

# Java

Pogojni stavek, tip boolean, primerjanje

DIRI 2003 – Programski jeziki

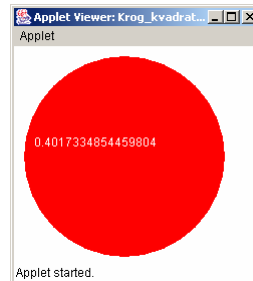
Z verjetnostjo 0.3 nariši kvadrat, z  
verjetnostjo 0.7 pa krog

- Vsi akcije doslej izvajale zaporedoma, zagotovo vse
- Odločitev
  - Če to, potem naredimo eno
  - Če drugače, naredimo drugo
- Pogojni stavek

## Zgled

```
import java.awt.*;
import java.applet.*;

public class Krog_kvadrat extends Applet {
    public void paint(Graphics g) {
        double verj = Math.random();
        g.setColor(Color.red);
        if (verj < 0.3)
        {
            g.fillRect(10,10,200,200);
        }
        else
        {
            g.fillOval(10,10,200,200);
        }
        g.setColor(Color.white);
        g.drawString("" + verj, 20, 100);
    }
}
```

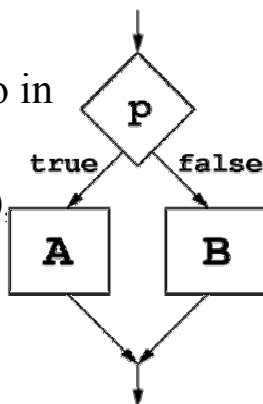


## Kaj se zgodi?

- Če je res, da je vrednost v spremenljivki `verj` manjša kot 0.3, se izvede stavek znotraj `{ }` (`g.fillRect...`).
- Potem nadaljujemo na stavku `g.setColor...`
- Če pa ni res, da je vrednost v spremenljivki `verj` manjša kot 0.3, se izvede stavek znotraj `{ }` za rezervirano besedo `else` (`g.fillOval...`).
- Potem nadaljujemo na stavku `g.setColor...`

## Shematsko

- Vejitev
- Preverimo pogoj  $p$
- Če je resničen (true), gremo in izvedemo akcijo A
- Če pogoj ni resničen (false), izvedemo akcijo B
- Nadaljujemo za vejitvijo.



## Pogojni stavek

- Dve obliki

```
• if (pogoj) {  
    stavek1a;  
    ...  
    stavekna;  
}  
else {  
    stavek1b;  
    ...  
    stavekmb;  
}
```

- Kaj pomeni: če je pogoj  $pogoj$  izpolnjen, se izvedejo  $stavek_{1a}, \dots, stavek_{na}$  sicer pa  $stavek_{1b}, \dots, stavek_{mb}$

## Pogojni stavek

- Druga oblika
- Sintaksa

```
• if (pogoj)
  { stavek1;
    stavek2;
    ...
    stavekn;
  }
```

- Pazi na oklepaje okoli pogoja.
- Stavki se izvedejo, če je pogoj izpolnjen.
- To obliko uporabimo, če takrat, ko pogoj ni izpolnjen, nimamo opraviti ničesar.

Matija Lokar,  
Fakulteta za matematiko in fiziko

DIRI2003  
Stavki v { } se torej izvedejo le, če je pogoj

## Pogoji

### □ Primerjanje (relacijski operatorji)

- >
- <
- >= (brez presledka!)
- <=
- == (pozor dva (2) enačaja)
- !=

Matija Lokar,  
Fakulteta za matematiko in fiziko

DIRI 2003

## Logične vrednosti

- Tip `boolean` (! mala črka !)

- Vrednosti le `true` in `false`.

```
boolean vRedu, konec;
```

```
vRedu = true;
```

```
konec = false;
```

- Kaj bo v `p` in `q`?

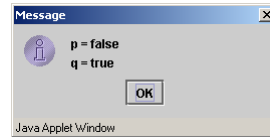
```
int a = 17;
```

```
int b = 5;
```

```
boolean p = (a < b);
```

```
boolean q = (a <= b + 12);
```

[PrimerBoolean.java](#)



## Operacije

- Logične vrednosti lahko združujemo z operatorji

- `&&` in

- `||` ali

- `!` ne

- `A && B`: res, če sta res in A in B (sta oba true)

- `A || B`: res, če je vsaj eden res oziroma narobe le, če sta oba false

- `!A`: res (true), če je A napačen (false)

## Povzetek log. operacij

- Vrednosti
  - `true` resnica
  - `false` neresnica
- Operacije (`p` in `q` sta logični vrednost oz. Izraza, ki vrmeta logično vrednost)
  - `p && q` in `q`
  - `p || q` ali `q`
  - `!p` negacija `p`
- Logične vrednosti najpogosteje dobimo kot rezultat primerjav
  - `A == B` A je enak B
  - `A != B` A ni enak B
  - `A < B` A je manjši od B
  - `A <= B` A je manjši ali enak B
  - `A > B` A je večji od B
  - `A >= B` A je večji ali enak B

