

Selekcija switch

# Naredba višestrukog izbora switch

```
switch (selektor) {  
    case konstanta_1:  
        naredba1_1;  
        naredba1_2;  
        break;  
    case konstanta_2:  
        naredba2_1;  
        naredba2_2;  
        break;  
    .....  
    case konstanta_n:  
        naredban_1;  
        naredban_2;  
        break;  
  
    default:  
        naredbaX_1;  
        naredbaX_2;  
        break;  
}
```

# Primer upotrebe naredbe switch-1

```
public static void main(String[] args) {  
    Random rnd = new Random();  
    int x = rnd.nextInt(4)+1; // 1 do 4  
    switch (x) {  
        case 1:  
            System.out.println("*");  
            break;  
        case 2:  
            System.out.println("**");  
            break;  
        case 3:  
            System.out.println("***");  
        default:  
            System.out.println("Generisan je broj 4");  
            break;  
    }  
}
```

# Primer upotrebe naredbe switch-2

```
public static void main(String[] args) {
    Random rnd = new Random();
    int x = rnd.nextInt(10); // 0 do 9

    switch (x) {
        case 0:
        case 2:
        case 4:
        case 6:
        case 8:
            System.out.println("Generisan je paran broj");
            break;
        case 1:
        case 3:
        case 5:
        case 7:
        case 9:
            System.out.println("Generisan je neparan broj");
            break;
    }
    System.out.println("x = " + x);
}
```

# Rešenje bez upotrebe switch naredbe

```
public static void main(String[] args) {  
    Random rnd = new Random();  
    int x = rnd.nextInt(10); // 0 do 9  
  
    if (x %2 == 0) {  
        System.out.println("Generisan je paran broj");  
    }  
    else {  
        System.out.println("Generisan je neparan broj");  
    }  
    System.out.println("x = " + x);  
}
```

# Programske petlje

# Programske petlje

- Programske petlje služe za izvršavanje neke naredbe ili grupe naredbi više puta
- Jedno izvršavanje grupe naredbi naziva se jedan ciklus ili jedna iteracija petlje

# While petlja

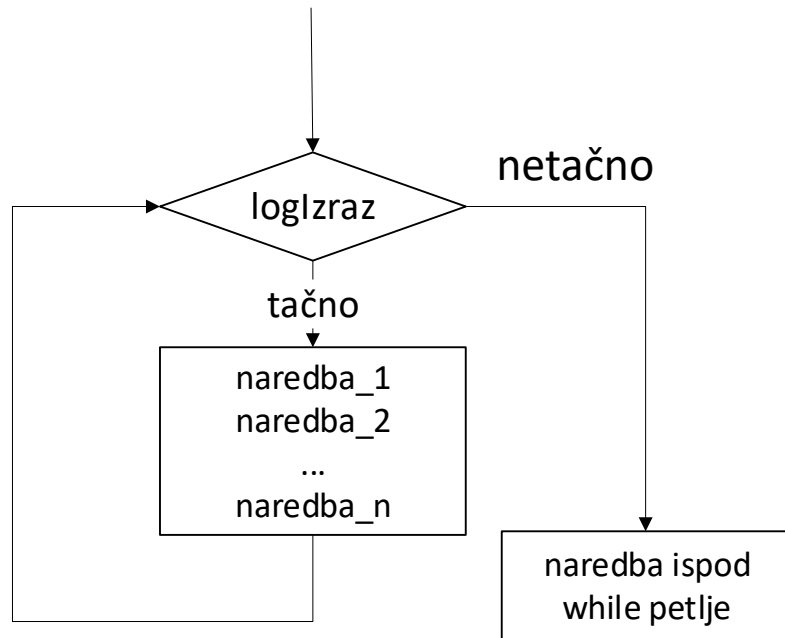
Opšti oblik while petlje je:

```
while(logIzraz) {  
    naredba_1;  
    ....  
    naredba_n;  
}
```

- While petlja je petlja sa izlaskom na vrhu tj. prvo se proverava istinitost logičkog izraza pa se tek izvršava telo petlje
- Ako je vrednost logIzraza tačna izvršiće se naredba ili blok naredbi koji čine telo ciklusa
- Zatim se ponovo izračunava tačnost logIzraza i ako je on tačan ponovo se izvršavaju naredbe koje čine telo ciklusa
- Unutar tela petlje mora postojati naredba koja menja istinitost logičkog izraza



# While petlja – algoritamski prikaz



```
while (logIzraz) {  
    naredba_1;  
    ....  
    naredba_n;  
}
```

Petlja sa izlaskom na vrhu.

