



Objektno orijentisano programiranje

Prolećni semestar 15/16
~branislavas

Tehnička pitanja



(1) Kolokvijum ~ 25p

(2) Pismeni deo ispita ~ 40p

- praktične prirode
- prag za izlazak: biće definisan

(3) Usmeni deo ispita ~ 35p

(4) Literatura:

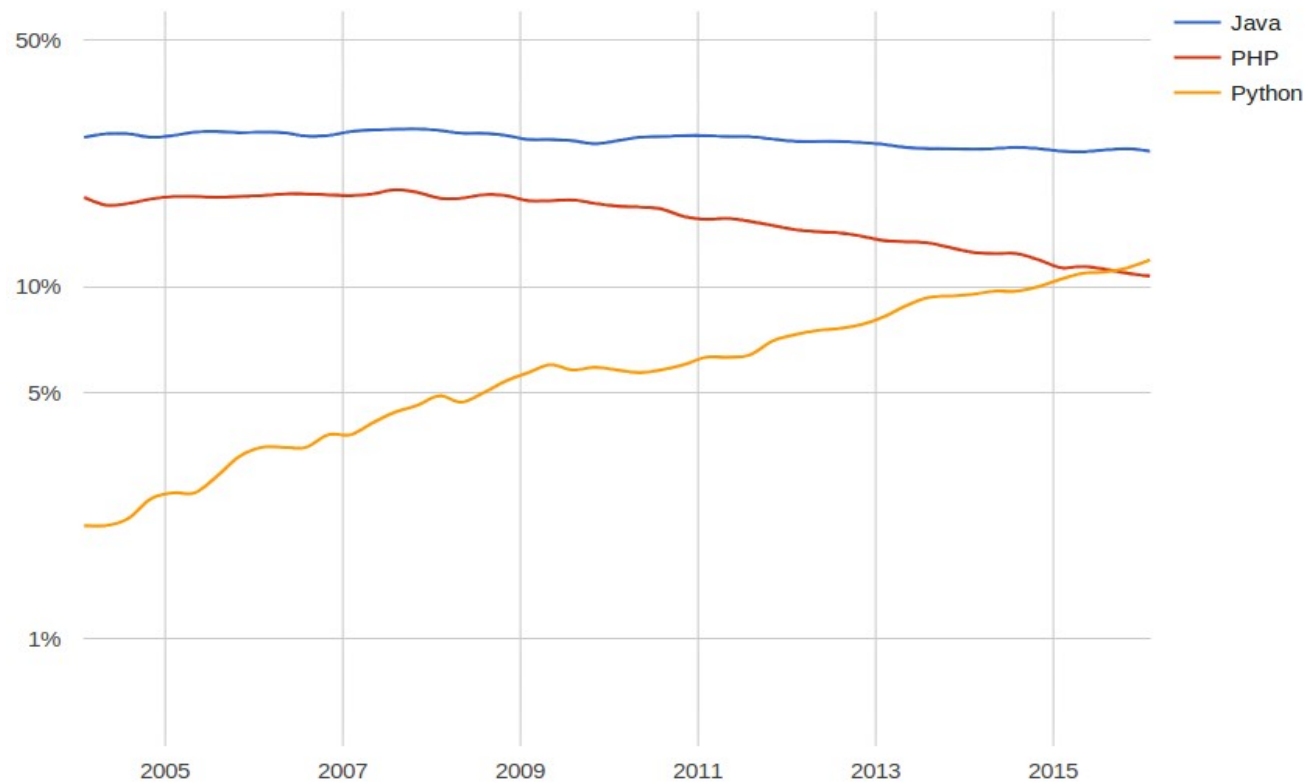
- Beginning Java 2 JDK 5 Edition, Ivor Horton
- Thinking in Java, Bruce Eckel

(5) Konsultacije

Zašto Java?



