Java

Zanke

Oglejmo si naslednji program

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

public class Tarca extends Applet
{
   int n; // stevilo krogov
   int r_max; // radij najvecjega kroga

   public void init()
   {
      // preberemo stevilo krogov in naj radij

      String branje;
      branje = JOptionPane.showInputDialog("Stevilo krogov ");
      n = Integer.parseInt(branje);
      branje = JOptionPane.showInputDialog("Radij najvecjega kroga ");
      r_max = Integer.parseInt(branje);
   }
```

Matija Lokar,
Fakulteta za matematiko in fiziko

Oglejmo si naslednji program

```
public void paint(Graphics g)
  {
     int debelina = r_max / n;
     int i = 1; // kateri krog risemo
     int radij = r_max; // radij tekocega kroga
     int x_odmik = 10;
     int y_odmik = 20;  // odmik najvecjega kroga

     while (i <= n)
     {
        g.drawOval(x_odmik, y_odmik, 2 * radij, 2 * radij);
        radij = radij - debelina;
        x_odmik = x_odmik + debelina;
        y_odmik = y_odmik + debelina;
        i = i + 1;
     }
   }
}
```

Tarca.java

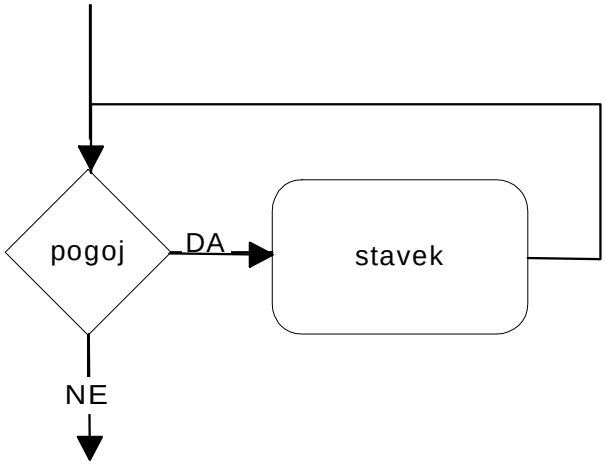
Kaj dela

- Poženemo
- Zakaj dela tako?
- Stavek while

ZANKE

- Seštej 10 števil, izpiši 20 zvezdic, nariši n krogov
- Ponavljanje
  - isti postopek, spremenjeni podatki
- osnovna zanka
  - while

while

- Sintaksa:
  - ```
    while (pogoj)
    {
       stavek1;
       ...
       Stavek_n;
    }
    ```
- Izvajanje
  - Preveri pogoj. Če je resničen, izvedi stavke v zavitih oklepajih.
  - Preveri pogoj. Če je resničen, ponovno izvedi stavke.
  - Preveri pogoj …
- Dokler je logični pogoj izpolnjen, izvajaj stavke v zavitih oklepajih.
    ```

While – shematski prikaz



While – zgled

- ```
  while (x>1)
  {
    x = x / 2;
  }
  ```
- Kaj se zgodi, če je v x na začetku vrednost 4.2
- ustavitev
- zaciklanje
  ```

While – zgled

```
while (x <= 10)
  {
    x = x + 1;
  }
```

- Kaj se dogaja, če je v x na začetku 1?
- Kaj se dogaja, če je x na začetku 100?

While – zgled

```
while (x <= 10)
  {
    x = x - 1;
  }
```

- Kaj se dogaja, če je x na začetku 100?
- Kaj se dogaja, če je v x na začetku 1?

Izpiši števila od 1 do 20

- Izpis
  - JOptionPane.showMessageDialog
- Kako do vsebine izpisa
  - tipkanje
  - odg = "1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20"
- Od1do20_tip.java
- Kaj pa, če bi želeli izpisati števila med 1 in 1000?

Izpiši števila od 1 do 20

- Števec od 1 do 20
- Zanka
  - Pogoj: dokler je števec manjši ali enak 20
  - `while (stevec <= 20)`
- Na vsakem koraku
  - števec dodamo k izpisu
    - `izpis = izpis + " " + stevec;`
  - Povečamo števec
    - `stevec = stevec + 1;`

Izpiši števila od 1 do 20

