

Java

Priprava programčkov

DIRI 2003 – Programski jeziki

Samostojno pisanje

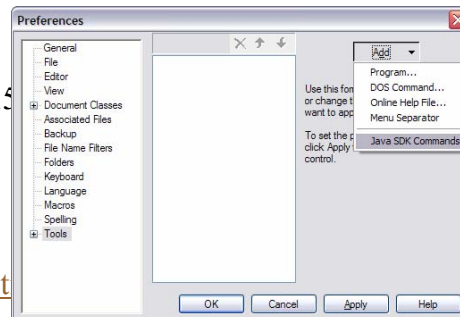
- Programčke lahko napišemo tudi sami
- Potrebujemo
 - Prevajalnik za Javo
 - [JavaSDK 1.4](#) (brezplačno razvojno okolje) (SUN)
 - JBuilder (Borland), IBM Visual Age for Java, ...
 - Znanje programskega jezika Java

Od problema do programa

- Problem
- Ideja
- Priprava datoteke s kodo
 - urejevalnik
- Prevajalnik
 - Podatki – datoteka s kodo
 - Rezultat – prevedeni program (class datoteka(e))
- Vključitev programčka v spletno stran
 - Priprava ustrezne HTM datoteke

Priprava

- JavaSDK
 - Trenutno 1.4.2, kmalu 1.5
 - CD
- Namestitev
 - <http://zaversnik.fmf.uni-lj.si/Gradiva/Java/namest>
- TextPad
 - Preizkusni program
 - Že “pozna” Javo
 - Configure/Preferences/Tools
 - Add / Java SDK Commands



Prvi programček

- TextPad
 - Urejevalnik teksta
 - Že pripravljeno okolje za prevajanje
- Lahko uporabimo tudi Beležnico ali poljubni drugi urejevalnik ... (kot za pisanje datotek HTML)
 - Glej CD
- Zelo pomembno poimenovanje datotek, velike/male črke, ...
- Prvi.java
- Prevedemo v Prvi.class
- Napišemo ustrezno datoteko HTML

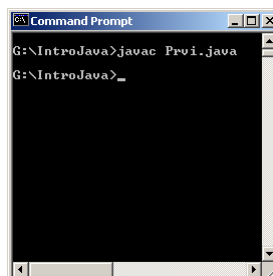
Prvi programček

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

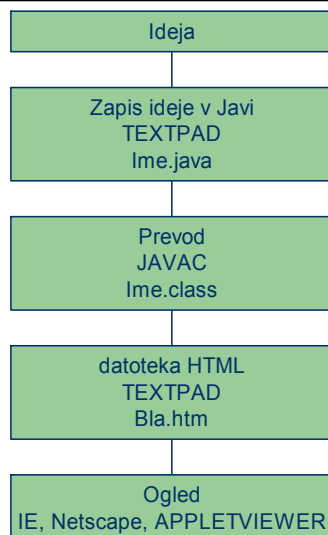
public class Prvi extends Applet
{
    public void paint(Graphics okno)
    {
        // Prvi program v Javi
        okno.drawString("Pozdravljen na koordinatah 15, 20",
            15, 20);
    }
}
```

Prvi programček

- Shranimo v Prvi.java (enako ime kot je napisano za public class (Prvi) in dodamo .java)
- Ukazna vrstica
 - javac Prvi.java
 - Nastane Prvi.class
- Neposredno v TextPadu
 - Tools
- HTML datoteka
- Ogled HTML datoteke
 - Appletviewer (prikaže le programček)
 - Brskalnik



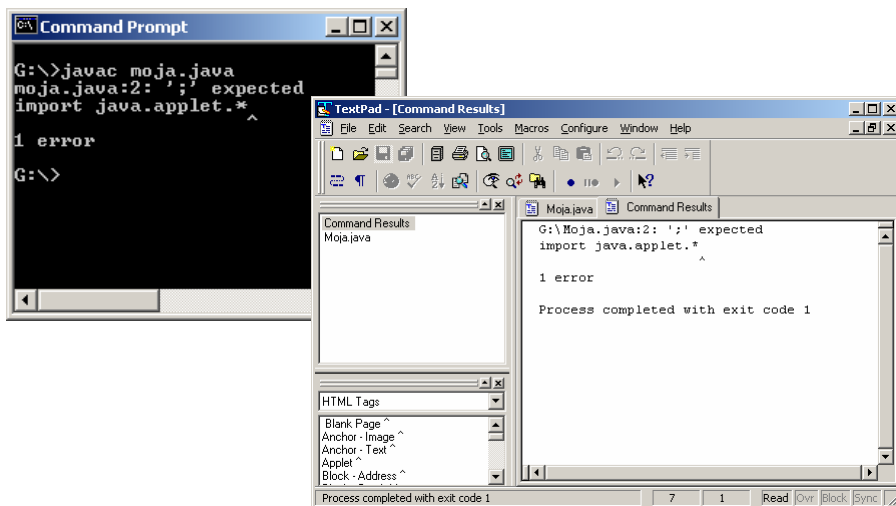
Razvoj programčka



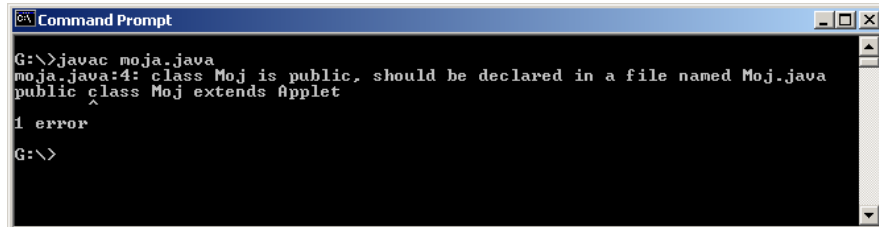
Pogoste napake

- Male/velike črke
 - V Javi bla, Bla, BLA, ... označujejo različne stvari
- Ime datoteke ni enako imenu razreda (Class)
 - Če napišemo `public class Moj extends ...`, moramo to shraniti v datoteko `Moj.java`
- Pozabljeni `{`, `;`, ...
 - Sintaksa jezika Java (in drugih pr. jezikov) je taka, da se jo je potrebno točno držati

Pozabljeno podpičje



Napačno ime datoteke