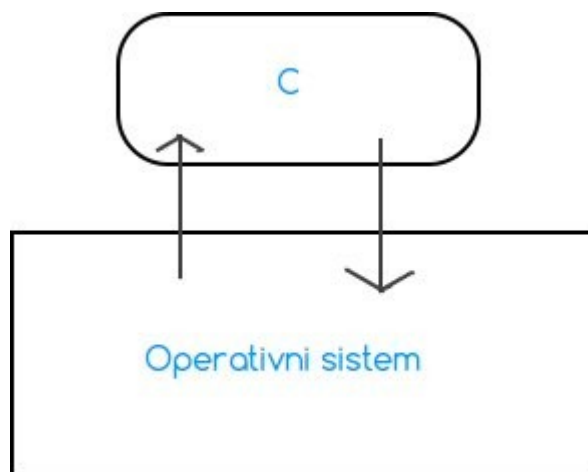
Top 3



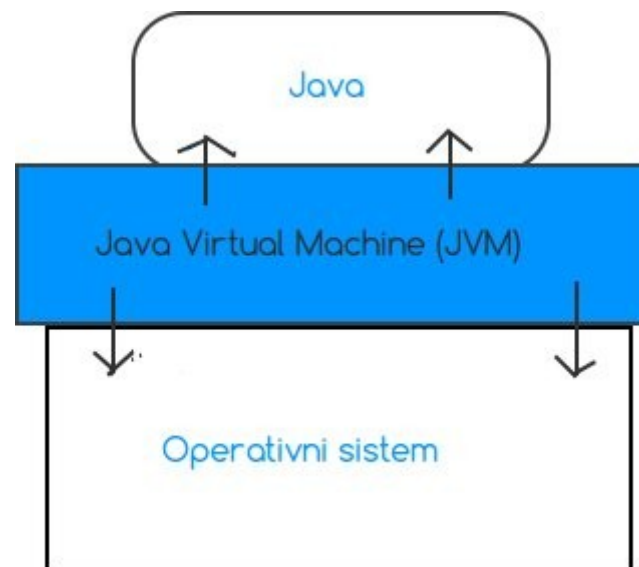
Kako je to moguće?



Na
primer...



Koristi resurse direktno



Koristi resurse JVM

JVM



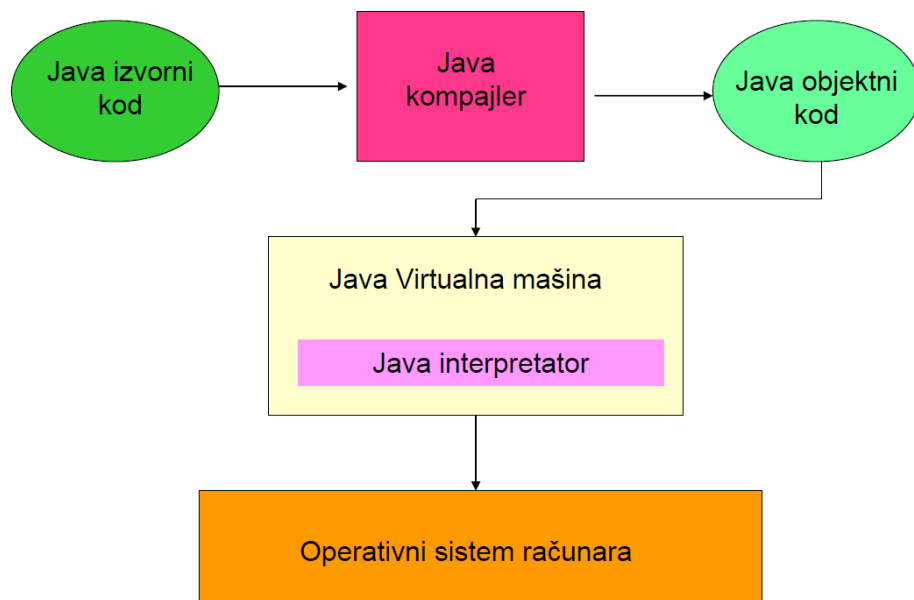
- (1) JVM je aplikativna virtualna mašina, tj. softverska implementacija hipotetičkog računara (postoji samo u unutrašnjoj memoriji računara i dizajnirana je za pokretanje jednog programa (aplikacije))
- (2) Implementacija JVM može biti drugačija na različitim operativnim sistemima, ali je rezultat izvršavanja Java programa isti bez obzira na operativni sistem i implementaciju JVM

JAVA application programming interface (API)



