



Internet programiranje

predavanje 13

Prof. dr Miroslav Lutovac
mlutovac@viser.edu.rs



Java programski jezik

- Java – izgovara se **java**, **džava**, 'dʒɑ:və
- **objektno-orijentisani** programski jezik
opšte namene **GNU General Public License**
- koncept **klasa** iz objektno-orijentisane paradigme
- sintaksa slična C i C++, stroži pri prevođenju
- **nezavisan od platforme** **Besplatna podrška Oracle**
- pojednostavljeno upravljanje memorijom
- najpopularniji jezik za **klijent-server veb aplikacije**
- u izradi **mobilnih** aplikacija za Android sisteme



Java programski jezik

- **jednostavan** - da se lako programira, brzo se uči
- **objektno orijentisan** - objekti i interfejs ka objektima
- **distribuiran** - biblioteke rutina za TCP/IP, HTTP, FTP, aplikacije da pristupaju objektima preko mreže i URL kao da pristupaju lokalnom sistemu datoteka
- **robustan** – pouzdan, proveriti mogućih problema, dinamičke proveriti tokom izvršavanja, eliminacija mogućih grešaka
- **bezbedan** - u mrežnim okruženjima, zaštita od virusa, zlonamerne modifikacije, da onemogućiti napade
prekoračenje izvršnog steka, pristup memoriji izvan dela dodeljenog procesu, čitanje ili upisivanje datoteka bez dozvole



Java programski jezik

- **neutralan** - kompajler stvara objektnu datoteku
 - nezavisno od operativnog sistema
 - kompajlirani kod se može izvršavati na raznim procesorima
 - generiše bajtkod instrukcije nezavisno od arhitekture
 - lako se interpretira, lako se prevodi u mašinski kod
- **prenosiv** - ne zavisi od implementacije
 - primitivni tipovi i ponašanje u aritmetici su fiksni
 - biblioteke definišu prenosive interfejse
- **interpretiran** - Java bajtkod se izvršava na svakom računaru sa Java interpreterom
 - linkovanje je lakše, razvoj je brži



Java programski jezik

- **performantan** - kada su potrebne bolje performanse, bajtkod može da bude preveden tokom izvršavanja u mašinski jezik
- **višenitan** - bolji interaktivni odzivi i rad u realnom vremenu
- **dinamičan** – za okruženja koje se stalno menja
 - biblioteke mogu da dodaju nove metode i polja, bez uticaja na klijente
 - jednostavno se pronalaze informacije pri izvršavanju



Java programski jezik

- Prvobitno namenjen za programiranje kućnih elektronskih uređaja (televizor, rekorder, aparat za kafu, bio je previše napredan kada je razvijen, sada i za konzole z aigru, superkompjutere, laptopovi)
- Java Coffee, logo je čaša vruće kafe
- Java platforma, Java okruženje u kome se program izvršava, da je što manje zavisn od karakteristika računarskog sistema

Java SE 9, septembar 2017



Java programski jezik

- izvršava se korišćenjem
Java virtuelne mašine
(JVM - Java Virtual Machine)
apstraktna mašina koji postoji samo u memoriji
- Java je jezik koji se prevodi i interpretira
- Java izvorni kod (*.java **fajl**)
- nakon kompajliranja kompajlerom **javac**, prevodi se u
izvorni kod bajtkod (*.class fajl)
- bajtkod je isti za sve vrste računara
- bajtkod predstavlja instrukcije za JVM



Java programski jezik

- **izvorni** java kod se dekoduje i interpretira u **izvršni** mašinski kod
- izvršavanje na računaru, hardveru
- radi u bilo kojoj kombinaciji hardvera i operativnog sistema
- od istog izvornog koda, različiti izvršni kodovi
- za **različite platforme**
- *Write once, run everywhere (WORA)*



Java programski jezik

- Prenosivost izaziva gubitak performansi
- tek prilikom interpretiranja bajtkod se prevodi u mašinski za konkretnu platformu
- problem, isti kod se više puta interpretira na istoj platformi
- Just-in-time, JIT, kompajler, prevodi u mašinski kod za konkretnu platformu na kojoj se koristi
- Krajnji korisnici koriste Java Runtime Environment, JRE, instaliran na sopstvenoj mašini za nezavisan rad Java aplikacije ili u veb pregledaču za Java applete