## Delitelj

- Napiši program, ki prebere celi števili `a` in `b` ter ugotovi, ali je število `a` delitelj števila `b`
- Če `a` deli `b` izpiši `true`, če pa `a` ne deli `b` izpiši `false`.
- Število je delitelj drugega, če je ostanek pri deljenju enak 0
- `deli = b % a == 0;`
- Izvajanje (npr. `b` je 16, `a` = 3)
  - Izračunamo `b % a` → dobimo 1
  - `1 == 0` → rezultat je `false`
  - V spremenljivko `deli` shranimo `false`

## Delitelj - program

```
public static void main(String[] b) {
    int a, b;
    boolean deli;
    String beri;

    beri = JOptionPane.showInputDialog
        ("Vnesi celo število, katerega deljivost te zanima: ");
    b = Integer.parseInt(beri);
    beri = JOptionPane.showInputDialog
        ("Vnesi morebitnega delitelja: ");
    a = Integer.parseInt(beri);

    deli = b % a == 0;

    JOptionPane.showMessageDialog(null, "Trditev, da " + a +
        " deli " + b + " je " + deli);
}
```

Delitelj.java

## Prestopno leto

- Napiši program, ki prebere leto (celo število) in pove, ali je prestopno.
- Leto je prestopno, če je deljivo s 4.
- Izjema so leta deljiva s 100, ki niso prestopna.
- Dvojna izjema so leta deljiva s 400, ki so prestopna.
  
- Leto 1980 je prestopno, ker je deljivo s 4 in ni deljivo s 100.
- Leto 1700 ni prestopno, ker je deljivo s 100.
- Leto 2000 je prestopno, ker je deljivo s 400.

## Prestopno leto

- Pogoj, da je leto prestopno, je torej, da je
  - deljivo s 4 IN NI deljivo s 100 ALI
  - Je deljivo s 400
- Deljivo s 4
  - `leto % 4 == 0 // POZOR NA == !!!!`
- NI deljivo s 100:
  - `leto % 100 != 0 // ali (!(leto % 100 == 0))`
- Deljivo s 400
  - `leto % 400 == 0`
- Deljivo s 4 IN NI deljivo s 100
  - `((leto % 4 == 0) && (leto % 100 != 0))`
- Prestopno:
  - `((leto % 4 == 0) && (leto % 100 != 0)) || (leto % 400 == 0)`

## Prestopno leto - program

```
import javax.swing.*;

public class Prestopno {
    public static void main(String[] aa)
    {
        int leto;
        boolean prestopno;
        String beri, odgovor;

        beri = JOptionPane.showInputDialog("Leto: ");
        leto = Integer.parseInt(beri);

        prestopno = ((leto % 4 == 0) &&
                    (leto % 100 != 0)) || (leto % 400 == 0);
    }
}
```



## Prestopno leto - program

```
    odgovor = "Leto " + leto;
    if (prestopno)
    {
        odgovor = odgovor + " je ";
    }
    else
    {
        odgovor = odgovor + " ni ";
    }
    odgovor = odgovor + "prestopno leto!";
    JOptionPane.showMessageDialog(null, odgovor,
        "Ali je leto prestopno?",
        JOptionPane.INFORMATION_MESSAGE);
}
}
```

[Prestopno.java](#)

## Prestopno leto II

- Če pobrskamo po dokumentaciji, ugotovimo, da razred `GregorianCalendar` že vsebuje metodo `isLeapYear`, ki ugotovi, ali je dano leto prestopno.
- `import java.util.GregorianCalendar;`
- `GregorianCalendar koledar = new GregorianCalendar();`
- `prestopno = koledar.isLeapYear(leto);`
- Naredimo objekt tipa `GregorianCalendar` in pokličemo metodo `isLeapYear`, ki pove, ali je leto prestopno.
- Edino važno: Javina knjižnica vsebuje vse mogoče uporabne metode
- [Prestopno2.java](#)

## Ali je datum veljaven?

- Preberimo dan, mesec, leto in ugotovimo, če predstavljajo veljavni datum
- Veljaven mesec – če bo med 1 in 12
  - `veljaven_mesec = (1 <= mesec) && (mesec <= 12);`
- Veljaven datum: 1, 3, 5, 7, 8, 10, 12 ima med 1 in 31 dni, ostali (razen 2) pa 30. Pri 2 moramo upoštevati še prestopnost

```
boolean veljaven_dan =
    ((mesec == 1) && (1 <= dan) && (dan <= 31)) ||
    ((mesec == 3) && (1 <= dan) && (dan <= 31)) ||
    ((mesec == 5) && (1 <= dan) && (dan <= 31)) ||
    ((mesec == 7) && (1 <= dan) && (dan <= 31)) ||
    ((mesec == 8) && (1 <= dan) && (dan <= 31)) ||
    ((mesec == 10) && (1 <= dan) && (dan <= 31)) ||
    ((mesec == 12) && (1 <= dan) && (dan <= 31)) ||
    ((mesec == 4) && (1 <= dan) && (dan <= 30)) ||
    ((mesec == 6) && (1 <= dan) && (dan <= 30)) ||
    ((mesec == 9) && (1 <= dan) && (dan <= 30)) ||
    ((mesec == 11) && (1 <= dan) && (dan <= 30)) ||
    ((mesec == 2) && !prestopno_leto && (1 <= dan) && (dan <= 28)) ||
    ((mesec == 2) && prestopno_leto && (1 <= dan) && (dan <= 29))
```