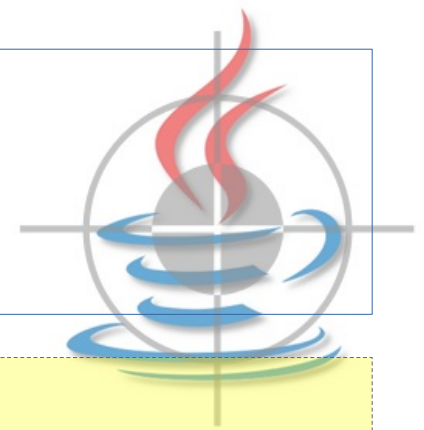
- (1) Skup komponenti koje obezbeđuju pisanje interaktivnih aplikacija u Javi – već kreiran skup klasa raspoloživih programeru za upotrebu, rad sa grafičkim korisničkim interfejsom i drugo (da ne bismo pisali sve ispočetka...)
- (2) Osnovne komponente Java platforme čine upravo Java virtuelna mašina i Java API

Pokretanje Java programa



- (1) Java prevodilac (poziva se komandom `javac`) prevodi Java izvorni kod (`.java`) u tzv. bajtkod (`.class`) koji je binarni i isti za sve platforme (*"prevedi jednom izvršavaj bilo gde"*)
- (2) Bajtkod (Bytecode) su mašinske instrukcije za JVM
- (3) Java program se izvršava pozivom Java interpretatora (komanda `java`), koji tumači bajtkod i izvršava akcije koje su specificirane u bajtkodu, unutar JVM

Instalacija



Java Development Kit

- Google: “jdk download”
- koji OS?

Java SE Development Kit 8u73

language and running on the Java platform.

See also:

- Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 8u73 Checksum
JDK 8u74 Checksum

Java SE Development Kit 8u73

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux ARM v6/v7 Hard Float ABI	77.73 MB	jdk-8u73-linux-arm32-vfp-hflt.tar.gz
Linux ARM v6/v7 Hard Float ABI	74.68 MB	jdk-8u73-linux-arm64-vfp-hflt.tar.gz
Linux x86	154.75 MB	jdk-8u73-linux-i586.rpm
Linux x86	174.91 MB	jdk-8u73-linux-i586.tar.gz
Linux x64	152.73 MB	jdk-8u73-linux-x64.rpm
Linux x64	172.91 MB	jdk-8u73-linux-x64.tar.gz
Mac OS X x64	227.25 MB	jdk-8u73-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.7 MB	jdk-8u73-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.08 MB	jdk-8u73-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.36 MB	jdk-8u73-solaris-x64.tar.Z
Solaris x64	96.78 MB	jdk-8u73-solaris-x64.tar.gz
Windows x86	181.5 MB	jdk-8u73-windows-i586.exe
Windows x64	186.84 MB	jdk-8u73-windows-x64.exe

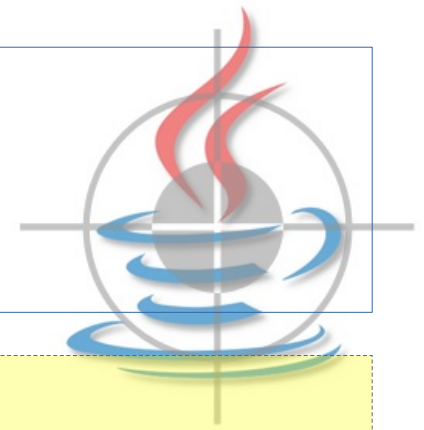
Java SE Development Kit 8u74

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

☐ Accept License Agreement ☒ Decline License Agreement

Product / File Description	File Size	Download
----------------------------	-----------	----------

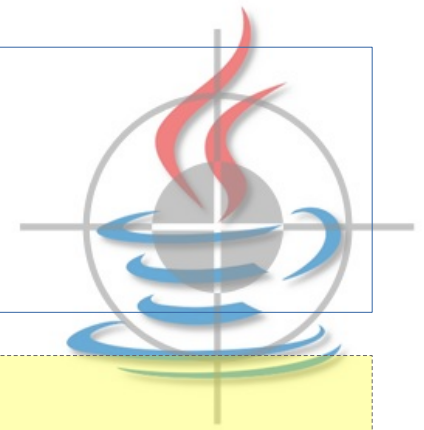
Prvi Java program



Zdravo.java :

