

# Java i Eclipse IDE

# Java

- ▶ Java je platforma za izvršavanje programskog koda
  - jednom napisan i preveden program može da se izvršava na bilo kojoj platformi koja podržava Javu
  - Byte code (class file-ovi) se izvršava u JVM
- ▶ Java je programski jezik
  - Objektno orijentisani programski jezik
  - Opšte namene

# Instalacija Jave

## ▶ JRE

- Skup alata za pokretanje Java programa
- `java -version`

## ▶ JDK

- Vama treba JDK, skup alata za razvoj i pokretanje Java programa
- `javac -version`

## ▶ <http://www.oracle.com/technetwork/java>

poziv komandi iz konzole

# Kompajliranje i pokretanje Java programa

- ▶ Kompajliranje
  - `javac Test.java`
- ▶ Pokretanje (napomena: moramo biti pozicionirani u folderu gde se nalazi fajl `Test.class`)
  - `java Test`
  - Test klasa mora imati main metodu
- ▶ CLASSPATH – promenljiva okruženja koja definiše gde kompajler i JVM traže potrebne klase

poziv komandi iz konzole

# Eclipse IDE

- ▶ Razvojno okruženje
  - open-source i open-architecture
  - U ovom momentu verovatno vodeće okruženje za Javu
  - Nije ograničeno samo za Javu, koristi se i za Python i druge jezike
- ▶ Za razvoj aplikacija, kompajliranje java fajlova i njihovo pokretanje na ovom kursu korišćemo alat Eclipse koji nam ove radnje olakšava

# Eclipse – Instalacija

- ▶ <http://www.eclipse.org/downloads/>
- ▶ Eclipse preuzima poslednju instaliranu verziju Jave kao aktuelnu za pokretanje i kompajliranje Java programa
- ▶ Podešavanje instalacija JRE može se postići u Eclipse programu odlaskom na
  - Window->Preferences->Java->Installed JREs
- ▶ Podešavanje JDK i kompatibilnosti kompajlera u Eclipse programu odlaskom na
  - Window->Preferences->Java->Compiler
- ▶ Prilikom prvog pokretanja potrebno je definisati Workspace

# Eclipse – Okruženje

- ▶ Perspektive
  - promena
- ▶ Views
  - otvaranje
  - zatvaranje
  - pomeranje
- ▶ Centralni editor

# Eclipse – Rad sa projektima

- ▶ Otvaranje
  - ▶ Zatvaranje
  - ▶ Brisanje
    - Logičko
    - Fizičko
  - ▶ Import/Export
  - ▶ Paketi
  - ▶ Klase
  - ▶ Fajlovi, folderi, itd.
- 



# Eclipse – kompajliranje i pokretanje

- ▶ Build automatically / Build all / Build project
- ▶ Output folder
  - Podrazumevani je bin, ali može da se promeni
- ▶ Pokretanje
  - Run as -> Java application
  - Run as -> Run configuration...
  - Prečica – Alt + Shift + X, J
    - Run CTRL+F11
    - Debug F11

pokreni primer01

# Eclipse – Debug

- ▶ Debugovanje – otkrivanje grešaka
- ▶ Postoji posebna perspektiva za to
- ▶ Mnoštvo opcija
- ▶ Mnoštvo literature o tome
  - <http://www.vogella.com/tutorials/EclipseDebugging/article.html>
  - materijali/Eclipse – debug.pdf

# Eclipse – Plugins

- ▶ Plugins– proširenja
- ▶ Open architecture
- ▶ Mogu se pisati proširenja za Eclipse
- ▶ Postoji mnogo korisnih proširenja, neka postanu standardni deo nove verzije Eclipse–a, a neki ne – <http://marketplace.eclipse.org/>
- ▶ Više načina da se instalira proširenje
  - *Help->Install New Software...*
    - *Potreban je update site URL, npr.*  
[http://subclipse.tigris.org/update\\_1.10.x](http://subclipse.tigris.org/update_1.10.x)
  - Direktno kopiranje kroz neki file browser

