deklaracija
`int[] niz;`
- Kreiranje niza
`niz = new int[3];`
- Inicijalizacija elemenata
`niz[0] = 1;`
`niz[1] = 8;`
`niz[2] = 4;`



Pristup elementima niza

- Dodela vrednosti elementu

```
niz[0] = 5;
```

- Čitanje vrednosti elementa

```
int treciElement = niz[2];
```

- Prolazak kroz niz

```
int ukupanZbir = 0;
for (int i = 0; i < niz.length; i++) {
    int itiElement = niz[i];
    ukupanZbir = ukupanZbir + itiElement;
}
```

Rezime

- Tri osnovne strukture u svakom programu su
 - sekvenca
 - selekcija i
 - iteracija
- Java omogućava različite načine implementacije iteracije
- Tipična primena petlje je prolazak kroz niz