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

Pokretanje

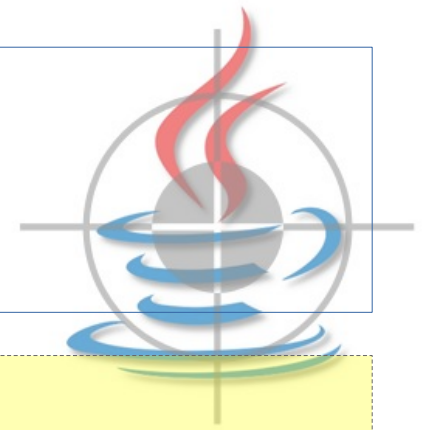


Zdravo.java => Zdravo.class

```
Command Prompt

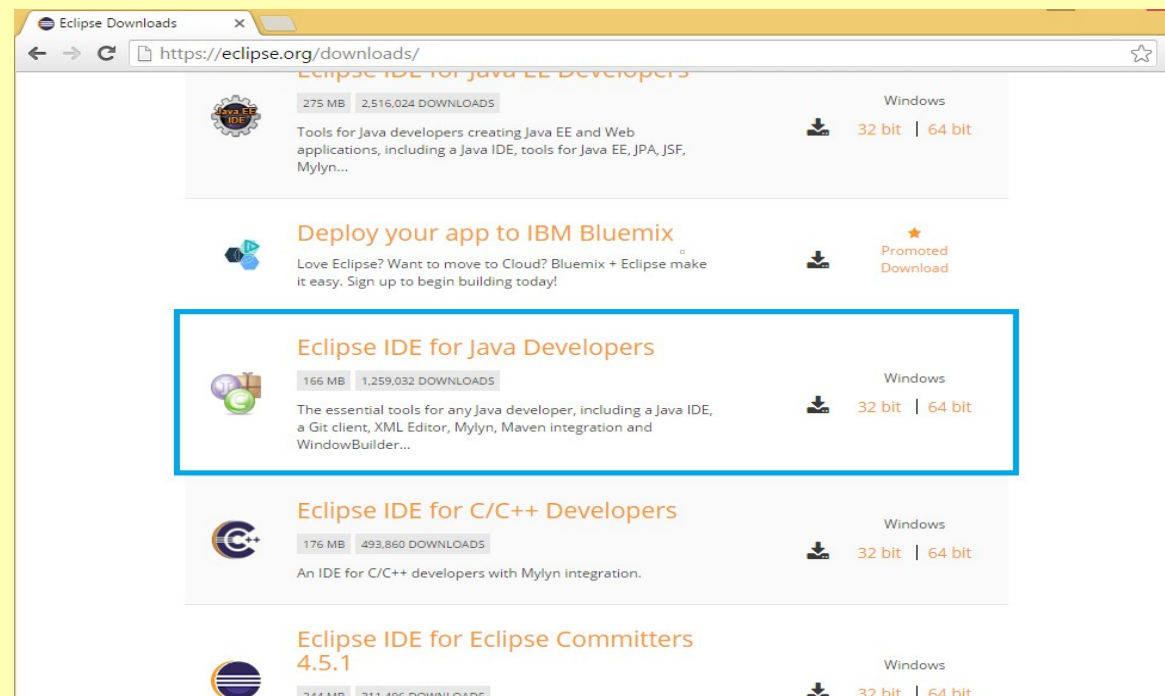
C:\Users\brislava\Desktop>set path=C:\Program Files\Java\jdk1.8.0_73\bin
C:\Users\brislava\Desktop>javac Zdravo.java
C:\Users\brislava\Desktop>java Zdravo
Zdravo svima!
C:\Users\brislava\Desktop>
```

Ipak...



(1) Grafička razvojna okruženja

- Eclipse, Net Beans ...
- Google: “eclipse download”
(nema instalacije – samo se raspakuje i pokrene)



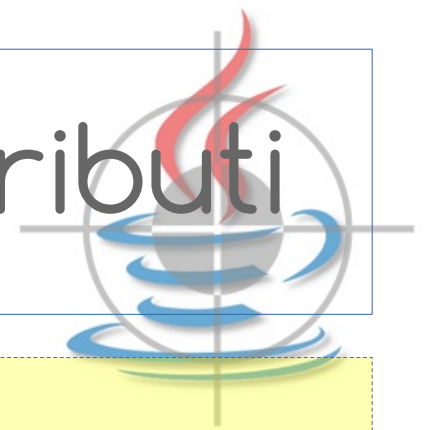
Modeliranje stvarnog sveta



Osoba

- ime, prezime, datum rođenja, JMBG, pol...
 - Predavač
 - kurs, staž, kabinet...
 - pregleda ispite, smišlja zadatke, sprema predavanja...
 - Student
 - ESBP poeni, godina upisa, broj indeksa...
 - sprema ispite, prati na vežbama...
- ~ Kako biste ovo predstavili u C-u? ~

