

Metode

Funkcije, podprogrami

DIRI 2003 – Programski jeziki

Zakaj metode?

- Doslej programi kratki, enostavni
 - Tudi naprej bodo taki ;-)
- "Pravi" programi so večji
 - Tudi nekaj milijonov vrstic
- Ne moremo napisati v enem kosu
- Razdelitev v manjše kose
- Vsak kos posebej
- Java – metode
- Drugi prog. jeziki: funkcije, podprogrami, procedure, subrutine, ...

Pripravljene metode

- Doslej smo uporabljali že cel kup metod
 - Risanje pravokotnika, določitev naključnega števila,
 - Izris okna z obvestilom
 - Izračun sinusa, izpis na zaslon
 - ...
- Knjižnice
- Metode lahko pišemo tudi sami
- Pravzaprav jih že ves čas
 - `main`, `paint`, `init`
 - Lahko pa jih poimenujemo po svoje
- `main`, `paint`, `init`
 - Metode s točno predpisanim namenom / obliko / ...

Zakaj metode

- "skrivanje" postopka - metodo napišemo in jo potem lahko uporabimo, ne da bi skrbeli, kako je napisana
- preglednost programa

Strukturirano programiranje

- program - zaporedje "neodvisnih" postopkov (branje podatkov, obdelava podatkov (lahko spet iz več neodvisnih postopkov), predstavitev rezultatov)
- Nariši histogram
 - Branje podatkov
 - Normalizacija (koliko velika naj bo enota)
 - Risanje

Izdelava mize

- Nariši načrt
 - Iz načrta izdelaj spisek potrebnega materiala
 - Nabavi material
 - Ugotovi najugodnejšo ceno lesa
 - Ugotovi najugodnejšo ceno barve
 - Opravi nakup lesa
 - Opravi nakup barve
 - Sestavi mizo
 - Nažagaj sestavne dele
 - Združi sestavne dele
 - Pobarvaj mizo
- ko kupujem, me ne zanima, kako bom npr. barval in ne, kako sem ugotovil, da potrebujem 3 l rdeče barve
- ko žagam, me ne zanima kako sem nabavil les

Kako deliti

- Pomembna odločitev
- Navadimo se predvsem s prakso
- Načela
 - Preglednost
 - Uporabnost v drugih programih
 - "samozadostnost"
 - Ne izpisujemo ali beremo, če to ni izrecni namen metode

Vrste metod

- Statične
- Objektne
- Najprej bomo obravnavali le statične

- Vsaka metoda vrača
 - rezultat
- in sprejme
 - argumente

Primer

```
public static int max(int a, int b)
{ /* vrne vecje od dveh celih stevil */
  if (a > b)
    { return a;}
  else
    {return b;}
}
```

```
x = max(12,x);
...
System.out.println("Vecje med stevili " + a + " in " +
  c + " je " + max(a,c));
...
y = max(max(a,b) + max(c,d), 2 + max(100,x * x));
```

Kaj pomeni

- `public`
 - Vsi deli programa lahko uporabljajo to metodo
 - Do nadaljnega vse metode "public"
- `static`
 - Statična metoda
 - Do nadaljnega vse metode "static"
- `int` (za `public static`)
 - Rezultat bo tipa `int`
- `max`
 - Ime metode (poljubno!)
- `(int a, int b)`
 - Argumenta metode
 - Oba sta tipa `int`
 - Formalni argument
- `return`
 - Kakšen rezultat vrne metoda
 - Konec izvajanja metode

Kako je videti program

```
public class MojProgram {  
  
    // tu napisemo metode, eno za drugo  
  
    // ne pozabimo napisati metode main!  
  
}
```