# Brojanje unapred

```
public static void main(String[] args) {  
    int brojac = 0;  
    while (brojac < 10) {  
        System.out.println(brojac);  
        brojac++;  
    }  
}
```

# Brojanje unazad

```
public static void main(String[] args) {  
    int brojac = 9;  
  
    while (brojac > -1) {  
        System.out.println(brojac);  
        brojac--;  
    }  
}
```

# Primer beskonačne while petlje

```
public static void main(String[] args) {  
    while (true) {  
        System.out.println("Java");  
    }  
}
```

# Naredba break

```
public static void main(String[] args) {  
  
    int i = 0;  
    while (true) {  
        System.out.println(i);  
        i++;  
  
        if (i > 9) {  
            break;  
        }  
    }  
  
}
```

Naredba break služi za izlazak iz petlje.

# Naredba continue

```
public static void main(String[] args) {  
    int i = 0;  
  
    while (i<100) {  
        i++;  
        if (i%2 ==0) {  
            continue;  
        }  
        System.out.println(i);  
    }  
}
```

Naredba continue završava tekuću iteraciju i prelazi na sledeću

# Funkcije printf() i format()

- Funkcija printf() u Javi omogućava formatiranje i ispisivanje teksta na standardni izlaz
- Može se koristiti i System.out.format()

```
System.out.printf("format", argumenti);
```

```
double pi = 3.141592653589793;  
System.out.printf("Pi sa 3 decimalna mesta: %.3f", pi);
```

- %d - celobrojna vrednost
- %f - decimalna vrednost
- %s - string
- %c - karakter
- %b - boolean

# Primena funkcije format

```
public static void main(String[] args) {  
    int i = 0;  
    while (i < 10) {  
        int kvadrat = i * i;  
        System.out.format("Broj: %d, Kvadrat: %d%n", i, kvadrat);  
        i++;  
    }  
}
```

%n znak se koristi za prelazak u sledeći red



# Primer upotrebe while petlje

Za sve dužine u cm počevši od 0cm do 1m sa korakom promene od 5 cm, nađi odgovarajuće dužine u inch. Zna se da je 1inch = 2.54 cm.

```
public static void main(String[] args) {
    int a = 0;
    float b = 0;
    System.out.println("a(cm)\t|\tb(inch)");
    System.out.println("_____");

    while (a <= 100)
    {
        b = a / 2.54f;
        System.out.format(Locale.ENGLISH, "%d\t|\t%.2f%n", a,b);
        a += 5;
    }
}
```

# Primer upotrebe while petlje

Tabelirati funkciju  $y = x^2 + 2x + 1$  za  $x$  od 0 do 10 sa korakom promene 0.1

```
public static void main(String[] args) {  
  
    System.out.println("x\t|\ty");  
    System.out.println(".....");  
  
    double x=0,y =0;  
    while(x<=10){  
        y = x * x + 2 * x + 1;  
        System.out.printf(Locale.ENGLISH, "%.2f\t|\t%.2f\n", x, y);  
        x+=0.1;  
    }  
}
```

# Primer

Tabelirati funkciju  $y = 1+x+x^2$  u opsegu od  $x_{\min}$  do  $x_{\max}$  sa korakom  $dx$ . Brojevi  $x_{\min}$ ,  $x_{\max}$  i  $dx$  su pozitivni realni brojevi. Ne dozvoliti da minimalna vrednost bude veća od maksimalne. Ne dozvoliti preveliki korak promene tj.  $x_{\min} + dx > x_{\max}$ . Štampu odraditi primenom while petlje.

# Rešenje

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    sc.useLocale(Locale.ENGLISH);

    // Unosimo vrednosti xmin, xmax i dx
    System.out.print("Unesite xmin: ");
    double xmin = sc.nextDouble();
    System.out.print("Unesite xmax: ");
    double xmax = sc.nextDouble();

    double dx = 0;
    while (dx <= 0 || xmin + dx > xmax) {
        System.out.print("Unesite dx: ");
        dx = sc.nextDouble();

        if (xmin + dx > xmax) {
            System.out.println("Korak promene dx je preveliki.");
        }
    }

    // Inicijalizujemo vrednost x na xmin

    double x = xmin;

    // Tabeliramo funkciju u opsegu od xmin do xmax sa korakom dx
    double y;
    // x<=xmax
    while (x < xmax + dx/2) {
        y = 1 + x + Math.pow(x, 2);
        System.out.printf(Locale.ENGLISH, "x = %.2f, y = %.2f\n", x, y);
        x += dx;
    }

    sc.close();
}
```