```
import javax.swing.*;

public class Od1do20 {
   public static void main(String[] args)
   {
     String izpis = "";
     int stevec = 1;

     while (stevec <= 20)
     {
        izpis = izpis + " " + stevec;
        stevec = stevec + 1;
     }
     JOptionPane.showMessageDialog(null, izpis);
   }
}
```

Od1do20.java

Števec ponovitev

- `stevec = 1;`
- `while (stevec <= stevilo_ponovitev)`
- `{`

```
        .... // naredimo nekaj
        stevec = stevec + 1;
   }
```

- Števec je zaporedoma: `1, 2, 3, ..., stevilo_ponovitev, stevilo_ponovitev + 1`

Izpis števil od a do b

- Izpiši števila od a do b (a in b podatka, ki ju preberemo)
  - Ne moremo narediti z direktnim izpisom
  - obvezna zanka
- Števec od a do b
- Začetna vrednost števca: a
  - `stevec = a;`
- Zanka
  - Pogoj: dokler je števec manjši ali enak b
  - `while (stevec <= b)`
- Na vsakem koraku
  - števec dodamo k izpisu
    - `izpis = izpis + " " + stevec;`
  - Povečamo števec
    - `stevec = stevec + 1;`

Izpis števil od a do b - program

```
String izpis = "";
int od_kje,
    do_kam; // "do" ne bi bilo dobro ime - rezervirana beseda!
int stevec;
String beri;

beri = JOptionPane.showInputDialog("Od kje naprej = ");
od_kje = Integer.parseInt(beri);
beri = JOptionPane.showInputDialog("Do kam = ");
do_kam = Integer.parseInt(beri);

stevec = od_kje;
while (stevec <= do_kam) {
   izpis = izpis + " " + stevec;
   stevec = stevec + 1;
}
JOptionPane.showMessageDialog(null, izpis);
```

OdAdoB.java

Izpiši soda števila od a do b

- Izpiši soda števila od a do b.
  - Različica, ko računalnik "bolj trpi"
  - Različica, ko zadevo malo bolj premislimo in računalniku ne povzročamo nepotrebnega dela

"Preveč" obremenjeni računalnik

- Osnovni program za izpis števil od A do B
- V zanki preverimo, če je število sodo!

```
while (stevec <= do_kam)
{
   if (stevec % 2 == 0)
     izpis = izpis + " " + stevec; // izjemoma
                                   // oblika brez {}
   stevec = stevec + 1;
}
```

- SodaOdAdoB.java

Brez nepotrebnega dela

- Izpisati je potrebno a, a + 2, a + 4, ...
  - a le, če je a sodo število
  - sicer a + 1, a + 3, a + 5, ...
- Začetni števec
  - a (če je a sod)
  - a + 1 (če je a liho število)

```
if (a % 2 == 0)
     stevec = a;
else
    stevec = a + 1;
```

- Povečevanje števca za 2
- SodaOdAdoB_2.java

Ogrlica

- Narišimo ogrlico
- Niz raznobarvnih kroglic – vodoravno
  - Lepša bi bila na elipsi, ampak potem bi se pritoževali, da je preveč matematike!
- Barvo kroglic bomo določali naključno
- Spreminja se levi zgornji kot
  - `x = x + 2 * radij;`
- Preberemo število kroglic
  - `JOptionPane.showInputDialog, Integer.parseInt`
- Preberemo radij kroglic
- Kolikor je `st_kroglic`, tolikokrat ponovimo
  - določi naključno barvo
  - nariši kroglico
  - določi novo koordinato levega zgornjega kota

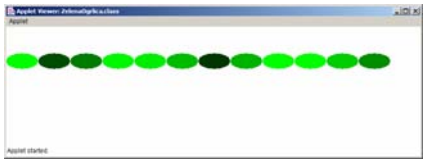
Ogrlica

