

## Deklaracija promenljivih

- Deklaracija promenljive obuhvata:
  - dodelu simboličkog imena promenljivoj i
  - određivanje tipa promenljive
 (tip određuje koja će vrsta memorijskog registra biti rezervisana i koje će se operacije na toj promenljivoj odn. tom registru moći primeniti)
- Deklaracija promenljivih u Javi:
 

```
<tip promenljive> <naziv promenljive>
    int num1;
    <tip promenljive> <naziv promenljive1,
    naziv promenljive2,... >
    float a1,b1;
```

1

## Inicijalizacija promenljivih

- Inicijalizacija promenljive predstavlja dodelu vrednosti promenljivoj
  - vrednost mora biti usklađena sa tipom promenljive (inače kompajler javlja grešku!)
- Osim u sklopu njene deklaracije, moguće je i odvojeno inicijalizovati promenljivu (negde u programu)
- Opšti oblik inicijalizacije promenljive u Javi:
 

```
Npr.
    <tip promenljive> <naziv promenljive> = <vrednost>
```

tip promenljive	naziv promenljive	vrednost
boolean	signal	= false;

2

## Deklaracija i inicijalizacija promenljivih

### Primer:

```
double num2 = 3.1419;
int pom, index = 2;
double gama = 1.2, prosek;
boolean rezOk = false;
int index = 7;
double razlomak = 5/4;
double razlomak = 5.0/4.0;
float ratio = 5.8f;
```

- 5.8f - *float* vrednost (sa tačnošću od 7 decimala)
- 5.8 - *double* vrednost (sa tačnošću od 15 decimala)
- bez f (ili F), podrazumeva se *double* vrednost

3

## Deklaracija i inicijalizacija promenljivih

### Zadatak:

Koje su ispravne a koje pogrešne deklaracije?

```
int index = 1.2;
boolean ok = 1;
double z1 = 5 / 4;
float kolicnik = 5.8f;
double z1 = 5.0 / 4.0;
```

4

## Deklaracija i inicijalizacija promenljivih

```
int index = 1.2;           // greška
boolean ok = 1;          // greška
double z1 = 5 / 4;       // ispravno
float kolicnik = 5.8f;   // ispravno
double z1 = 5.0 / 4.0;   // ispravno
```

5

## Deklaracija i inicijalizacija promenljivih

- Ako se inicijalizuje promenljiva koja je već deklarirana, tada je opšti oblik inicijalizacije:

<ime promenljive> = <vrednost>

Primer:

```
boolean ok;
String ime;
int brojac_parnih;
.....
ok = true;
ime = "Ana";
brojac_parnih = 0;
```

6

## Inicijalizacija promenljivih prostog tipa

- Ako promenljivoj nije dodeljena vrednost pre njene upotrebe, kompajler javlja grešku
- Java inicijalno postavlja vrednosti prostih tipova promenljivih na **nulu**,  
odn. na **false** u slučaju promenljive tipa **boolean**
- Prosti tipovi u Javi su **jedini tipovi promenljivih koji nisu objekti**
- Dok se svi objekti u Java programu moraju kreirati uz pomoć operatora **new**,  
to nije potrebno za promenljive prostog tipa

7

## Imenovanje konstante

- U Javi konstanta se definiše pomoću ključne riječi **final**
  - time joj se dodeljuje ime koje će program učiniti nezavisnim od podataka

Opšti oblik instrukcije:

```
final <tip> <naziv konstante> = <vrednost>
```

- Primer:

```
final int X = 3;
final int MAX = 9;
final float PI = 3.14;
```

8

## Naredba dodele

- Opšti oblik naredbe:  
`<promenljiva> = <izraz>`
  - Izraz** je kombinacija promenljivih, konstanti, operatora i obliha zagrada, koji **ima vrednost**
  - Specijalni slučaj ove naredbe je  
`<promenljiva> = <konstanta>`
  - Treba razlikovati '=' kao operator dodele i  
 '==' kao relacioni operator!
- Primer: `x = 1; //promenljiva x dobija vrednost 1`  
`++a; a++; // a = a + 1`  
`b- -; - -b; // b = b - 1`  
`pom = a + b; // izračuna se zbir i dodeli u pom`  
`d*=5; // d = d * 5`

9

## Naredba dodele

Primer:

```
int x = 5;
int y = 2;
int z = 4;
z = z/y;
x = x*y + z;
y = x*(y + z); /* oble zagrade za promenu
                ugrađenog prioriteta operatora */
System.out.println(z); // 2
System.out.println(x); // 12
System.out.println(y); // 48
```

10

## Dodela vrednosti boolean promenljivoj

```
public class Demo {
    public static void main (String[] arg) {
        boolean b;
        b = (2 + 2 == 4);
        System.out.println(b);
    }
}
```

Pitanje:  
Šta će biti prikazano na izlazu?

11

## Dodela vrednosti boolean promenljivoj

Pitanje:  
Šta će biti prikazano na izlazu?  
  