Java programski jezik

- Standardne biblioteke obezbeđuju generički postupak da se pristupi host-specifičnim karakteristikama kao što su grafika ili umrežavanje
- Zbog overhead dela da se bajtkod interpretira u mašinski jezika, program se sporije izvršava
- Neke platforme nude direktnu podršku za hardver, npr
za mikrokontrolere umesto Java virtual machine,
za Advanced RISC (reduced instruction set computing) Machine



Java programski jezik

- **garbage collector**
- uklanjanje nepotrebnih objekata iz memorije
- radi nezavisno od pokrenutih programa i samostalno odlučuje koje objekte se uklanjaju
- vrši defragmentaciju memorije
- ne mora se voditi računa o destruktorima, automatski



Java programski jezik

- 2 distribucije
- **Java Runtime Environment**, JRE, za **korisnike**, sadrži deo Java SE platforme
- **Java Development Kit**, JDK, za softver **developere** i sadrži razvojne alate kao što su Java compiler, Javadoc, Jar, debugger



Java sintaksa

- Sve se koduje unutar klasa
- Svaki podatak je objekt
izuzetak su primitivni tipovi podataka
celobrojni, brojevi sa pokretnom tačkom, logički tip
vrednosti (boolean), karakteri
- Java koristi metode kao na primer printf()

{

}



Java identifikator

- služi za identifikovanje neke konstrukcije ili entiteta (promenljivih, klasa, metoda)
- Počinje slovom, znakovima \$ i _ a zatim mogu da budu slova i cifre
- Razlikuju se velika i mala slova true, false, null
- Rezervisane reči su identifikatori koji imaju specijalnu namenu i ne mogu se koristiti za imenovanje drugih:
abstract, boolean, break, byte, case, catch, char, class, const, continue, default, do, double, else, extends, final, finally, float, for, goto, if, implements, import, instanceof, int, interface, long, native, new, package, private, protected, public, return, short, static, strictfp, super, switch, this, throw, synchronized, throws, transient, try, void, volatile, while



Java literali

- Literali su konstante primitivnog tipa ili konkretan primerak klase String
- Celobrojni literali mogu biti zapisani kao: dekadni, oktalni ili heksadekadni, binarni
- Realni literali su konstante sa pokretnom tačkom
- Sadrži bar jednu cifru, decimalu, eksponent
- Realni literali: float i double
- Literal tipa float, na kraju se navodi slovo f ili F
- Double je podrazumevani tip za realni literal, ne mora se navesti slovo d ili D



Java literali

- Logički literali su vrednosti true (istinito) i false (neistinito), "da" i "ne", "istinito" i "neistinito"
- Znakovni literal je bilo koji znak, osim apostrofa, obrnute kose crte, i eskejp sekvenca
 - '\" - apostrof
 - '\"' - navodnik
 - '\\' - obrnuta kosa crta
 - '\r' - znak za povratak na početak reda
 - '\n' - znak za prelazak u novi red
 - '\f' - znak za prelazak na novu stranu
 - '\t' - znak tabulatora
 - '\b' - znak za povratak na jedno mesto unazad



Java separatori

specijalni znaci da odvoje određene logičke celine u kodu
(), **obične zagrade**, za odvajanje liste parametara od poziva metode, za naglašavanje prioriteta

{}, **vitičaste zagrade**, za ograđivanje vrednosti automatski inicijalizovanih nizova, za definisanje blokova

[], **uglaste zagrade**, za izdvajanje vrednosti članova nizova, za deklarisanje samih nizova

; **tačka zarez**, zaključuje naredbu

, **zarez**, za razdvajanje identifikatore u deklaraciji promenljive, za povezivanje naredbi unutar petlji

. **tačka**, za razdvajanje naziva paketa od potpaketa i klasa, za razdvajanje promenljivih ili metoda od Identifikatora

bin

FileHomeShareView

←→↕↑

This PC > Windows8_OS (C:) > Users > MIROSLAV > workspace2 > obuka01 > bin >

Documents

Pictures

aAnita

Internet program

merni informacie

Name

obuka01

Date modified

12/11/2017 9:09 A...

Type

File folder

1 item

obuka01

FileHomeShareView

←→↕↑

This PC > Windows8_OS (C:) > Users > MIROSLAV > workspace2 > obuka01 > bin > obuka01 >

Downloads

Music

Pictures

Videos

Windows8_OS (C:)

Name

ZdravoLM

Date modified

12/11/2017 9:14 A...