Klase, objekti, metode i atributi



(1) Klasa Osoba

- Klasa Student

- Student Petar Petrović (128 ESBP, 3. godina, 102/14)

- Student Mihajlo Mihajlović

- Klasa Profesor

- Profesor Belić (analiza, 20 godina, kabinet 123)

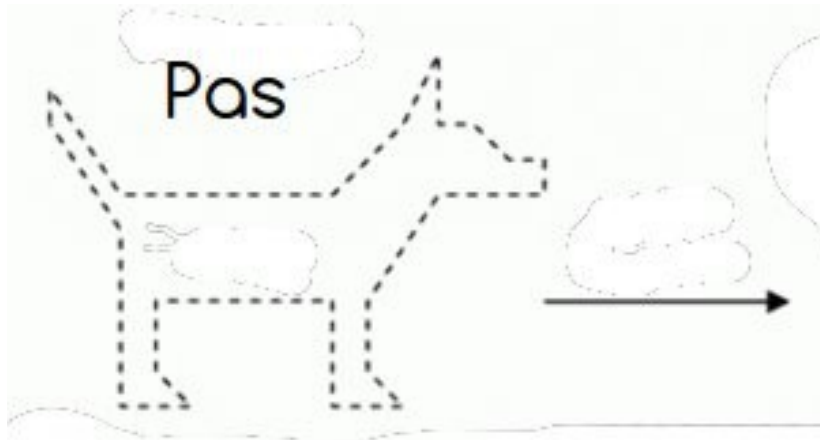
- Profesor Žutić

(2) Konkretni primerci klase su **objekti**

(3) Osobine tih objekata su **atributi**

(4) Njihovi aktivnosti su **metode**

Na primer...



Maza



Osobine

- boja dlake
- boja očiju
- rasa
- visina
- masa
- starost

Metodi

- sedi
- lezi
- daj šapu
- donesi loptu

Osobine

- zlatna dlaka
- smeđe oči
- labrador
- 30cm
- 4kg
- 5 meseci

Metodi

- sedi
- lezi
- daj šapu
- donesi loptu

Formalno...



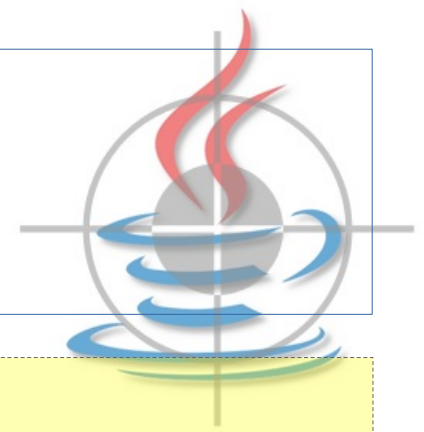
- (1) **Objekat** je integralna celina podataka i procedura za rad sa njima
- (2) **Objektno orijentisano programiranje** je programska paradigma zasnovana na skupu objekata koji dejstvuju međusobno. Glavne obrade zasnivaju se na rukovanju objektima
- (3) **Klasa** je opis (specifikacija) jedne vrste objekata. Ime Java klase počinje velikim slovom. Na primer, **Osoba** je klasa. Definicija svake Java klase nalazi se u posebnoj datoteci koja ima isto ime kao i klasa, a ekstenziju **.java**.

Formalno...(2)

