Objektno orijentisano programiranje 2

Uvod u jezik Java

Motivacija

- Prenosivost programa sa jedne na drugu platformu
  - osnovni motiv za razvoj jezika
- Jezici kakvi su C i C++ nisu platformski potpuno nezavisni
  - standardi obezbeđuju značajnu prenosivost, ali samo na nivou izvornog koda
- Programiranje aparata za domaćinstvo
  - prvobitni povod za razvoj Jave
- Proboj Interneta i njegovih servisa (npr. WWW)
  - dodatni motiv koji se tek kasnije pojavio
- Od posebnog interesa – prenosivost izvršnog koda
- Interesantno: i mikrotalasnim rernama i univerzumu Interneta su potrebni prenosivi programi

Istorijat

- Jezik je koncipirao tim firme *Sun Microsystems, Inc.* 1991. godine
  - J.Gosling, P.Naughton, C.Warth, E.Frank, M.Sheridan
- Razvoj prve radne verzije trajao je 18 meseci
- Prvobitno ime jezika *Oak*, od 1995 ime je promenjeno u *Java*
- Od 1993. godine fokus se premešta sa kućne elektronike na WWW
- Proširenje tima koji je usavršio prototip i uobličio *Javu* 1995:
  - B.Joy, A.v.Hoff, J.Payne, F.Yellin, T.Lindholm
- Trenutno aktuelna verzija: 8u128 (2017.)
  - prethodne godine u ovo vreme: 8u74 (2016.)
- Ranije verzije: 1.0, 1.1, 1.2, 1.3, 1.4.0, 1.4.1, 1.4.2, 1.5 (5), 6, 7
- Razvojno i izvršno okruženje:
  - Oracle Java SE (*Standard Edition*)
- Java SE obuhvata:
  - JRE (*Java Runtime Environment*) i JRE Server
  - JDK (*Java Development Kit*)

Mehanizam

- Osnova za platformsku nezavisnost
  - interpretativan kod
- Prevođenjem izvornog Java programa dobija se interpretativni "bajtkod " (*bytecodes*)
- Interpreter za bajtkod – Java virtuelna mašina (JVM)
- Bajtkod je "mašinski jezik" za JVM
- JVM je "standardizovana"
  - jedan bajtkod će se identično interpretirati na proizvoljnoj JVM
- Implementacija JVM zavisi od konkretne platforme, ali interfejs prema bajtkodu ostaje isti
- Interpretiranje bajtkoda je efikasno, ali ipak slabijih performansi od izvršenja programa na mašinskom jeziku
- Rešenje – JIT (*Just In Time*) prevodioci za bajtkod
  - deo po deo koda se prevodi u toku njegovog interpretiranja

Aplikacije i apleti

- Programi na Javi (na klijentskoj strani):
  - aplikacije
  - apleti
- Aplikacije su klasični „desktop“ programi koji se:
  - distribuiraju na konvencionalan način
  - pokreću pod operativnim sistemom računara (pod kojim je instalirana JVM)
- Apleti su (uglavnom mali) programi koji se:
  - distribuiraju preko Interneta, odnosno WWW, kao delovi HTML stranica
  - pokreću u okviru WWW čitača (*browser* programa)

Koncepti i osobine jezika

- Objektna-orijentacija
  - moderan OO jezik: klase, nasleđivanje, polomorfizam, interfejsi
- Jednostavnost
  - C/C++ sintaksna sličnost, ali jednostavniji OO model
- Prenosivost
  - postiže se interpretacijom bajtkoda
- Sigurnost
  - JVM pruža zaštitu od virusa koji bi se prenosili kroz izvršni kod
- Robusnost
  - stroga provera tipova, proveravani izuzeci, sakupljanje đubreta
- Efikasnost
  - JIT prevodioci

Podrška kroz biblioteku klasa

- Programiranje GUI i obrada događaja
  - paketi AWT, SWING, SWT, skup paketa JavaFX
- Perzistencija
  - seriajlizacija objekata
- Konkurentnost
  - klasa `Thread` – objekti su aktivni – programske niti
- Distribuiranost
  - RMI *(Remote Method Invocation),* servleti, veb-servisi
- Komponentizacija
  - *Java Beans* i *Enterprise Java Beans*

Alati za razvoj

- Oracle (ranije Sun Microsystems):
  - alati iz komandne linije: Java SE (Std. Edition) i druge edicije
  - integrisano okruženje (IDE): Java Studio (ranije ONE Studio, još ranije Forte)
    - Java Studio Enterprise – prelazak na NetBeans IDE
- NetBeans IDE (*Open Source*, sponzorstvo Sun Microsystems, pa Oracle)
- Borland: J-Builder
- BlueJ (La Trobe University, Australia, & University of Kent, UK)
- Xinox Software: JCreator
- Eclipse IDE for Java
- Microsoft:
  - ne podržava više Javu
  - ranije: Visual Studio .NET Visual J#
  - još ranije: Visual Studio 6.0 Visual J++
- IKVM.NET – java za .NET i Mono (VM, Javina biblioteka klasa)

Program na jeziku Java

- Programi na Javi su sastavljeni od klasa (i drugih kategorija tipova)
- Klasa u osnovi sadrži dve vrste članova:
  - polja (*fields*), atributi – podatke koji pripadaju objektima klase ili samoj klasi (statička polja)
  - metode (*methods*) – serije naredbi koje rade nad poljima
- Polja služe za reprezentaciju stanja objekta, odnosno klase
- Metodi definišu ponašanje objekta, odnosno klase
  - dohvataju stanje objekta, odnosno klase (ako su metodi statički) – inspektori
  - menjaju stanje objekta, odnosno klase – mutatori

Primer programa

- Izvorni kod programa “Pozdrav” na jeziku Java (fajl: `p.java`):

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

- `main` – metod koji se izvršava kada se pokrene klasa kao aplikacija
- Prevođenje:
  `javac p.java`
  - kao rezultat se dobija bajtkod `Pozdrav.class`
- Izvršavanje:
  `java Pozdrav`