

Uvod u programiranje - vežbe

Kontrola toka izvršavanja programa

Naredbe za kontrolu toka

- `if, if-else, switch`
 - uslovni operator (`?:`)
- `for, while, do-while`
- `break, continue, return`



if

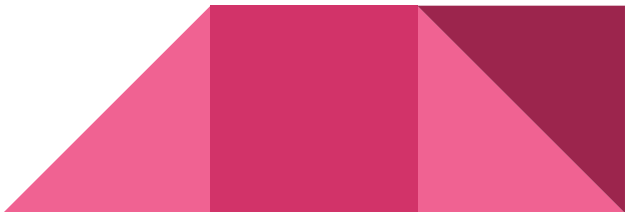
```
if (uslov)
    naredba;
```

```
if (uslov) {
    naredba;
    ...
}
```

```
if (uslov)
    naredba1;
else
    naredba2;
```

```
if (uslov) {
    naredba1;
    ...
} else {
    naredba2;
    ...
}
```

```
if (uslov1)
    naredba1;
else if (uslov2)
    naredba2;
else if (uslov3)
    naredba3;
...
else
    naredban;
```



if

```
int a=3;
int b=5;
if (a>b) {
    System.out.println("a je veće od b");
} else if (a<b) {
    System.out.println("a je manje od b");
} else {
    System.out.println("a je jednako b");
}
```



Ugnježden if

```
int a=18;
if (a>0) { // pozitivan
    if (a%2==0) { // paran
        // a je pozitivan paran broj
    } else { // neparan
        // a je pozitivan neparan broj
    }
} else if (a<0) { // negativan
    if (a%2==0) { // paran
        // a je negativan paran broj
    } else { // neparan
        // a je negativan neparan broj
    }
} else { // nula
    // a je nula
}
```

```
// varijanta bez ugnježdenog if
int a=18;
if (a>0 && a%2==0) {
    // a je pozitivan paran broj
} else if (a>0 && a%2!=0) {
    // a je pozitivan neparan
} else if (a<0 && a%2==0) {
    // a je negativan paran broj
} else if (a<0 && a%2!=0) {
    // a je negativan neparan broj
} else {
    // a je nula
}
```



Uslovni operator (? :)

- Sintaksa

```
uslov ? izraz1 : izraz2
```


- Primer

```
int starost=22;  
System.out.println("osoba je " + (starost<18?"maloletna":"punoletna"));
```

switch

```
switch (izraz) {  
    case vrednost1: naredba1;  
        break;  
    case vrednost2: naredba2;  
        break;  
    ...  
    default: naredban;  
}
```

Koristi se za:

- primitivne tipove (`byte`, `short`, `char`, `int`) i njihove wrapper klase
 - enumerisane tipove
 - `String` (od Java SE 7)
- 

for

- Sintaksa

```
for (inicijalizacija; uslov; korak) {  
    ...  
}
```

- Primer

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



Različiti oblici deklarisanja naredbe for

- izostavljena inicijalizacija

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

- izostavljen uslov

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

- izostavljen inkrement odn. korak

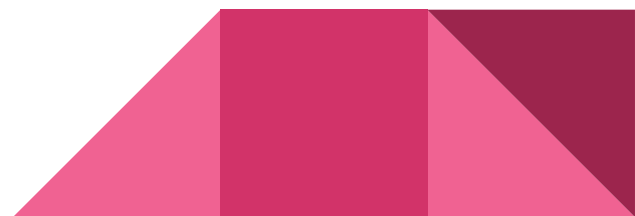
```
for (int i=0; i<10;) {
    System.out.println("i=" + i++);
}
```

- primer sa dve varijable

```
for (int i=0,j=10; i<10 && j>0; i++,j--) {
    System.out.println("i="+i + ", j="+j);
}
```

- primer beskonačne petlje

```
for (;;) {
    ...
}
```



while, do - while

- Sintaksa

```
while (uslov) {  
    ...  
}
```

- Primer


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

- Sintaksa

```
do {  
    ...  
} while (uslov);
```

- Primer

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



break

- napušta tekući for, while, do-while **odn.** switch blok

```
for (int i=0; i<10; i++) {  
    if (i==5)  
        break;  
}
```

- kod ugnježenih struktura, prekida onu koja je poslednja započeta
- korišćenjem labela može se precizirati određen iterativni blok

```
for (int i=0; i<5; i++) {  
    for (int j=0; j<5; j++) {  
        if (j==2)  
            break;  
    }  
}
```

```
spoljnaPetlja:  
for (int i=0; i<5; i++) {  
    for (int j=0; j<5; j++) {  
        if (i>j)  
            break spoljnaPetlja;  
    }  
}
```

continue

- prekida tekuću iteraciju i počinje sledeću u for, while, do-while bloku

```
for (int i=0; i<10; i++) {  
    if (i==5)  
        continue;  
}
```

- kod ugnježenih struktura, odnosi se na onu koja je poslednja započeta
- korišćenjem labela može se precizirati određen iterativni blok

```
for (int i=0;i<5;i++) {  
    for (int j=0;j<5;j++) {  
        if (j==2)  
            continue;  
    }  
}
```

```
spoljnaPetlja:  
for (int i=0;i<5;i++) {  
    for (int j=0;j<5;j++) {  
        if (i==j)  
            continue spoljnaPetlja;  
    }  
}
```

return

- prekida izvršavanje tekuće metode

```
int broj = 3;
if (broj % 2 == 0) {
    System.out.println("broj je paran");
    return;
}
System.out.println("broj je neparan");
```

- u metodama koje imaju deklarisan tip povratne vrednosti (nisu `void`), navodi se i povratna vrednost

```
return vrednost;
```

