

Predavanja (3)	Konsultacije
sreda: 11:15 – 13:50	sreda: 15 – 17 h
Učionica 16	Kabinet 43

**Uvod u programiranje**  
dr Ninoslava Savić

## Uvod u programiranje

- Fond časova: 3+3
- Broj ESPB: 7
- Ocena znanja (max. broj poena je 100)

Predispitne obaveze (min 33)	65 poena
Prisustvo na predavanjima i vežbama	5
Provera znanja u toku nastave	40
Aktivnost studenata na predavanjima i vežbama	10
Praktičan rad	10
Završni ispit (min 18)	35 poena
Pismeni ispit (sa usmenim delom)	35

Minimum bodova za prolaz: 33 + 18 = 51

## Sadržaj predmeta

- Razvoj i podela programskih jezika
- Metode projektovanja programa
- Klasifikacija tipova podataka
- Osnovni koncepti sintakse programskog jezika
  - Tipovi podataka programskog jezika
  - Kontrola programskog toka
  - Objektni koncepti programskog jezika
- Programiranje klasičnih algoritama
- Programiranje klasa i potklasa
- Kreiranje poslovnih aplikacija – prvi nivo

## Literatura

- Materijali sa predavanja i vežbi za predmet *Uvod u programiranje* - na sajtu Škole
- Lemay L., Cadenhead R., *Java - Naučite za 21 dan*, Kompjuter biblioteka, 1999. - prvih 7 poglavlja (kao i novija izdanja ove knjige)
- Eckel B., *Misliti na Javi*, Mikro knjiga
- Free alati i dokumentacija sa službenog sajta
  - <https://www.oracle.com/>
  - <https://netbeans.org/downloads/8.0.2/>

## Tehnologije

- Programski jezik
  - Java
- Razvojno okruženje
  - JDK (Java Development Kit)
  - NetBeans IDE
- Sistem za upravljanje bazama podataka
  - MySql

5

## Metode programiranja (projektovanja programa)

- Klasično programiranje  
(sa upotrebom *goto* instrukcije, tkz. "špageti programi")
  - *Cobol, Fortran*
- Strukturirano programiranje
  - *Pascal, C, ..*
- Objektno programiranje
  - *Java, C++, C#*

6

## Klasifikacija tipova podataka

- Prosti tipovi podataka
  - ugrađeni u programski jezik  
(celobrojni, realni, logički, karakter tip, ...)
- Složeni (strukturirani) tipovi podataka
  - neka struktura nad prostim tipom podatka  
(niz, skup, lista, stek, ...)
  - definiše ih programer

7

## Logičke kontrolne strukture (LKS)

- LKS su sredstvo kontrole programskog toka
- Važi sledeće tvrđenje:  
Svaki algoritam može se predstaviti kombinacijom tri osnovne logičke kontrolne strukture:
  - Sekvencija  
(niz naredbi koje se izvršavaju tačno jednom, redoleđom kojim su napisane)
  - Selekcija (grananje)
  - Iteracija (petlja)
- Na ovom tvrđenju počiva metoda strukturiranog programiranja – programiranja `bez goto instrukcije`

8

## Algoritam, program

- **Algoritam** je konačan, uređen skup, nedvosmislenih izvršnih koraka za rešavanje određenog problema:
  - u konačnom vremenskom periodu,
  - sa konačnom količinom podataka.
- **Program** predstavlja:
  - sekvencu instrukcija napisanih u nekom programskom jeziku
  - koje se mogu prevesti u instrukcije koje računar može direktno izvršiti (mašinske instrukcije)

9

## Programski jezik

- Programskim jezikom definiše se:
  - skup znakova,
  - skup službenih reči,
  - skup instrukcija,
  - sintaksa
    - skup pravila za građenje složenih konstrukcija jezika (reči, iskaza, instrukcija) i njihovo kombinovanje u programu,
  - semantika (značenje instrukcija u programu)

10

## Generacije programskih jezika

- Mašinski jezik (u binarnom obliku)
- Asemblerski i makroassemblerski jezici (simbolički jezici)
- Proceduralni programski jezici (jezici treće generacije – jezici visokog nivoa): Basic, Fortran, Pascal, Cobol, C, C++, Java, C#
- Neproceduralni programski jezici (jezici 4. generacije, deklarativni programski jezici - jezici vrlo visokog nivoa): Prolog, SQL, generatori programa...
- Jezici veštačke inteligencije i prirodni programski jezici

11

## Podela programskih jezika

- Proceduralni programski jezici
  - FORTRAN, COBOL, Basic, Pascal, Ada
- Objektno-orijentisani programski jezici
  - SIMULA, Smalltalk, C++, Java, C#
- Deklarativni (logički) programski jezici
  - Prolog, SQL, ...
- Funkcionalni programski jezici
  - LISP, Scheme, ...