Kako je videti program

```
import javax.swing.*;  
public class MojProgram {  
    public static int max(int a, int b)  
    { /* vrne vecje od dveh celih stevil */  
        if (a > b)  
            { return a;}  
        else  
            {return b;}  
    }  
    public static void main(String[] parametri) MojProgram.java  
    { // določi največje število med prebranimi //  
        String pod;  
        int x, y, z;  
        int naj;  
        pod = JOptionPane.showInputDialog("Prvo število: ");  
        x = Integer.parseInt(pod);  
        pod = JOptionPane.showInputDialog("Drugo število: ");  
        y = Integer.parseInt(pod);  
        pod = JOptionPane.showInputDialog("Tretje število: ");  
        z = Integer.parseInt(pod);  
        JOptionPane.showMessageDialog(null, "Največje število med " +  
            x + ", " + y + " in " + z + " je " + max(x, max(y, z)));  
    }  
}
```

Kako je videti program

```
public class MojProgram {
    public static int max(int a, int b)
    { /* vrne vecje od dveh celih stevil */
        if (a > b)
        { return a; }
        else
        { return b; }
    }
    public static void main(String[] parametri)
    { // določi največje število med prebranimi //
        ...
        JOptionPane.showMessageDialog(null, "Največje število med "
            +
            x + ", " + y + " in " + z + " je " + max(x, max(y, z)));
    }
}
```

Definicija metode

Klic metode

Klic metode

Matija Lokar,
Fakulteta za matematiko in fiziko

DIRI 2003

Formalni in dejanski argument

```
public class MojProgram {
    public static int max(int a, int b)
    { /* vrne vecje od dveh celih stevil */
        if (a > b)
        { return a; }
        else
        { return b; }
    }
    public static void main(String[] parametri)
    { // določi največje število med prebranimi //
        ...
        JOptionPane.showMessageDialog(null, "Največje število med " +
            x + ", " + y + " in " + z + " je " + max(x, max(y, z)));
    }
}
```

Formalna argumenta

Dejanska argumenta

Dejanska argumenta

Matija Lokar,
Fakulteta za matematiko in fiziko

DIRI 2003

Formalni in dejanski argumenti

- Formalni argument
 - Za opis delovanja metode
- Dejanski argument
 - Začetna vrednost argumenta metode
- Klic `max(12, y)`
 - Spremenljivka (parameter) `a` v metodi dobi vrednost 12
 - Parameter `b` v metodi dobi vrednost, ki je shranjena v spremenljivki `y`

Klic funkcije

- `ime(dejanski parametri)`
- `ujemanje parametrov`
 - prenaša se vrednost parametra
 - število
 - tip
- `max(12, 23 * 7)`
- `max(a - b, x)`

Odgovori

```
public static int vsota(int n)
{
    int r = 1;
    int i = 1;
    while (i < n)
    {
        i = i + 1;
        r = i * r;
    }
    return r;
}
```

- Kako je ime metodi?
- Kakšen je tip rezultata, ki ga vrača metoda?
- Koliko argumentov sprejme metoda?
- Kakšni so tipi in imena argumentov?
- Kaj metoda počne? Ali lahko predlagaš boljše ime za metodo?
- Napiši program [Demo.java](#), ki demonstrira delovanje metode.

Zgled

- Napiši metodo inicialke, ki sprejme dva niza znakov (ime in priimek) in vrne niz sestavljen iz inicialk. Na primer, inicialke("France", "Preseren") naj vrne niz "F.P."
- Ime metode
 - inicialke
- Tip metode
 - String
- Število argumentov
 - 2
- Tip in imena argumentov
 - String ime
 - String priimek

Inicialke

```
public static String
    inicialke(String ime, String priimek)
{
    char zp, zi;
    zi = ime.charAt(0);
    zp = priimek.charAt(0);
    return zi + "." + zp + ".";
}
```

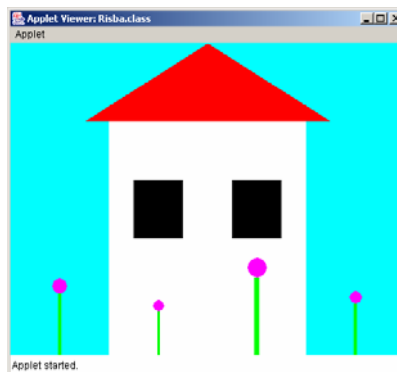
[Demo inicialke.java](#)