```
x_koor = 0;
y_koor = 0;
stevec = 1;
while (stevec <= st_kroglic)
{
  rdeca = (int)(Math.random() * 256);
  zelena = (int)(Math.random() * 256);
  modra = (int)(Math.random() * 256);
  g.setColor(new Color(rdeca, zelena, modra));
  g.fillOval(x_koor, y_koor, 2 * r, 2 * r);
  x_koor = x_koor + 2 * r;
  stevec = stevec + 1;
}
```

Ogrlica.java

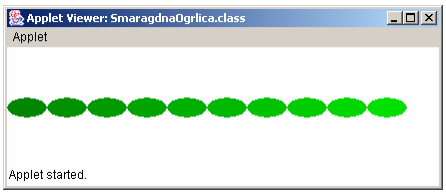
Smaragdna ogrlica

- Denimo, da bi radi narisali ogrlico iz ovalnih smaragdov
- Oval: Višina "kroga" naj bo r
- Smaragd: zelena barva
  - rdečo in modro komponento barve pustimo enako 0
- ZelenaOgrlica.java



Barvno usklajena ogrlica

- Barva naj se preliva od temno zelene do svetlo zelene.

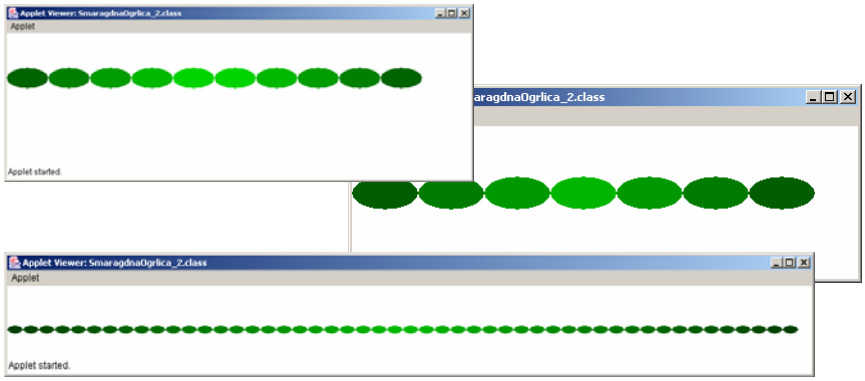


Barvno usklajena ogrlica

- Naključno določimo temno zeleno, nato svetlo zeleno
  - "vidna" zelena se začne nekako pri 50
  - `temna_zelena = (int)(Math.random() * 103) + 50;`
  - `svetla_zelena = (int)(Math.random() * 103) + 154;`
- Izračunamo vmesne odtenke
  - "Razmik" odtenkov
    - `razmik_odtenkov = (svetla_zelena - temna_zelena) / (st_kroglic - 1);`
  - V zanki
    - `zelena = temna_zelena + (stevec - 1) * razmik_odtenkov;`
    - `g.setColor(new Color(0, zelena, 0));`
- SmaragdnaOgrlica.java

Barvno usklajena ogrlica

- Kaj pa od temno zelene preko svetlo zelene in nazaj do temno zelene?



Barvno usklajena ogrlica

- pol manj odtenkov
  - `razmik_odtenkov = (svetla_zelena - temna_zelena) / (st_kroglic / 2);`
- do polovice zanke kot običajno
  - `zelena = temna_zelena + (stevec - 1) * razmik_odtenkov;`
  - `Množimo z 0, 1, 2, ..., (st_kroglic/2 - 1)`
- v drugi polovici barvo moramo jemati odtenke "nazaj"
  - `zelena = temna_zelena + ?? * razmik_odtenkov;`
  - Množiti moramo z `(st_kroglic/2 - 1), (st_kroglic/2 - 2), ...`
  - Števec je takrat `st_kroglic / 2 + 1, st_kroglic / 2 + 2, ...`
  - `???` je torej `st_kroglic - stevec`
- SmaragdnaOgrlica_2.java

Kaj počne tale del

```
int i = 100;
String odgovor = "";
while (i > 100)
{
    odgovor = odgovor + "riba raca rak ";
    i = i + 1;
}
JOptionPane.showMessageDialog(null, odgovor);
```

Zanke1.java

Kaj pa tale

```
String odgovor = "";
while (true)
{
  odgovor = odgovor + "klop pod klopjo, ";
  JOptionPane.showMessageDialog(null, odgovor);
}
JOptionPane.showMessageDialog(null, "Končali smo...");
```

Zanke2.java

In ta?

```
String odgovor = "";
i = 1;
while (i <= 10)
   odgovor = odgovor + "klop pod klopjo, ";
   JOptionPane.showMessageDialog(null, odgovor);
   i = i + 1;
JOptionPane.showMessageDialog(null, "Končali smo ...");
```

Zanke4.java

In ta?