## Limone

- Na trgu sodelavec prodaja limone (zbira denar za sindikalno zabavo)
- Vsakih nekaj minut se oglasi s klicem  
*KUPITE! KUPITE! ŠE 3 LIMONE!*  
oziroma  
*KUPITE! KUPITE! ŠE 75 LIMON!*  
glede na to, koliko limon ima.
- Seveda je po nekaj dneh že čisto hripav!
- Zato mu boste pripravili sintetizator govora, ki bo vpil namesto njega
- Kot prvi korak napišimo program, ki bo prebral število limon in izpisal stavek *KUPITE! KUPITE! ŠE x LIMON!*
- Seveda v pravilni slovenščini!
  - 1 limona, 2 limoni, 3 limone, 4 limone, 5 limon, 6 limon, 7 limon, ...

# Limone

- Ločimo primere
  - 1 (limona)
  - 2 (limoni)
  - 3 ali 4 (limone)
  - 5 ali več (limon)
- 4 pogojni stavki
- [Limone1.java](#)
- Kaj pa 103, 28304, 201, ...
- Zgodba se ponovi na vsakih 100!
  - Dejansko za pravilno obliko upoštevamo ostanek pri deljenju s 100
  - % 100
- [Limone2.java](#)

# Največja številka

- Preberemo naravno število, manjše kot 1.000.000.
- Ugotovi največjo številko.
- $4367 \rightarrow 7, 445124 \rightarrow 5, 16621 \rightarrow 6, 781 \rightarrow 8 \dots$
- Ideja
  - Zaporedoma jemljemo številke
    - enice (%)
    - /
  - Če je trenutna številka večja od doslej največje, si jo zapomnimo!
  - Ko smo pregledali vseh 5, poznamo največjo!

## Za eno

---

```
stevka = stevilo % 10;
if (stevka > naj_stevka)
{ naj_stevka = stevka;
}
stevilo = stevilo / 10;
```

[Naj\\_stevka.java](#)

## Uporaba { }

---

```
g.drawString("Najprej nekaj izpišimo!");
if (2 == 3)
    g.drawString("To se ne izpiše!");
    g.drawString("Tudi to se ne izpiše!");
g.drawString("To se vedno izpiše!");
```

- [If z oklepaji.java](#)
- V telesu pogojnega stavka je dovoljen le en stavek!
- Če jih potrebujemo več – *sestavljene stavke*

## Sestavljeni stavek

- Poljubno zaporedje stavkov med { }  
{  
    stavek<sub>1</sub>;  
    stavek<sub>2</sub>;  
    ...  
    stavek<sub>n</sub>;  
}
- Na koncu ni ;
- Lažje – pri pogojnem stavku VEDNO uporabimo { }
- Najpogosteje takoj naredimo oba oklepaja – ni težav s pozabljanjem
- Zamikanje – eden pod drugim

## Ali je datum pred drugim?

- Datum podan z letom, mesecem in dnevom
- Primerjamo leti
  - Če sta enaki, primerjamo meseca
    - Če sta enaka, primerjamo dneva
      - Zvemo odgovor
    - Če nista, poznamo odgovor
  - Če nista, poznamo odgovor
- Torej datum1 je pred datumom 2, če velja
- (let1 < leto2) ALI
- (let1 == leto2 IN mesec1 < mesec2) ALI
- leto1 == leto2 IN mesec1 == mesec2 IN dan1 < dan2

## "Gnezdeni" pogojni stavki

- Znotraj pogojnega stavka je lahko poljuben stavek – tudi pogojni stavek!

```
if (g == 10)
{
    g = 0;
    dag = dag + 1;
    if (dag == 100)
    {
        dag = 0;
        kg = dag + 1;
    }
}
```

## Primerjaj števili po velikosti

- Izpiši v kakšnem "odnosu" sta si števili

- Manjše
- Večje
- Enako

- $st1 < st2$

- DA: vemo odgovor
- NE: dve možnosti!

```
if (st1 < st2)
{
    odg = "manjše";
}
else
{
    if (st1 == st2)
    {
        odg = "enako";
    }
    else
    {
        odg = "večje";
    }
}
```