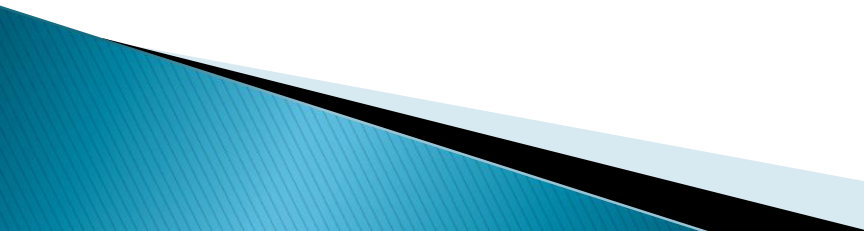
# Java

Osnove programskog jezika

# Osnove

- ▶ Objektno orijentisani jezik opšte namene
- ▶ Sintaksa je slična sintaksi C++, ali ima konceptualnih razlika
  - pokazivači, preklapanje operatora, višestruko nasleđivanje, destruktori, itd.
- ▶ Sve ključne reči programskog jezika Java se pišu malim slovima.
- ▶ Svaki Java program se sastoji iz deklaracije paketa, import sekcije, deklaracije atributa i definicije metoda. Sva četiri dela su opcionalna.

# Osnove

- ▶ Izrazi u programskom jeziku Java se obično završavaju simbolom tačka–zarez ';'.  

- ▶ Programski blok čine dve ili više naredbi ili izraza i on je ograden vitičastim zagradama {}. Ako imamo samo jednu naredbu ili izraz, ne moramo da je pišemo unutar vitičastih zagrada.
- ▶ Izvršenje programa počinje od *main* metode koja može primiti i parametre u vidu liste stringova.

objasnjavanje primer01

# Komentari

- ▶ Jednolinijski

- `//`
- `Ctrl + //`

- ▶ Višelinijski

- `/* ... */`
- `Ctrl + Shift + //`

- ▶ JavaDoc komentari

# Tipovi podataka

## ▶ Primitivni

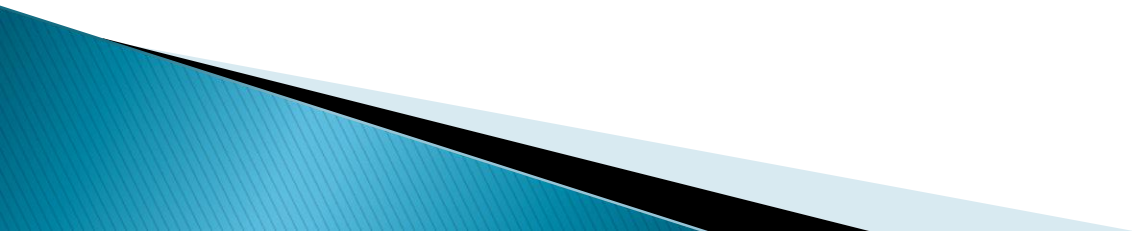
- kao lokalne promenljive i parametri metoda, čuvaju se na steku
- uvek se prenose po vrednosti

## ▶ Klase

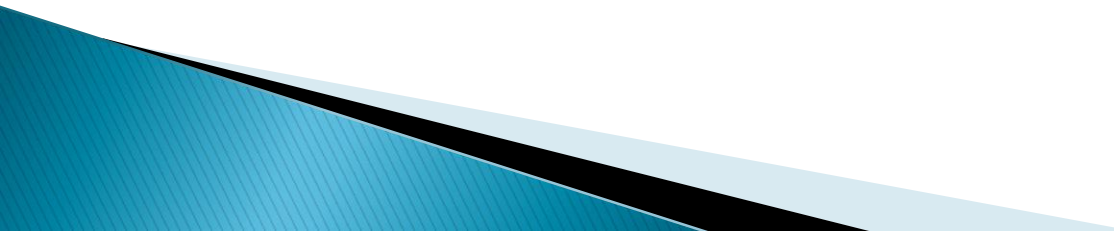
- Instance – objekti
- Reference na objekte
- Reference mogu biti na steku, ne i objekti, oni su na heap-u
- klasa String ima donekle specijalan tretman kod Java virtuelne mašine



# Metode

- ▶ funkcije i procedure
  - ▶ nema funkcija i procedura koje su definisane van klase
  - ▶ sintaksa: `povratna_vrednost naziv(parametri) {...}`
  - ▶ postoje i static, postoje i final metode, postoje i konstruktori
- 