```
String odgovor = "";
 int i = 1;
 while (i <= 10);
{
  odgovor = odgovor + "klop pod klopjo, ";
  JOptionPane.showMessageDialog(null, odgovor);
  i = i + 1;
 }
JOptionPane.showMessageDialog(null, "Končali smo ...");
```

Zanke3.java

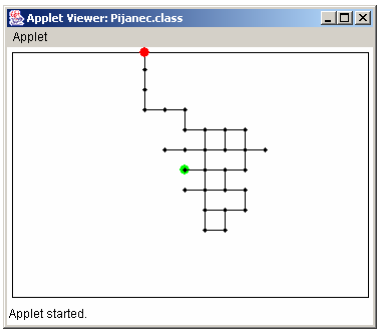
Morebitne težave

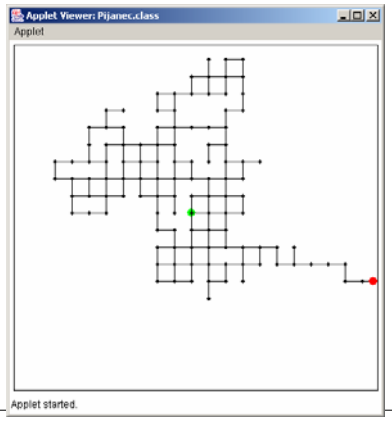
- Pogoj napačen
  - Zanka se nikoli ne konča!
- Pozabljeni { }
  - Kot pri pogojnem stavku
  - V telesu le en stavek
  - Ker običajno želimo narediti več stvari: sestavljeni stavek
  - Zamikanje ne pomaga (prevajalniku je vseeno)
- Napačno ; takoj za pogojem
  - Zanka, ki ima v telesu "prazen" stavek

Zgledi

Pijanec

- Narišimo gibanje pijanca





Pijanec

- Tavamo, dokler se ne zvrnemo v jarek
- Tavamo
  - Na vsakem koraku se premaknemo v eni od 4 smeri neba (naključna izbira) za 20 enot
- Jarek
  - Zaidemo izven zaslona

Pijanec

- Logična spremenljivka ni_v_jarku
- Na začetku `true`
- Zanka, dokler ni_v_jarku
  - `while (ni_v_jarku)`
- V zanki
  - Nov položaj
    - Naključno 0, 1, 2, 3
    - Ustrezno naračunamo nove koordinate
  - Ugotovimo, ali smo v jarku
  - Če nov položaj neugoden (izven zaslona)
    - Popravimo na rob
  - Narišemo premik
- `Pijanec.java`

Pijanec - izboljšave

- po vsakem koraku "počijemo"
  - `System.currentTimeMillis()`
    - trenutni čas v milisekundah
  - zapomnimo si trenutni čas
  - toliko časa gledamo na uro, dokler ne bo trenutni čas večji za npr. 500 ms
  - `long trenutniCas = System.currentTimeMillis();`
  - `while (System.currentTimeMillis() < trenutniCas + 500);`
  - Zanka ne dela nič!
    - čaka, da bo minilo 500 milisekund
  - `Pijanec_pocasi.java`
- Barvno kodiranje, kje smo
  - oranžno: trenutna točka
  - prejšnja točka mora postati ponovno črna!
  - `Pijanec_pocasi_1.java`

Zanka while

- Včasih vemo vnaprej, koliko bo ponovitev
  - števec
    - števec postavimo na začetno vrednost PRED zanko
    - Pogoj zanke
      - (stevec <= končna_vrednost)
    - V zanki števec povečujemo
      - stevec = stevec + povečanje
      - (ali zmanjšujemo ≡ povečujemo negativno – takrat seveda pogoj drugačen >=)
  - Pogosto uporabimo drug tip zanke (for)
    - `for (i = 1; i <= 10; i++) { A; }`
    - `i = 1; while (i <= 10) { A; i = i + 1;}`

Zanka while

- Včasih števila ponovitev ne vemo vnaprej
  - Pijanec
  - Ponovitve odvisne od dogajanja v zanki
  - Ponavljaj, dokler se ne zgodi nekaj (npr. streljaj v tarčo, dokler ne zadaneš, ...)
  - Bolj zapleten pogoj
  - "Prava" while zanka

Kako sestavljamo program z zanko

- Premislimo, kaj se dogaja v splošnem
  - Tekoča ponovitev zanke
    - Rišemo i-ti krog
    - Pregledujemo tekočo vrstico
    - ...
- Dogajanje na začetku (pred vstopom v zanko)
  - Posebni pogoji ...
  - Nastavitev števcev
  - vrednost kontrolne spremenljivke taka, da se zanka sploh začne, ...
- Dogajanje na koncu
  - Ali je potrebno z zadnjim elementom kaj posebnega narediti
  - Smo števec po "nepotrebnem" preveč povečali
  - ...

Povprečje števil

- Beri cela števila in izračunaj povprečje
- števec prebranih števil
- vsota števil
- zanka: branje, povečanje števca, prištej k vsoti
- ko prebereš 0, zaključi zanko

Splošni korak

- Preberemo število
  - `str = JOptionPane.showInputDialog("podatek:");`
  - `pod = Double.parseDouble(str);`
- Prištejemo število k skupni vsoti
  - `vsota = vsota + pod;`
- Povečamo število prebranih
  - `prebranih = prebranih + 1;`

Zanka

- Pogoj
  - Konec: prebrano število je 0
  - Torej je pogoj za ponavljanje, da prebrano število NI enako 0
  - `(pod != 0)`
- Pred zanko
  - Vsota je 0, prav tako število prebranih
  - Kako "vstopiti" v zanko?
    - Prvi podatek prebrati posebej
    - Prvi podatek preberemo v zanki, kot ostale
      - Spremenljivko pod nastaviti na neko poljubno vrednost, le različno od 0!

Po zanki

- Števec prebranih je prevelik
  - Šteli smo tudi 0, ki ne spada med podatke
  - `prebranih = prebranih - 1;`
  - "Povečava" vsote ni problematična
- Morda:
  - Kaj, če podatkov ni
    - Že prvo prebrano 0
    - Ne moremo računati povprečja
- Izračun povprečja

Povprečje - program

