

Uvod u programski jezik Java

Sadržaj predavanja

- Povijest programskog jezika Java
- Svojstva programskog jezika Java
- Razlike između Jave i C++-a
- Verzije Jave
- Okruženja za razvoj Java aplikacija
- JRE i JDK
- Java API
- Faze kod kreiranja i izvršavanja Java programa

Povijest programskog jezika Java (1)

- Java je nastala iz C++-a
- Sun Microsystems je 1991. pokrenuo istraživački projekt „Green” u sklopu kojeg je nastao novi jezik temeljen na jeziku C++
- James Gosling, autor novog jezika daje mu ime „Hrast” (engl. *Oak*), međutim to ime je već imao neki drugi programski jezik pa je preimenovan u „Java”
- Projekt „Green” propada jer se potrošačka elektronika nije razvijala brzinom koju je Sun predvidio
- U to vrijeme, 1993. godine naglo je počeo razvoj World Wide Weba i ljudi iz Suna su uočili veliki potencijal Jave u tom području
- Java je omogućavala dodavanje dinamičkih interaktivnih sadržaja i animacija web stranicama

Povijest programskog jezika Java (2)

- Sun prvi put predstavlja Javu na konferenciji 1995. godine gdje je privukla veliku pažnju zbog značajnog interesa oko Weba
- Java se danas osim za razvoj desktop aplikacija koristi i za *enterprise* aplikacije, aplikacije za potrošačku elektroniku (100% Blu-ray playera koristi Javu)
- 2009. godine Oracle preuzima Sun Microsystems
- 21.09.2017. objavljuje se inačica Java 9
- Na Javi se temelje drugi programski jezici nove generacije kao što su Clojure, Groovy, Scala, JRuby, Jython, Fantom itd.

Svojstva programskog jezika Java

- Objektno orijentirani jezik
- Neovisan o platformi, isti program se može izvršavati na različitim operacijskim sustavima (Windows, Linux, OS X...)
- Prenosivost između različitih platformi omogućena je **Java virtualnim strojem** (engl. *Java Virtual Machine - JVM*) koji je potrebno instalirati na okolinu na kojoj se razvijaju i izvršavaju Java aplikacije
- Java programi pretvaraju se u format podataka koji se naziva **bytecode** kojeg pokreće (interpretira) Java virtualni stroj
- *Bytecode* nije izvršni kod, već viskooptimirani skup instrukcija dizajniran za izvođenje unutar JVM-a

Razlike između Jave i C++-a

- Java automatski zauzima i oslobađa memoriju pomoću *Garbage Collectiona*, čime oslobađa programera od te obveze
- Java ne uključuje pokazivače
- Java omogućava nasljeđivanje samo jedne klase
- C++ se prevodi u izvršni kod, dok se Java prevodi u *bytecode* koji je potrebno interpretirati

Verzije Java

- Java Standard Edition (Java SE) – za razvoj *desktop* i poslužiteljskih aplikacija
 - Uključuje četiri programske paradigme:
 - Proceduralno programiranje
 - Objektno-orijentirano programiranje
 - Generičko programiranje
 - Funkcionalno programiranje (od Java 8)
- Java Enterprise Edition (Java EE) – za razvoj distribuiranih mrežnih aplikacija i aplikacija baziranih na webu
- Java Micro Edition (Java ME) – dio Java SE, za razvoj aplikacija koje se izvode na uređajima ograničenih resursa (pametni satovi, MP3 *playeri*, TV oprema itd.)

Okruženja za razvoj Java aplikacija (IDE)

- Engl. *Integrated Development Environment*
- **Eclipse Oxygen** – koristi se na laboratorijskim vježbama
- IntelliJ IDEA
- Netbeans
- Uz sam IDE je potrebno instalirati i JDK

JRE i JDK

- JRE (engl. *Java Runtime Environment*) – okruženje za pokretanje Java aplikacija
 - Uključuje JVM
- JDK (engl. *Java Development Kit*) – sadrži sve što je potrebno za razvoj Java aplikacija
 - Sadrži JRE, prevoditelje i razne alate (poput Javadoc alata, *debuggera* itd.)