(true)

12

## Aritmetički operatori

- + - sabiranje i predznak
- - oduzimanje i predznak
- \* - množenje
- / - deljenje (celobrojno i realno)
- % - ostatak pri deljenju  
(važi samo nad celobrojnim tipom)
- ++ - povećavanje operanda za 1
- - smanjenje operanda za 1

13

## Aritmetički operatori

Primer:

```

class ArithmeticDemo {
    public static void main (String[] args){
        int result = 1 + 2;
        System.out.println(result); // 3
        result = result - 1;
        System.out.println(result); // 2
        result = result * 2;
        System.out.println(result); // 4
        result = result / 2;
        System.out.println(result); // 2
        result = result + 8;
        System.out.println(result); // 10
        result = result % 7;
        System.out.println(result); // 3
    }
}

```

14

## Razlika između `i++` i `++i`

```

class PrePostDemo {
    public static void main(String[] args){
        int i = 3;
        i++;
        System.out.println(i); // "4"
        ++i;
        System.out.println(i); // "5"
        System.out.println(++i); // "6"
        System.out.println(i++); // "6"
        System.out.println(i); // "7"
    }
}

```

15

## Primer 1: Aktuelni datum

- Napisati program koji prikazuje **aktuelni datum**  
// program koristi klasu `Date` iz paketa `java.util` koja sadrži aktuelni datum

```

import java.util.*;

public class Datum{
    public static void main(String[] args) {
        System.out.println("Danas je: ");
        System.out.println(new Date());
    }
}

```

16

## Unos sa tastature

- `import java.util.Scanner;` // potrebna klasa `Scanner`
  - `Scanner ulaz = new Scanner(System.in);` // objekat `ulaz`
- Naredbe za unos različitih tipova promenljivih:
- `boolean bool = ulaz.nextBoolean();`  
// može se uneti samo *true* ili *false*
  - `String s = ulaz.next();` ili `ulaz.nextLine();`
  - `byte b = ulaz.nextByte();`
  - `int i = ulaz.nextInt();`
  - `short sh = ulaz.nextShort();`
  - `long l = ulaz.nextLong();`
  - `float f = ulaz.nextFloat();`
  - `double d = ulaz.nextDouble();`

17

## Primer 2: Unos sa tastature

```
import java.util.Scanner; // za unos potrebna klasa Scanner
public class Sabiranje {
    public static void main(String[] args) {
        int prviBroj;
        float drugiBroj;
        float suma;
        Scanner ulaz;

        // deklaracija promenljive ulaz koja je tipa Scanner, tj.
        // predstavlja objekat klase Scanner
        ulaz = new Scanner(System.in);

        // kreiranje objekta ulaz klase Scanner
        System.out.print("Unesite prvi, ceo broj: ");
        prviBroj = ulaz.nextInt();

        // metod nextInt() učitava ceo broj
        System.out.print("Unesite drugi, decimalni broj: ");
        drugiBroj = ulaz.nextFloat();

        // metod nextFloat() učitava realan broj tipa float
        suma = prviBroj + drugiBroj;
        System.out.println("Zbir je: " + suma);
    }
}
```

18

## Primeri literala za primitivne tipove

```
10           // decimalan broj
010          // oktalni zapis (cifre 0-7)
0x10        // heksadecimalni zapis (cifre 0-9, a-f, A-F)
0b10        // binarni zapis (cifre 0,1)
1e2         // eksponencijalni zapis
10l, 10L    // long
10.5f, 10.5F // float
10.5        // double (default format)
true, false // boolean
'a'         // char
'\n'       // char
'\u0061'   // unicode 'a'
```

19

## Operacije i tipovi podataka

- primeri aritmetičkih operacija (rezultat je "šireg" tipa)
 

<code>2 + 1.0</code>	// double	<code>2 + 1L</code>	// long
<code>2 + 1F</code>	// float	<code>2 + 1</code>	// int
- primeri logičkih i operacija poređenja (rezultat je tipa boolean)
 

<code>boolean test1 = 2 &lt; 5;</code>	// true
<code>boolean test2 = false &amp;&amp; true;</code>	// false
- primeri operacija nad bitovima (nad logičkim i nad celobrojnim vrednostima)
 

<code>int test3 = 1 ^ 2;</code>	// 3
<code>boolean test4 = false &amp; true;</code>	// false

20

## Konverzija promenljivih

- **Implicitna konverzija** (*implicit cast*)

bez gubitka - sa 'užeg' na 'širi' tip podatka - (*widening*)  
 byte → short → int → long → float → double

uz gubitke - sa 'šireg' na 'uži' tip podatka - (*narrowing*)

- **Explicitna konverzija** (*explicit cast*)

uz pomoć operatora casting-a :  
 opšti oblik:

(ciljni tip) ime\_promenljive

21

## Primeri gubitka pri konverziji

```
double d = 1;
float f = (float) d;
long l = (long) f;
int i = (int) l;
short s = (short) i;
byte b = (byte) s;
int v1 = 1 000 000;
int v2 = (short) v1;    // 16960
```

22