

# PRESEK

List za mlade matematike, fizike, astronome in računalnikarje

ISSN 0351-6652

Letnik 31 (2003/2004)

Številka 5

Strani 283-286

Matjaž Zaveršnik:

## SREČANJE Z JAVO

Ključne besede: računalništvo, programiranje, programski jeziki, java, applet.

Elektronska verzija: <http://www.presek.si/31/1569-Zaversnik.pdf>

© 2004 Društvo matematikov, fizikov in astronomov Slovenije

© 2010 DMFA - založništvo

Vse pravice pridržane. Razmnoževanje ali reproduciranje celote ali posameznih delov brez poprejšnjega dovoljenja založnika ni dovoljeno.

## SREČANJE Z JAVO

V Preseku se pogosto pojavljajo naloge, ki zahtevajo razmeroma veliko računanja in preverjanja, kar od reševalca zahteva dobršno mero natančnosti in potrpežljivosti. Namesto s svinčnikom in papirjem se lahko takih nalog lotimo tudi s pomočjo ustreznega računalniškega programa. Kar precej programov, ki so se pojavljali v Preseku, je bilo napisanih v enem od programskih jezikov pascal, C ali logo. Čas je, da spoznamo še programski jezik java.

### Osnovno o javi

Programski jezik java so razvili pri podjetju Sun. Njihova želja je bila sestaviti enostaven splošno namenski programski jezik, ki bi omogočal pisanje prenosljivih programov, torej programov, ki bi jih lahko izvajali na računalniku s poljubnim operacijskim sistemom ali kakšni drugi napravi. Ali jim je to uspelo ali ne, tukaj ne bi razpravljali. Dejstvo je, da je java kmalu po prvi javni predstavitvi, 23. maja 1995 na konferenci SunWorld, postala eden najbolj priljubljenih programskih jezikov.

Java je objektno usmerjen programski jezik. Na prvi pogled je podoben programskemu jeziku C++, v resnici pa je med njima kar precej razlik. Ker naj bi bila java enostaven programski jezik, so iz jezika C++ v javo prenesli samo najbolj uporabne koncepte (manjkajo npr. večkratno dedovanje, nadgradnja operatorjev, vzorci in kazalci), jih precej bolj strogo definirali (velikosti celoštevilskih tipov so v javi točno določene, medtem ko so bile v C++ veliko bolj ohlapne) ter dodali nekaj novih (samodejno sproščanje pomnilnika, ki ga ne potrebujemo več).

Java pa ni samo oskubljeni C++, pač pa vsebuje obsežno zbirko paketov, ki programerju omogočajo uporabo grafike, pisanje večnitnih programov in pisanje programov za internet. Te možnosti so v standardnem C++ pogrešali številni programerji.

Če si bomo zaželeli programirati v programskem jeziku java, bomo na svojem računalniku potrebovali urejevalnik znakovnih datotek (Notepad ali pa kaj boljšega), s katerim bomo program napisali, ter programsko podporo za javo, ki jo dobimo na Sunovih spletnih straneh. Obstaja v različnih izvedbah (Enterprise, Standard, Micro), različicah (zadnja je 1.4.2), za različne operacijske sisteme (Windows, Linux) in različne uporabe (JRE vsebuje samo podporo za izvajanje programov, SDK pa tudi prevajalnik in nekatera druga orodja). Običajno je povsem dovolj standardna izvedba SDK, seveda pa izberemo zadnjo različico za svoj operacijski sistem.

Ko na svoj računalnik prenesemo izbrano datoteko (njena velikost je okoli 44MB), jo samo poženemo in sledimo navodilom, ki jih izpisuje

namestitveni program. Edina stvar, ki jo moramo še sami opraviti (če delamo pod operacijskim sistemom Windows), je nastavitev sistemske spremenljivke `PATH`. Dodati ji moramo opis poti do mape `bin`, ki jo imamo v mapi, kamor smo namestili `java`.

### Preprost program

Program, napisan v programskem jeziku `java`, je sestavljen iz definicij enega ali več razredov. V podrobnosti objektnega programiranja se ne bi spuščali, zato si zaenkrat samo predstavljajmo, da so razredi vrste objektov, ki jih uporabljamo v našem programu. Običajno vsak razred definiramo na svoji datoteki s končnico `java`. Razred vsebuje komponente in metode, ki so lahko statične (nekateri jim pravijo tudi razredne) ali objektne. Podrobnosti ta hip niso pomembne.

Spodnji program je sestavljen iz javnega (`public`) razreda (`class`) z imenom `Pozdrav`, ki vsebuje javno (`public`) statično (`static`) metodo z imenom `main`. Ta dobi za parameter tabelo nizov (`String[]`) z imenom `args` in ne vrne nobenega rezultata (`void`). Metoda `main` vsebuje stavke, ki se izvedejo, ko poženemo program. V našem programu je to stavek, ki na standardni izhod (običajno je to zaslon) izpiše besedilo `Pozdravljeni!`.

```
public class Pozdrav
{
    public static void main(String[] args)
    {
        System.out.println("Pozdravljeni!");
    }
}
```

V `java` je zelo pomembno, da definicijo razreda shranimo na datoteko s končnico `java`, ki ima enako ime, kot ga ima razred. Pri tem so pomembne tudi male in velike črke. Zgornji program moramo zato shraniti na datoteko `Pozdrav.java`.