Java API

- Engl. *Application Programming Interface*
- Još se koristi i naziv „Java class library”
- Skup predefiniranih klasa i sučelja koje je moguće koristiti kod programiranja
- Dokumentacija u obliku HTML dokumenta
- Navigacija pomoću linkova
- Moguć *download* i korištenje *offline*

Faze kod razvoja i izvršavanja Java programa (1)

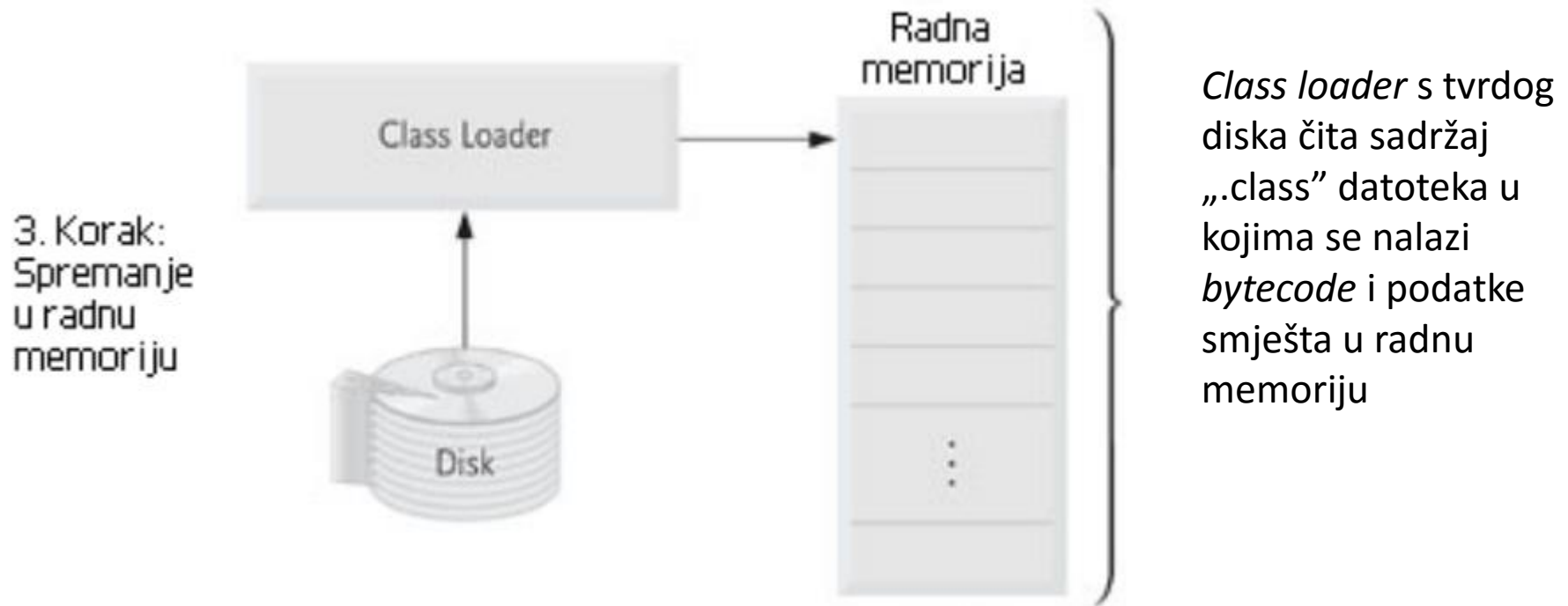
- Prva dva koraka: pisanje i prevođenje programa



- Unutar IDE-a objedinjena oba koraka
- Prevođenje se obavlja tijekom pisanja, a pretvorba u „class” datoteku nakon spremanja

Faze kod razvoja i izvršavanja Java programa (2)