```
Command Prompt
G:\>javac moja.java
moja.java:4: class Moj is public, should be declared in a file named Moj.java
public class Moj extends Applet
      ^
1 error
G:\>
```

"Okostje" programčka

- Vsi naši prvi programčki bodo videti približno takole

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

public class Ime_programčka extends Applet
{
    public void paint (Graphics g)
    {
        ...
    }
}
```

Okostje.java

- S tem povemo, da naj se na grafični površini `g` prikaže tisto, kar napišemo namesto `...`. To se zgodi vsakič, ko se mora osvežiti stran, ki prikazuje programček z imenom `Ime_programčka`.

Izvajanje metod

- Imamo grafični objekt `g`
- Določen z `Graphics g`
- Nad objektom lahko izvedemo metodo
 - `ime_objekta.ime_metode(parametri)`
 - Metode so postopki, ki jih objekt "pozna"
- Da obstaja grafični objekt in ustrezne metode "poskrbi" prej omenjeno "okostje".
- `g.drawString("xx", 2, 3);`

`drawString(kaj, kje_x, kje_y)`

- `Kaj`: niz, ki ga želimo izpisati
 - "Matija", "Bla", "Moj prvi program"
 - Zaporedje znakov med "
- `kje_x, kje_y`: koordinati levega spodnjega kota napisa
 - (0,0) je levo zgoraj, koordinate merimo v pikslih (osnovnih točkovnih elementih)
 - zaslon: mreža kvadratkov (pikslov), ki so lahko določene barve
 - `x`: stolpci, naraščajo v desno
 - `y`: vrstice, naraščajo navzdol

Naslov

- Napišimo programček, ki napiše naslov
- `g.drawString("Matija Lokar", 100, 100);`
- `g.drawString("Kranj", 100, 120);`
- Podpičja na koncu stavkov!
- Na mesto 100, 100 bomo postavili napis Matija Lokar, na mesto 100, 120 pa Kranj



Naslov – celotni programček

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

public class Naslov extends Applet
{
    public void paint (Graphics g)
    {
        g.drawString("Matija Lokar", 100, 100);
        g.drawString("Kranj", 100, 120);
    }
}
```


Napis s senco

- Našemu napisu bi radi dodali senco

Matija Lokar

- Napišemo napis še enkrat, le premaknjenega za (1,1)

```
g.drawString("Matija Lokar", 100, 100);  
g.drawString("Matija Lokar", 101, 101);
```

Uokvirjen napis

- Napis uokvirimo
- `drawRect(x, y, širina, višina)`
- (x,y) levi zgornji kot pravokotnika
- Ustrezne parametre določimo s poskušanjem
- `g.drawRect(95, 85, 75, 20);`

Uokvirjen napis - program

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

public class UoNap extends Applet
{
    public void paint (Graphics g)
    {
        g.drawString("Matija Lokar",100,100);
        g.drawString("Matija Lokar",101,101);
        g.drawRect(95, 85, 75, 20);
    }
}
```

Seznam nekaj grafičnih metod

- `drawLine(x1, y1, x2, y2)`
- `drawOval(x, y, širina, višina)`
- `drawRoundRect(x, y, šir, viš, šir_lok, viš_lok)`
- `drawArc(x, y, šir, viš, zač_kot, kot)`
- `fillOval(x, y, širina, višina)`
- `fillRect(x, y, šir, viš)`
- `fillArc(x, y, šir, viš, zač_kot, kot)`

□ Kaj počnejo, si oglejte in preizkusite sami!

□ Ne pozabimo:

objekt .metoda

Seznam grafičnih metod

- Kaj delajo posamezne metode, si lahko ogledamo v dokumentaciji, npr.
- <http://haka.fmf.uni-lj.si/java/docs/api/java/awt/Graphics.html>