Program prevedemo iz ukazne vrstice z ukazom `javac`, ki mu damo za parametre imena tistih datotek, ki jih je potrebno prevesti (te datoteke imajo vse končnico `java`). V zgornjem primeru je program sestavljen iz enega samega razreda, torej je potrebno prevesti samo eno datoteko (`Pozdrav.java`). Če prevajalnik v programu ne odkrije nobenih napak, dobimo za vsak razred, ki smo ga definirali, novo datoteko z imenom razreda in končnico `class`. Te datoteke vsebujejo prevedene razrede. Program poženemo z ukazom `java`, ki mu damo za parameter ime tistega razreda, ki vsebuje metodo `main` (v našem primeru je to razred `Pozdrav`).

```
C:\> javac Pozdrav.java
C:\> java Pozdrav
Pozdravljeni!
```

```
C:\>
```

Java torej vsebuje prevajalnik (program `javac`), ki nam prevede napisani program. Pri tem ne dobimo izvršljive datoteke, kot nam jo dajo prevajalniki drugih programskih jezikov (pod operacijskim sistemom Windows običajno dobimo datoteko s končnico `exe`). Tisto, kar dobimo, so datoteke s končnico `class`. To je isti program, zapisan v veliko krajši obliki, ki jo računalnik lahko veliko hitreje obdela. Da bi tak program lahko izvedli, pa potrebujemo še tolmač (program `java`), ki prebira in izvaja prevedene ukaze.

Prevedeni programi v programskem jeziku `java` so torej samo pogojno prenosljivi, saj na računalniku, kjer želimo izvajati take programe, potrebujemo tolmač za `java`. Tolmači pa obstajajo že za skoraj vse operacijske sisteme, tako da s tem ne bi smelo biti težav.

## Applet

Eden od razlogov, zakaj je `java` postala tako popularna, je v tem, da lahko v `javi` pišemo tudi programe, ki jih potem vgradimo na svoje spletne strani. Tak program se s tujko imenuje `applet`. Na spletni strani ga vidimo kot pravokotno grafično področje, znotraj katerega so prikazani rezultati programa (slika, animacija).

Poglejmo si, kako napišemo preprost `applet`. Podobno kot prejšnji program bo tudi `applet` izpisal besedilo `Pozdravljeni!`.

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

public class PozdravApplet extends Applet
{
    public void paint(Graphics g)
    {
        g.drawString("Pozdravljeni!", 50, 30);
    }
}
```

V programu smo najprej napovedali uporabo razredov iz standardnih paketov `java.applet` in `java.awt`. To sta standardna paketa, ki ju dobimo skupaj s prevajalnikom. V prvem paketu so definirani razredi, ki jih potrebujemo pri izdelavi `appleta`, v drugem pa razredi, ki jih potrebujemo pri delu z grafiko. Pri definiciji razreda `PozdravApplet` smo določili, da je ta razred razširitev standardnega razreda `Applet`. Namesto metode `main`

imamo pri appletih metodo `paint`, ki na grafični objekt `g` (ta predstavlja pravokotno grafično področje v brskalniku) nariše sliko. Ta metoda se izvede vsakič, ko mora brskalnik obnoviti sliko. To je seveda takoj na začetku, ko naloži spletno stran, pa tudi kasneje, ko odmaknemo kakšno okno, ki nam je sliko delno ali v celoti prekrivalo.

Ker applet deluje v grafičnem načinu, besedila ne moremo izpisati, pač pa ga moramo narisati. To stori metoda `drawString`, ki dano besedilo nariše na grafični objekt na dano mesto. Točka (50,30) določa, kje bo levi spodnji vogal besedila. Koordinatni sistem je postavljen tako, da je izhodišče v levem zgornjem vogalu, koordinata  $x$  narašča proti desni, koordinata  $y$  pa navzdol.

Če želimo preizkusiti delovanje appleta, ga moramo vgraditi na spletno stran. To pomeni, da moramo sestaviti datoteko v obliki HTML, v kateri bomo applet vgradili z uporabo značke `applet`. Ta ima več lastnosti, najpomembnejše so `width` in `height`, ki določata velikost pravokotnega področja, znotraj katerega bo deloval applet, in `code`, ki določa, katera datoteka vsebuje prevedeno razširitev razreda `Applet` (v našem primeru je to datoteka `PozdravApplet.class`).

```
<applet width="400" height="200" code="PozdravApplet.class">
</applet>
```

Delovanje appleta si potem lahko ogledamo v skoraj vsakem, ne pre-starem spletnem brskalniku ali pa iz ukazne vrstice s pomočjo programa `appletviewer`, ki nam applet požene v samostojnem oknu. Programu damo za parameter ime datoteke HTML.

```
C:\> appletviewer PozdravApplet.html
```

### Koristne povezave

<http://java.sun.com/>

Osnovna spletna stran podjetja Sun, namenjena Javi.

<http://java.sun.com/j2se/>

Spletna stran podjetja Sun, namenjena standardni izvedbi Jave.

<http://java.sun.com/docs/books/jls/>

Knjiga "The Java Language Specification", ki vsebuje podroben opis programskega jezika Java.

<http://java.sun.com/j2se/1.4.2/docs/api/>

Dokumentacija o vseh standardnih paketih in razredih.

<http://erzatz.free.fr/storage/jdk142-hh.zip>

Dokumentacija o vseh standardnih paketih in razredih v obliki `HtmlHelp`.

*Matjaž Zaveršnik*