Metode in programčki

- Seveda tudi pri pisanju programčkov (appletov) lahko uporabljamo metode
- Velja vse enako
- [Demo inicialke applet.java](#)

Hiša z rožami

- Napisati želimo programček (applet), ki nariše hišo z rožicami:



Hiša z rožami

- Metoda za risanje hiše
- Metoda za risanje rož
- Kakšne rezultate vrmeta metodi
 - Nobenega
 - Če metoda ne vrača rezultata, potem pravimo da vrne rezultat praznega tipa, ki se mu v Javi reče void.
- `void main, void paint, ...`
- `public static void hisica(???)`

Argumenti metod, ki rišejo

- Doslej smo risali v metodi `paint`
- Ta ima parameter `Graphics g`
- Ta parameter moramo posredovati vsem metodam, kjer želimo risati!
- Metode za risanje rožic in hišice naj sprejmejo argument `Graphics g`.
- Verjetno je smiselno napisati metodo, ki nariše eno rožico. Ta metoda sprejme za argumente položaj rožice in njeno velikost.

Glave metod

- `public static void rozica
 (Graphics g, int x, int y, int velikost)`
- `public static void hisica
 (Graphics g, int x, int y, int sirina,
 int visina)`

paint

```
public void paint(Graphics g)
{
    int sirina = getWidth();
    int visina = getHeight();

    setBackground(new Color(0, 255, 255));
    hisica(g, sirina/4, visina, sirina/2, (3*visina)/4);
    rozica(g, sirina/8, visina, visina/5);
    rozica(g, (3*sirina)/8, visina, visina/7);
    rozica(g, (5*sirina)/8, visina, visina/4);
    rozica(g, (7*sirina)/8, visina, visina/6);
}
```

hisica

```
public static void hisica(Graphics g, int x, int y, int sirina, int visina)
{
    // hisa
    g.setColor(new Color(255, 255, 255));
    g.fillRect(x, y - visina, sirina, visina);

    // okna
    g.setColor(Color.black);
    g.fillRect(x + sirina/8, y - (3*visina)/4, sirina/4, visina/4);
    g.fillRect(x + (5*sirina)/8, y - (3*visina)/4, sirina/4, visina/4);

    // streha
    g.setColor(Color.red);
    int[] a = {x - sirina/8, x + sirina + sirina/8, x + sirina/2};
    int[] b = {y - visina, y - visina, y - visina - visina/3};
    g.fillPolygon(a, b, 3);
}
```

rozica

```
public static void rozica(Graphics g, int x,
                          int y, int velikost)
{ // pecelj
  g.setColor(new Color(0, 255, 0));
  g.fillRect(x - velikost/30 ,
            y - velikost, velikost/15, velikost);

  // cvet
  g.setColor(new Color(255, 0, 255));
  g.fillOval(x - velikost/8,
            y - velikost - velikost/4,
            velikost/4, velikost/4);
}
```

Risba.java

Kaj so argumenti in kaj tipi metod

- `public static String prvaBeseda(String besedilo)`
- `public static double[] produkt(double[] A, double[] B)`
- `public static void foo()`
- `public static int najdi(String geslo, String[] slovar)`

Glave metod

- Za vsakega od naslednjih opisov zapišimo deklaracijo metode (kakšen tip metoda vrača in kakšne argumente sprejme) Metod ni treba napisati!.
 - `public static String najdaljsi(String[] a)`
- Metoda, ki v podani tabeli nizov poišče najdaljši niz in ga vrne.
 - `public static double kvadratna(double a, double b, double c)`
- Metoda, ki najde eno izmed obeh rešitev dane kvadratne enačbe (pod predpostavko, da so rešitve realna števila).
 - `public static String[] uredi(String[] a)`