12

## Programski jezik Java

- 1995. godine Sun Microsystems lansirala programski jezik Java (*James Gosling*)
- Java nastaje kao jezik Internet programiranja
- Java je u potpunosti objektno-orientisan jezik
- Osobine:
  - objektno-orientisani jezik, nezavistan od računarske platforme, prenosiv, bezbedan
- Objektno-orientisano programiranje (OOP) je metodologija razvoja softvera u kojoj se program:
  - realizuje pomoću objekata koji međusobno komuniciraju
  - objekti se kreiraju korišćenjem šablona koji se nazivaju *klase*

13

## Programski jezik Java

- **Platformska nezavisnost** je osobina programa da se izvršava bez modifikacija u različitim radnim okruženjima.
- Kako Java realizuje ovo svojstvo?
  - Java programi se prevode u format koji se naziva *bajtkod* (još uvek nije mašinski kod)
  - bajtkod se izvršava u okviru bilo kog operativnog sistema koji sadrži interpreter programskog jezika Java
- Java Development Kit (JDK) – besplatno dostupan na <https://www.oracle.com>
- Java integrisana razvojna okruženja: **NetBeans IDE, Eclipse, JBuilder, ...**

14

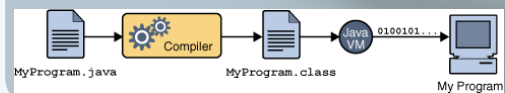
## Programski jezik Java

- Objektno-orientisano programiranje je pristup razvoju programa pomoću modelovanja objekata iz realnog sveta.
- Java implementira principe objektno-orientisanog programiranja na sledeći način:
  - Svaki program se sastoji od elemenata koji se nazivaju *klase* i postupaka za kreiranje i manipulisanje objektima ovih klasa - *metoda*
  - Definisane klase podrazumeva:
    - definisanje atributa klase i
    - definisanje metoda (ponašanja klase)
  - Međusobno povezivanje klasa vrši se:
    - nasleđivanjem i
    - korišćenjem paketa i interfejsa.

15

## Programski jezik Java

- Izvorni Java kod piše se u tekstualnim fajlovima sa ekstenzijom *.java*
- Izvorni fajlovi se kompajliranjem pretvaraju u fajlove sa ekstenzijom *.class*
- *.class* fajl sadrži *bajtkod* koji još uvek nije razumljiv procesoru (nije izvršni kod)
- Bajtkod se izvršava pokretanjem Java virtuelne mašine (*Java Virtual Machine - JVM*)
- Java programi se pokreću uz pomoć **JVM**



16

## Opšta sintaksa Java aplikacije

- Svaka Java aplikacija je klasa
- Main metod je polazna tačka - početak izvršnog dela svakog Java programa
- Svaka oznaka početka bloka ( { ) mora se zatvoriti ( } )

```
class <ime_klase> {
    // deklaracije atributa
    // definicije metoda
    .....
    // zaglavlje main metoda
    public static void main(String args[]){
        _____;
        _____;
        .....
    }
}
```

17

## Primer Java programa

- Program koji ispisuje "Zdravo!"

```
class ZdravoApp {
    public static void main(String[] args) {
        System.out.println("Zdravo!");
        // ispis stringa ← komentar
    }
}
```

- Tekst programa se zapisuje u fajl sa imenom ZdravoApp.java

18

## Pravila sintakse Java programa i konvencije za imenovanja

- Primer programa za ispis zbira dva cela broja:

```
public class Zbir {
    public static void main(String[] args) {
        int zbir = 35 + 257;
        System.out.println("35 + 257 = " + zbir);
    }
} // ime fajla u kome je ovaj program - Zbir.java
```

- naziv fajla = nazivu klase.java
- programski kod se smešta unutar klase
- izvršavanje počinje od metoda main
- sve naredbe se završavaju tačka-zarezom ;
- blokovi koda su ograničeni sa { }

19

## Java – pokretanje programa

1. Editovanje  
----> ZdravoApp.java
2. Kompajliranje:  
javac ZdravoApp.java
- Kompajler generiše fajl:  
----> ZdravoApp.class
3. Pokretanje programa:  
java ZdravoApp

20

## Osnovni elementi Java jezika

- Osnovni elementi Java jezika pomoću kojih se grade složenije konstrukcije jezika su:
  - Osnovni skup znakova
  - Identifikatori (imena)
  - Literali (konstante)
  - Specijalni simboli
  - Rezervisane reči

21

## Osnovni skup znakova

- Slova
    - A-Z, a-z
  - Cifre
    - 0,1,2,...,9
  - Specijalni simboli
    - ` , ` , \$ , \_ , simboli operatora (sledeći slajd)
  - Sintaksa Java jezika **pravi razliku između malih i velikih slova (case sensitive)!**
- Primer:
- *Ime* i *ime* su dva različita identifikatora