Type

CLASS File

Size

1 KB

1 item

File Explorer window showing the directory structure: This PC > Windows8_OS (C:) > Users > MIROSLAV > workspace2 > obuka01 > src > obuka01.

The left sidebar shows the navigation pane with the following items: Downloads, Music, Pictures, Videos, and Windows8_OS (C:).

The main pane displays a table of files:

Name	Date modified	Type	Size
ZdravoLM	12/11/2017 9:14 A...	JAVA File	1 KB

1 item

Notepad++ window showing the source code for ZdravoLM.java:

```
1 package obuka01;
2
3 import java.util.*;
4 public class ZdravoLM {
5     public static void main(String[] args) {
6         System.out.println("Zdravo, danas je: ");
7         System.out.println(new Date());
8     }
9 }
10
```

Java source file | length : 204 | lines : 10 | Ln : 1 | Col : 1 | Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

← → ↕ ↑ > This PC > Windows8_OS (C:) > Users > MIROSLAV > workspace2 > obuka01 > bin > obuka01

Downloads

Music

Pictures

Videos

Windows8_OS (C:)

Name

Date modified

Type

Size

ZdravoLM

12/11/2017 9:14 A...

CLASS File

1 KB

1 item

C:\Users\MIROSLAV\workspace2\obuka01\bin\obuka01\ZdravoLM.class - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

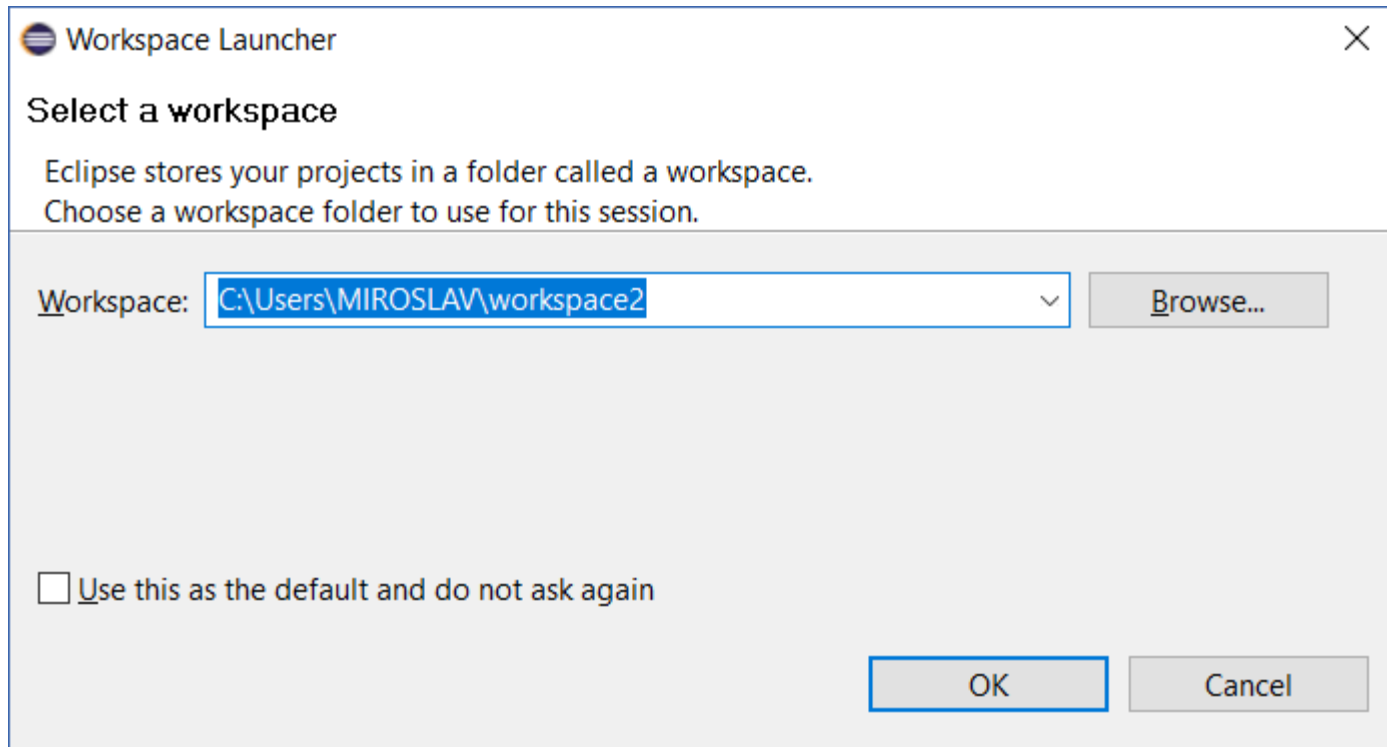
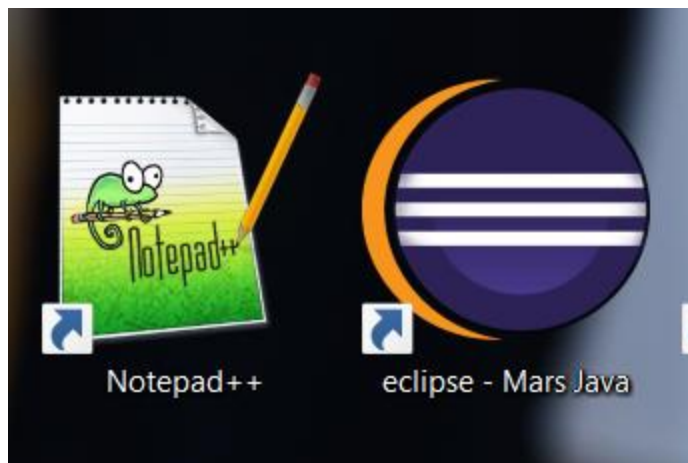