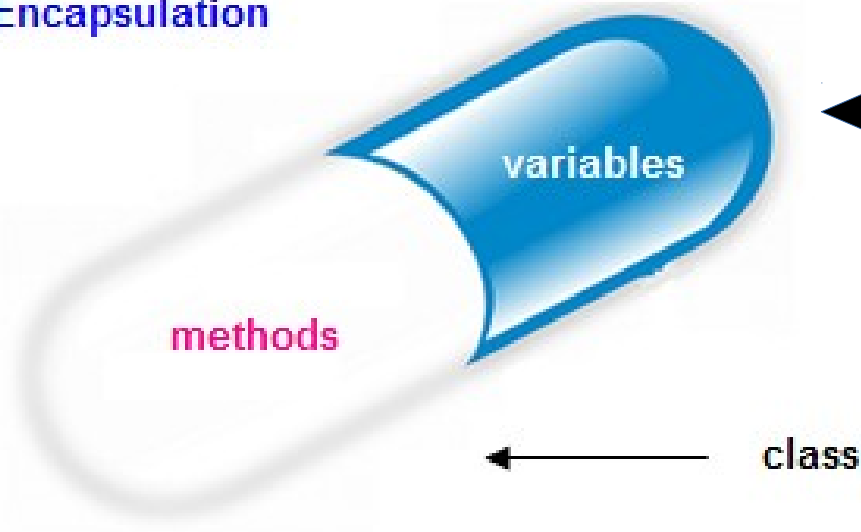


- (1) **Instanca klase** je konkretan objekat klase
- (2) **Atributi** su podaci koji opisuju svojstva objekata te klase. Mogu biti promenljive primitivnog tipa ili objekti neke druge klase
- (3) Klasa pored podataka (instancnih promenljivih/atributa) specifikuje i šta sve može da se radi sa objektima te klase, odnosno, definiše moguće operacije nad objektima klase. To se postiže blokom programskog koda koji se naziva **metod**. Sintaksno gledano, **metod** je zapravo funkcija koja je deo objekta i kojom se realizuje poruka upućena objektu

Bitni pojmovi OOP



Encapsulation



(1) I još:

- Apstrakcija
- Konkretizacija
- Enkapsulacija
- Nasleđivanje
- Polimorfizam

Nasleđivanje



šolja



šolja
za
kafu



šolja
za
čaj



šolja za
espresso



šolja za
latte

U Javi klasa Object je klasa nad klasama



- (1) Prilikom projektovanja programa uočavaju se veze između pojedinih klasa, gde je bitna uloga nasleđivanja
- (2) **Nasleđivanje** je postupak definisanja novih klasa na osnovu postojećih. Na taj način se uspostavljaju relacije između jedne i više drugih klasa
- (3) U Javi imamo standardnu klasu **Object** koja je natklasa svih klasa

Organizacija Java programa



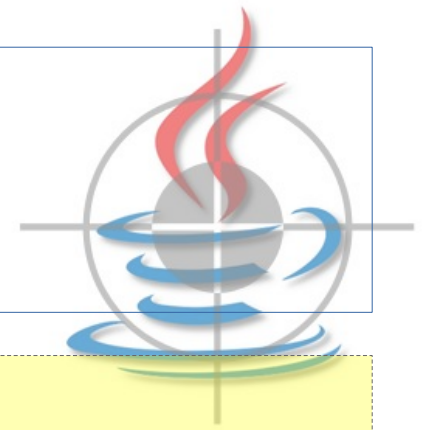
- (1) Java program se sastoji iz više klasa
- (2) Klase mogu biti organizovane po paketima
- (3) U file system-u, paketi se vide kao folderi
- (4) Biblioteka klasa u Javi je kolekcija već postojećih klasa
- (5) Korisnik sam može da pravi svoje pakete koji će sadržati odgovarajuće klase

Standardni paketi...



- (1) **java.lang** – osnovne karakteristike jezika, rad sa nizovima i stringovima; klase iz ovog paketa su uvek dostupne, tj. automatski se uključuju u program (npr. klase Integer, String, Character, Math, ...)
- (2) **java.io** – klase za ulazno/izlazne operacije
- (3) **java.util** – veliki broj korisnih klasa za razne namene (Scanner, ArrayList (uopšteni niz), Stack, ...)
- (4) **java.awt**, **java.swing** – paketi za rad sa grafičkim komponentama koji su ranije korišćene
- (5) **javafx** – osnovni paket (koji sadrži druge potpake) za rad sa JavaFX grafičkim komponentama

import deklaracija



(1) Upotreba klasa iz nekog paketa omogućena je import deklaracijom koja se navodi na početku programa (drugim primerom se uključuju sve klase iz paketa):

```
import java.util.ArrayList;
```

```
import java.util.*;
```

(2) Može se koristiti klasa iz nekog paketa i bez import deklaracije, ali se mora navesti njeno puno ime u programu:

```
java.util.ArrayList a;
```

Gde se biblioteke nalaze fizički?