22

## Grupe operatora

- aritmetički (+, -, \*, /, %, ++, --)
- poređenja (==, !=, <, >, <=, >=)
- logički (&&, ||, ^, !)
- uslovni operator (? :)
- nad bitovima (&, |, ^, <<, >>)
- dodele (=, +=, -=, \*=, /=, %=)
- nad objektima (*instanceof*)

23

## Spisak Java operatora

>	veće		disjunkcija i disjunkcija nad bitovima
<	manje	^	ekskluzivna disjunkcija
!	negacija	%	ostatak pri deljenju ( mod )
~	negacija nad bitovima	<<	pomeranje bitova u levo
?	uslovni operator	>>	pomeranje bitova u desno
==	jednako kao relacioni operator	>>>	pomeranje bitova u desno
<=	manje ili jednako	=	jednako kao operator dodele
>=	veće ili jednako	+=	dodala sa prethodnim sabiranjem
!=	nejednako	-=	dodala sa prethodnim oduzimanjem
&&	uslovna konjunkcija	*=	dodala sa prethodnim množenjem
	uslovna disjunkcija	/=	dodala sa prethodnim deljenjem
++	povećanje vrednosti za jedan	&=	dodala sa prethodnom primenom &
--	smanjenje vrednosti za jedan	=	dodala sa prethodnom primenom
+	sabiranje ili predznak	^=	dodala sa prethodnom primenom ^
-	oduzimanje ili predznak	%=	dodala sa prethodnom primenom %
*	množenje	<<=	dodala sa prethodnom primenom <<
/	deljenje	>>=	dodala sa prethodnom primenom >>
&	konjunkcija i konjunkcija nad bitovima	>>>=	dodala sa prethodnom primenom >>>

24



## Identifikatori (imena)

- Koriste se za imenovanje različitih elemenata programa:
  - promenljivih, metoda, klasa, objekata, interfejsa, ...
- Pravilo za građenje identifikatora:
  - niz sastavljen od: slova, cifara, znaka `_`
  - počinje slovom
  - razlikuje se od svake rezervisane reči

Primer:

- Ispravni identifikatori
  - `ime_i_prezime`, `temp`, `var2`, `upisiBroj`, `Max`
- Neispravni identifikatori
  - `Ime-i-prezime`, `licni_dohodak`, `real`, `11max`

25

## Literali (konstante)

Literali u Javi mogu biti:

- **numerički literali** (brojevi)
  - 234 (celobrojni)
  - 3.14 (zapis u pokretnom zarezu)
  - 4.0E-74 (eksponencijalni zapis)
- **znakovne konstante**
  - između apostrofa - npr. `'h'`
- **stringovne konstante**
  - između navodnika - npr. `"Novi Sad"`
- **logičke konstante** - `true` i `false`
- **konstanta** `null`

26

## Rezervisane reči

- Koriste se za kreiranje naredbi Java programa i imaju tačno određeno značenje
- Ne mogu se koristiti kao identifikatori

```
abstract default if package this
boolean do implements private throw
break double import protected throws
byte else instanceof public transient
case extends int return null
try const for new switch strictfp
continue while goto synchronized super
catch final interface short void
char finally long static volatile
class float native assert
```

27

## Pravila imenovanja - konvencije

- **Nazivi promenljivih:** počinju malim slovom, kod složenica - svaka naredna reč počinje velikim slovom ili se odvaja donjom crtom
- **Nazivi metoda:** obično su glagoli, počinju malim slovom, dok svaka naredna reč počinje velikim slovom
  - primer: `promeniBojuOkvira()`
- **Nazivi klasa i interfejsa:** počinju velikim slovom
  - primer: `Student`, `Faktura`
- **Nazivi konstanti:** počinju velikim slovom
  - primer: `Font.BOLD`, `Font.ITALIC`

28

## Komentari

- Delovi programa koji ne utiču na suštinu programa i zanemaruju se prilikom prevođenja (kompajler ih ignoriše, važni su za programera i za dokumentovanje programa)

Simboli	Upotreba
//	Komentar od simbola // do kraja reda
/* . . . . . */	Komentar na proizvoljnom mestu u kodu
/** . . . . . */	Komentar za automatsko generisanje dokumentacije programa

29

## Prosti tipovi podataka

- Celobrojni
  - int
  - short
  - long
  - byte
- Realni brojevi
  - float
  - double
- Znakovni
  - char
- Logički
  - boolean

30

## Prosti tipovi podataka

Prosti tip	Veličina	Minimum	Maximum
boolean	—	—	—
char	16-bit	Unicode 0	Unicode $2^{16}-1$
byte	8-bit	-128	+127
short	16-bit	$-2^{15}$	$+2^{15}-1$
int	32-bit	$-2^{31}$	$+2^{31}-1$
long	64-bit	$-2^{63}$	$+2^{63}-1$
float	32-bit	IEEE754	IEEE754
double	64-bit	IEEE754	IEEE754

31