# Literali

- ▶ “Predstave podataka”
  - ▶ Kasnije ćemo se baviti svim literalima za primitivne tipove – Celobrojni; Razlomljeni; Heksadecimalni; Oktalni; Binarni; Boolean; Znakovni
  - ▶ Sada su nam bitni String literali: **“ovo je tekst”**
  - ▶ String se ne mora kreirati konstruktorom, može mu se dodeliti string literal
- 

# Identifiers

- ▶ Nazivi dodeljeni klasama, metodama, promenljivama
- ▶ Niz karaktera, ali ne bilo kojih
  - Velika slova
  - Mala slova
  - cifre
  - `_` (underscore)
  - `$`
- ▶ Na početku ne može biti cifra
- ▶ Ne sme biti razmaka i bilo koji whitespace karaktera
- ▶ Ključne reči Java programskog jezika se ne smeju koristiti (for, if, class, main, ...)
- ▶ Case sensitive: car i Car nije isto (čak je i For, If dozvoljeno)

# Promenljive

- ▶ *tip identifikator*
- ▶ Važe do kraja bloka u kome su definisane
- ▶ Ako je primitivni tip ili String može se odmah dodeliti određenim literal,
  - na primer, String name = "Marko";
- ▶ Ako je objekat onda konstruktorom, ali o tome ćemo kasnije
  - Na primer, Student student = new Student(1234, "Marko", "Markovic");
- ▶ Stek ili heap ? – i o tome ćemo kasnije

# Operatori

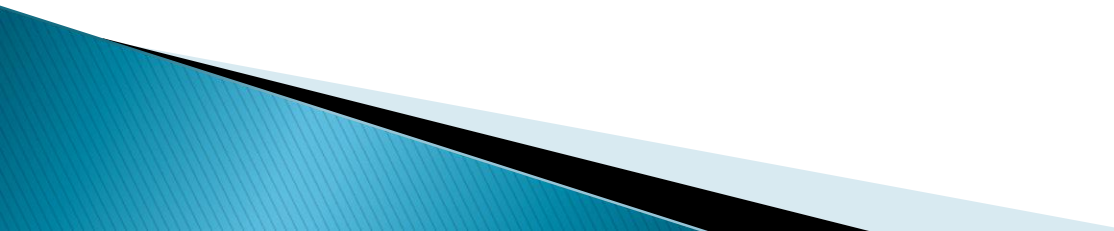
- ▶ Vrste operatora:
  - Aritmetički operatori
  - Relacioni i logički
  - Bit-operatori
  - Operatori pomeranja
  - Operator dodele
  - Cast operator
- ▶ Za današnje vežbe nam trebaju relacioni i logički:
  - Vraćaju true ili false
  - Da li je nešto tačno ili nije
  - Primer: `x == 5`
  - Za String: `name.equals("marko") == true`, a može i samo `name.equals("marko")`, ali ne može `name == "marko"`

# Konkatenacija stringova

- ▶ Operator + je preklopljen za klasu String (iako u javi nema preklapanja operatora)
- ▶ Kreira se novi objekat na heap-u
- ▶ Može se vršiti i konkatenacija sa primitivnim tipovima, ali i sa objektima (poziva se njihova toString metoda)
- ▶ Primer
- ▶ `String name = "marko" + " " + "markovic";`

objasnjavanje primer02

# Kontrola toka programa

- ▶ if else
  - ▶ switch
  - ▶ for
  - ▶ while
  - ▶ do while
  - ▶ break
  - ▶ continue
- 

# If else

- ▶ if (uslov)
- ▶                      akcija
- ▶ ili
- ▶ if (uslov)
- ▶                      akcija
- ▶        else
- ▶                      druga\_akcija

objasnjavanje primer03



# If else

- ▶ akcije mogu biti blok naredbi
- ▶ uslov je kombinacija relacionih i logičkih operatora
- ▶ else deo može ponovo kao akciju imati if naredbu (postoji i ograničenje o dubini ugnježdavanja if naredbi)

# Uslovni operator

- ▶ Uslovni operator
- ▶ `a=(i < 10 ? i * 100 : i * 10);`
- ▶ isto kao:
- ▶ `if (i < 10)`
- ▶ `a = i * 100;`
- ▶ `else`
- ▶ `a = i * 10;`