```
int prebranih = 0; // stevilo prebranih podatkov;
double vsota = 0.0; // vsota prebranih podatkov
double pod = 10.0; // trenutni podatek (karkoli pred zanko, le != 0
double povprecje;  // povprečje prebranih

while (pod != 0)
{
   pod = Double.parseDouble(
           JOptionPane.showInputDialog("Vnesi podatek / 0 = konec"));
   vsota = vsota + pod;
   prebranih = prebranih + 1;
}
prebranih = prebranih - 1; // 0 ne štejemo zraven
if (prebranih == 0)
{
   JOptionPane.showMessageDialog(null, "Ni podatkov - ni povprečja!");
}
else
{
   povprecje = vsota / prebranih;
   JOptionPane.showMessageDialog(null, "Povprečje je " + povprecje);
}
```

Povprecje.java

Obrnjeno število

- Celo število izpiši obrnjeno
  - 123456 --> 654321
- `enice = stevilo % 10` ---> izpišemo
- `stevilo = stevilo / 10`
- ponavljamo, dokler ni `stevilo == 0`

IzpisiObrnjenoStevilo.java

123456 6543

Naredi obrnjeno število

- osnova - prejšnji program
- obrnjeno stevilo množimo z 10 in prištevamo enice

ObrniStevilo.java

Zanka v zanki

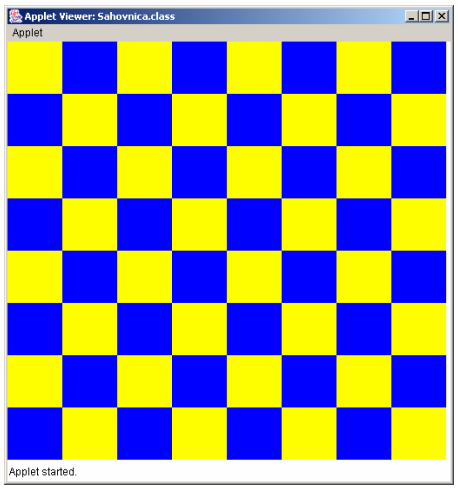
Zanka v zanki

- 10 števil izpiši obrnjeno