Glave metod

- Metoda, ki na ekran izpiše "Kdor to bere, je osel!".
 - `public static void osel()`
- Metoda, ki sprejme dve barvi (objekta razreda Color) in vrne njuno mešanico.
 - `public static Color zmesaj(Color a, Color b)`
- Metoda, ki v dani grafični objekt (razred Graphics) nariše daljico z danimi oglišči in dane barve.
 - `public static void daljica(Graphics g, int x1, int y1, int x2, int y2, Color barva)`

Kaj naredi ta program

```
import javax.swing.*;

public class KajProgram {
    public static int povecaj(int a)
    { // povecam a za 1
        a = a + 1;
        return a;
    }
    public static void main(String[] parametri)
    { String pod;
      int a;
      int naj;
      pod = JOptionPane.showInputDialog("Število: ");
      a = Integer.parseInt(pod);
      naj = povecaj(a);
      JOptionPane.showMessageDialog(null, "naj = " +
        naj + "\na = " + a);
    }
}
```

[KajProgram.java](#)

Parametri se prenašajo po vrednosti

- Ob klicu parameter metode dobi vrednost, kot je določena v dejanskem parametru
- Potem ni nobene povezave med dejanskim in formalnim parametrom

Tabela

- ❑ Sestavi tabelo naključnih celih števil med 1 in 100.
- ❑ Prepiši jih v novo tabelo tako, da bodo v novi tabeli najprej elementi, ki imajo v prvi tabeli indekse 0, 2, 4,, potem pa še elementi z lihimi indeksi.
- ❑ Če ime prvotna tabela elemente 2, 4, 23, 5, 45, 6, 8 so v novi tabeli elementi razporejeni kot 2, 23, 45, 8, 4, 5, 6.
- ❑ Izpiši obe tabeli po 10 v vrsto.

Tabela

- ❑ Razbijemo na metode
 - Generiraj – ustvari tabelo
 - Preloži – preloži na zahtevan način
 - Izpiši – izpiše po 10 v vrsto
- ❑ `public static int[] generiraj(int velikostTab, int sm, int zm)`
- ❑ `public static int[] prelozi(int[] tabela)`
- ❑ `public static void izpisi(int[] tabela)`

Tabela – glavni program

```
public static void main(String[] arg)
{ // generiraj naključno tabelo
  // prelozi v novo, . tako da so elementi s sodimi
  // indeksi pred lihimi
  int veltab = Integer.parseInt(arg[0]);
    // parameter programa je velikost tabele
  int[] tabOrig;
  int[] tabNova;
  int sM = 1; // spodnja meja števil v tabeli
  int zM = 100; // zgornja meja števil v tabeli

  tabOrig = generiraj(veltab, sM, zM);
  tabNova = prelozi(tabOrig);
  izpisi(tabOrig);
  izpisi(tabNova);
}
```

[SodiIndeksiPredLihe.java](#)

Uporaba metod iz drugih programov

- Poenostavljena zgodba
- Zares
 - Kasneje, ko bo govora o objektih, razredih, ..
- Datoteka: NekiProgram.java
- V njej metoda: MetodaA
 - `public static int metodaA(int x)`
- Datoteka: DrugiProgram.java
 - Klic metode: `NekiProgram.metodaA(izrazTipaInt)`
- Metode drugih programov (razredov) torej kličemo kot
 - `ImeRazreda.imeMetode(parametri)`
- Velja za metode z določili
 - `static` in `public`

Prejšnji zgled v dveh razredih

- En razred naj vsebuje metode
 - `public static int[] generiraj(int velikostTab, int sm, int zm)`
 - `public static void izpisi(int[] tabela)`
- Drugi
 - `public static int[] prelozi(int[] tabela)`
- [GenTab.java](#)
- [SodiIndeksiPredLiheVar2.java](#)