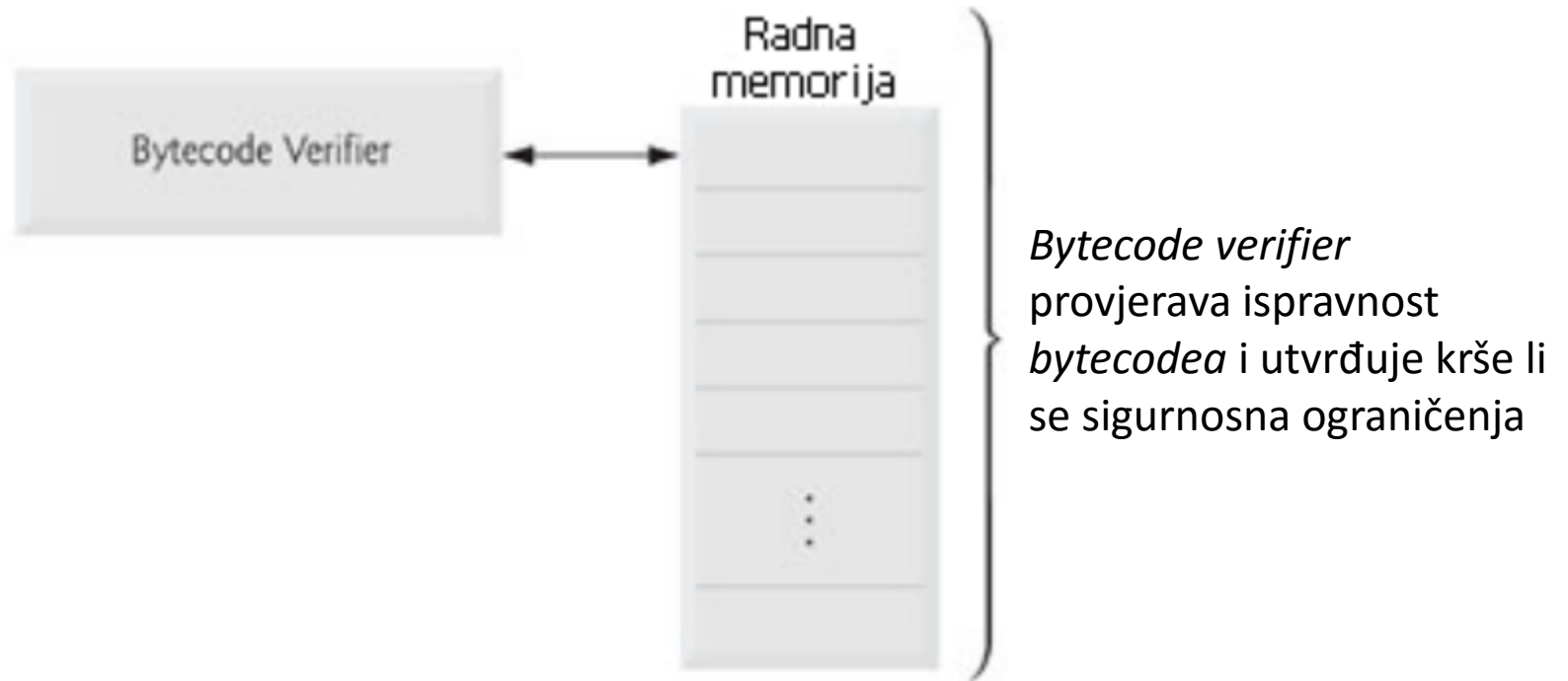
- Treći korak: premještanje programa u radnu memoriju



Faze kod razvoja i izvršavanja Java programa (3)

- Četvrti korak: provjera *bytecodea*

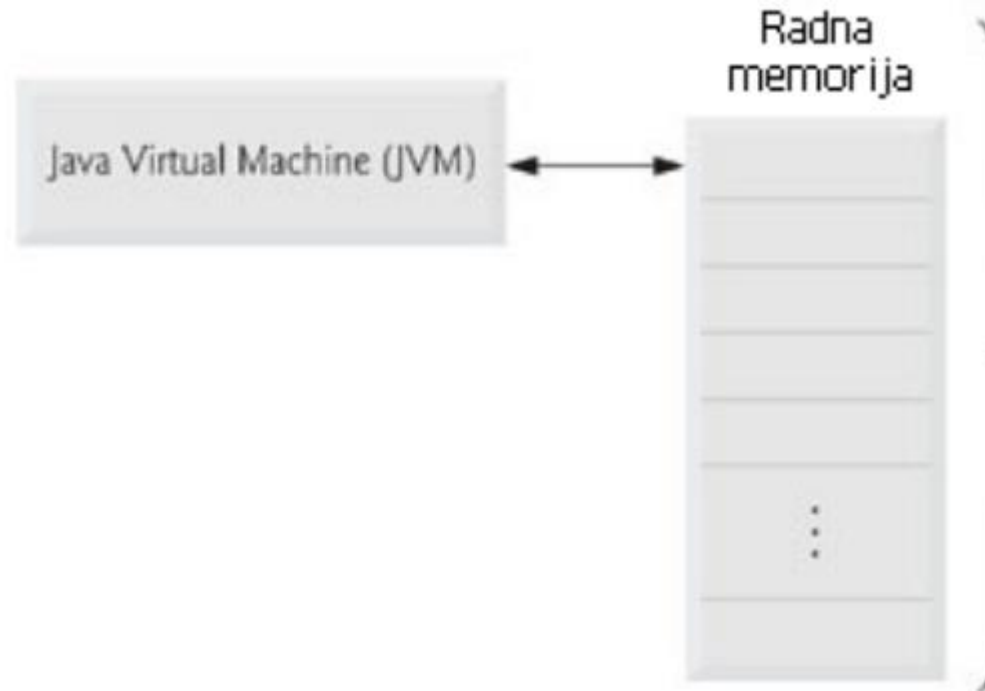
4. Korak:
Provjera
bytecodea



Faze kod razvoja i izvršavanja Java programa (4)

- Peti korak: izvršavanje programa

5. Korak:
Izvršavanje
programa



Prilikom izvršavanja programa JVM čita *bytecode* i prevodi ga u jezik koji „razumije” računalo na kojem se izvodi. Tijekom izvođenja program također zapisuje podatke u radnu memoriju.

Pitanja?