```
/* 10 x izpisi stevilo obrnjeno */
stpon = 1;
while (stpon <= 10)
{
  /* izpisi stev obrnjeno */
  stev = Integer.parseInt(JOptionPane.showInputDialog(null,"Stevilo"));
  ob_stevilo = 0;
  while (stev != 0) // dokler ne zmanjka stevk
  {
    enice = stev % 10;
    ob_stevilo = ob_stevilo * 10 + enice; // lepimo zraven k rezultatu
    stev = stev / 10; /* odrezemo zadnjo stevko */
  }
  JOptionPane.showMessageDialog(null, ob_stevilo);
  stpon = stpon + 1; /* nova ponovitev */
}
```

DesetObrnjenih.java

Šahovnica



Šahovnica

- Narisati moramo 8 vrstic
- V vsaki vrstici imamo 8 stolpcev (kvadratkov)
- Kako določiti barvo?
  - Če seštejemo vrstico in stolpec
  - Vse sode vsote so ene barve, vse lihe pa druge

```
if ((vrstica + stolpec) % 2 == 0)
{
    g.setColor(svetlaBarva);
}
else
{
    g.setColor(temnaBarva);
}
```

Šahovnica - program

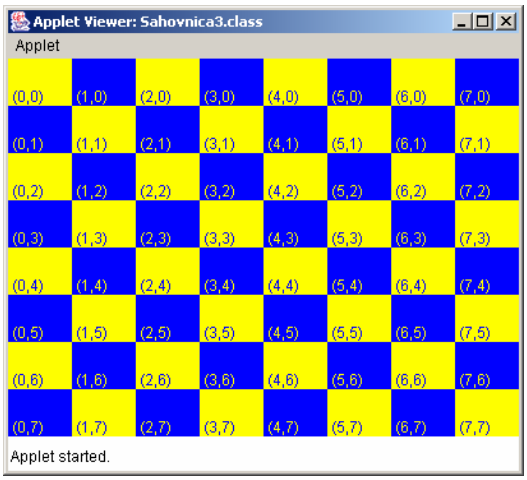
```
import java.applet.*;
import java.awt.*;

public class Sahovnica extends Applet {

   public void paint(Graphics g) {
    int N = 8;

    int stolpec;
    int sirina = getWidth() - 10; // sirina prostora programcka - 10
    int visina = getHeight() - 10; // visina prostora programcka - 10

    Color svetlaBarva = new Color(255, 255, 0);
    Color temnaBarva = new Color(0, 0, 255);

    // a in b sta sirina in visina kvadratka
    int a = sirina / N;
    int b = visina / N;
```


Šahovnica - program

```
      int vrstica = 0;

      while (vrstica < N) {

          stolpec = 0;

          while (stolpec < N) {
           if ((vrstica + stolpec) % 2 == 0) {
               g.setColor(svetlaBarva);
           }
           else {
               g.setColor(temnaBarva);
           }

           g.fillRect(stolpec * a, vrstica * b, a, b);

           stolpec = stolpec + 1;
          }
          vrstica = vrstica + 1;
      }
   }
  }
```

Sahovnica.java

Šahovnica s koordinatami



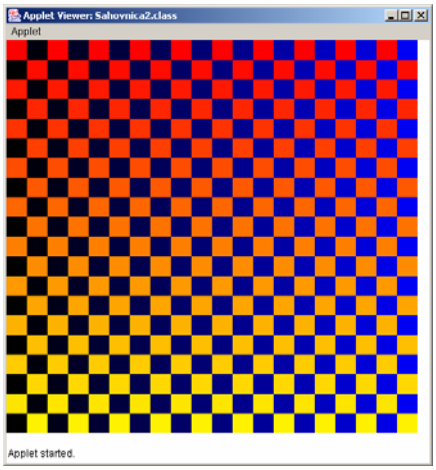
Dodamo le

```
g.drawString("(" + stolpec + "," + vrstica +
   ")",
              stolpec * a + 3, vrstica * b +
   b - 3);
```

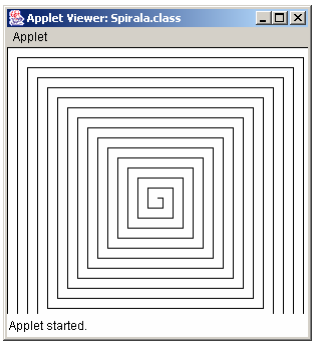
- Sahovnica3.java

Prelivajoče se šahovnica

Sahovnica2.java



Spirala



Spirala.java