- (1) Standardne klase su spakovane u jednu kompajliranu datoteku `rt.jar` koja se nalazi u `jre/lib` poddirektorijumu koji se kreira prilikom instalacije JDK-a
- (2) Standardne JavaFX klase spakovane su u datoteku `jfxrt.jar` koja se nalazi u poddirektorijumu `jre/lib/ext`

Glavni metod



(1) Svaki Java program mora imati klasu koja u sebi sadrži glavni metod:

```
public static void main (String[] args) { ... }
```

(2) Prilikom pokretanja programa, kompajler “traži” takvu klasu i prvo pokreće kôd koji je sadržan unutar glavnog metoda

(3) Ključne reči: public, static, void

(4) Ime metoda: main

(5) Argumenti metoda: String, String[], args

(6) Sve zajedno: **potpis metoda**

Objašnjenje



public – globalno dostupna klasa i metod

static – metod je dostupan i kada ne postoje objekti klase

void – metod nema povratnu vrednost

String[] args – niz argumenata komandne linije. Argumenti su objekti klase String, kojom se opisuju niske karaktera. Klasa String je definisana u paketu java.lang

System – standardna klasa koja omogućava rad sa standardnim ulazom/izlazom. Nalazi se u paketu java.lang

out – objekat (klase PrintStream) koji je član klase System i predstavlja standardni izlazni tok

println() – metod koji pripada objektu out (metod klase PrintStream) koji ispisuje string zadat između dvostrukih navodnika

Elementarne konstrukcije (1)



(1) Elementi jezika koje prevodilac izdvaja tokom prevođenja programa kao nedeljive celine (**token**):

– Identifikatori

- nizovi Unicode karaktera (може u ћирилица!)
- nemaju ograničenje dužine
- počinju slovom, podvlakom ili znakom \$
- ostali karakteri: slova, cifre, _ ili \$
- ne smeju se koristiti ključne reči
- poželjno koristiti kamilju notaciju

Elementarne konstrukcije (2)



– Literali

- Celobrojni
 - dekadni (ne počinju 0): 123, -789
 - oktalni (počinju cifrom 0): 0377777, 01256775
 - heksadekadni (počinju 0x ili 0X): 0xAA1234, 0XABCDEF
 - binarni (počinju sa 0b ili 0B): 0b000100101110, 0b11001101111100
 - ne postoji ključna reč **unsigned**, svi su označeni!

- Realni
 - podrazumevano tipa **double**
 - može float: 1.423, 1.423F, 2.54E8, 9E-28F ili **kastovanje**
- Logički
 - true, false
- znakovni
 - konstantna vrednost tipa char: 'a', '\n', '3', '\\', '\u0041' (koji karakter?)
- stringovni
 - “Ovo je jedan String literal”

Elementarne konstrukcije (3)



- Separatori: () {} [] ;
- Beline
- Operatori
 - aritmetički, relacioni, bitovski, logički, dodele, razni
- Komentari
 - C stil /*... */
 - jednolinijski komentari //...
 - dokumentacioni komentari

```
/**
```

```
 *  
  ...
```

```
 *  
  ...
```

```
 */
```

Ključne reči Java jezika

© Copyright 1998-2002 by Markus Falkhausen, all rights reserved.
Most recent version: www.falkhausen.de

Category	Keyword	Example	Incidents per 100 Lines
Primitive types	boolean	<code>boolean isOpen = true;</code>	.82
	byte	<code>byte i1 = -128;</code>	.67
	char	<code>char c = '\uFFFF';</code>	.27
	short	<code>short i2 = -32768;</code>	.14
	int	<code>int i = -2147483648;</code>	4.35
	long	<code>long i8 = -9223372036854775808L;</code>	.19
	float	<code>float x4 = -3.402823e+38f;</code>	.32
	double	<code>double x = -1.79769313486231e+308;</code>	.33

Control flow	for	<code>for (int i=0; i<10; i++) {...}</code>	.63
	do	<code>do {...} while (i<10);</code>	.01
	while	<code>while (i<10) {...}</code>	.22
	if	<code>if (i==10) {...}</code>	3.41
	else	<code>else if (i<0) {...}</code>	.92
		<code>else {...}</code>	
	switch	<code>switch (i) {</code>	.08
	case	<code>case 1: ...</code>	.44
		<code>break;</code>	
	default	<code>default:...</code>	.05
	break	<code>break label;</code>	.28
	continue	<code>continue label;</code>	.03
	return	<code>return i;</code>	2.91
	try	<code>try {</code>	.30
	throw	<code>... throw new MyException ();</code>	.38
		<code>...</code>	
	catch	<code>} catch (MyException ex) {...}</code>	.34
	finally	<code>finally {...}</code>	.01
	throws	<code>void func () throws MyException {...}</code>	.53

Category	Keyword	Example	Incidents per 100 Lines
Modifier	public	<code>public int i;</code>	4.65
	protected	<code>protected int i;</code>	.64
	private	<code>private int i;</code>	1.10
	static	<code>static int i;</code>	1.56
	final	<code>final int i;</code>	.92
	abstract	<code>abstract void func ();</code>	.21
	synchronized	<code>synchronized (object) {...}</code>	.26
	native	<code>native int func ();</code>	.05
	transient	<code>public transient int i;</code>	.07
	volatile	<code>public volatile int i;</code>	0

Classes	class	<code>class A {...}</code>	.86
	interface	<code>interface I {...}</code>	.07
	extends	<code>class B extends A {...}</code>	.44
	implements	<code>class B implements I {...}</code>	.22
	package	<code>package de.falkhausen.util;</code>	.36
	import	<code>import java.awt.*;</code>	1.40

Miscellaneous	(true)	<code>boolean isOpen = true;</code>	.45
	(false)	<code>boolean isOpen = false;</code>	.53
	(null)	<code>Object obj = null;</code>	2.00
	void	<code>void func () {...}</code>	2.15
	this	<code>this.x = x;</code>	1.16
	new	<code>Object obj = new Object ();</code>	2.71
	super	<code>super ("text");</code>	.43
	instanceof	<code>if (o instanceof String) String s = (String) o;</code>	.25

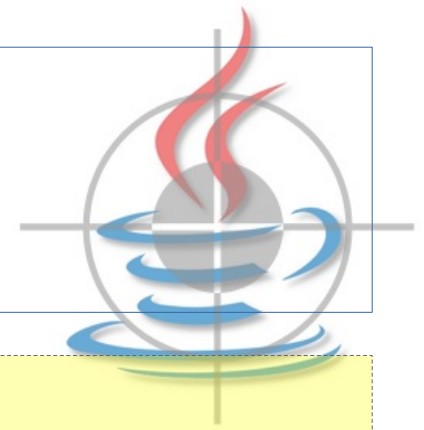
Reserved/New `const, goto / strictfp, assert` 0

Java vs. C



- (1) nema automatske konverzije iz boolean u int i obrnuto, pa ne može `if(a)` , već mora `if(a != 0)`
- (2) promenljiva se može uvesti bilo gde u bloku, pa čak i u `for(int i=0; i<n; i++)`
- (3) `if`, `else`, `switch`, `for`, `while`, `do`, `break`, `continue` - kao u C-u
- (4) nema operatora za pokazivače, niti ključnih reči `sizeof`, `typedef`, `struct`, `union`
- (5) promenljiva klasnog tipa je **OBAVEZNO** referenca/pokazivač na objekat

Java vs. C



(1) stringovi su objekti. Klase String i StringBuffer (java.lang)

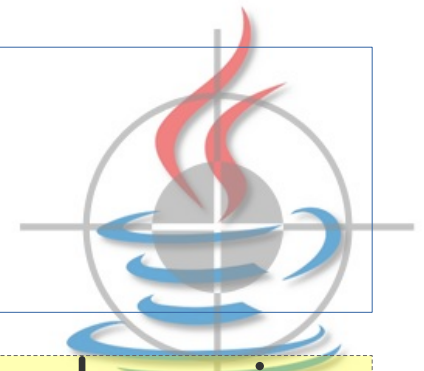
- ime.length()
- ime.charAt(0)
- ime.charAt(3)
- ime.charAt(ime.length()-1)

(2) sabiranje String-ova

(3) argumenti komandne linije (String [] args)

- args.length()
- args[0], args[1]

java.util.Scanner



- (1) Klasa kojom se implementira jednostavni parser koji parsira primitivne tipove i stringove koristeći regularne izraze
- (2) Ulaz može biti fajl ili tok, uključujući i standardni ulazni tok `System.in`
- (3) Ulaz se deli na tokene koristeći kao podrazumevani delimiter beline (' ', '\n', '\t', ...).
- (4) Dobijeni tokeni se mogu konvertovati u vrednosti različitih tipova koristeći razne `next...` metode