ZdravoLM.java ZdravoLM.class

```

1  Êp°¾NULNULNUL4NUL (BELNULSTXSOHNULDLEobuka01/ZdravoLM
2  NULETXNUL FENULENONULACKSOHNULSILineNumberTableSOH
3  NULEMNULESCBELNULSUBSOHNULDC3java/io/PrintStreamFFN
4  NULRSNUL
5  NULEMNUL"FFNULEFSNUL#SOHNULNAK (Ljava/lang/Object;)VS
6  SourceFileSOHNUL
7  ZdravoLM.javaNUL!NULSOHNULETXNULNULNULNULNULSTXNULS
8  NULNULNULACKNULSOHNULNULNULEOTNULVTNULNULNULFFNULS
9  NULNULNUL NULSONULSINULSOHNULBELNULNULNULHNULETXN
10 NULNULNULSONULETXNULNULNULACKNULBSNULBELNULNAKNULB

```



Java - obuka01/src/obuka01/ZdravoLM.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access Java

Package Explorer

- ip1
- obuka01

Klasa.java ZdravoLM.java

```
1 package obuka01;
2
3 import java.util.*;
4 public class ZdravoLM {
5     public static void main(String[] args) {
6         System.out.println("Zdravo, danas je: ");
7         System.out.println(new Date());
8     }
9 }
10
```

Task List

Find All Activate...

Connect Mylyn

Connect to your task and ALM tools or create a local task.

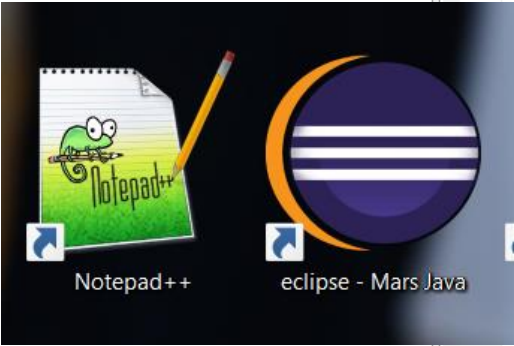
Outline

- obuka01
 - ZdravoLM
 - main(String[]): void

Problems Javadoc Declaration Console

No consoles to display at this time.

Writable Smart Insert 4:1



Notepad++

eclipse - Mars Java

Java - obuka01/src/obuka01/ZdravoLM.java - E

File Edit Source Refactor Navigate Search



Package Explorer

> ip1
> obuka01

Outline

obuka01
ZdravoLM
main(String[]) : void

Problems Javadoc Declaration Console

<terminated> ZdravoLM [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 11, 2017, 1:03:00 PM)

Zdravo, danas je:

Mon Dec 11 13:03:01 CET 2017

Writable

Smart Insert

4 : 1

Problems Javadoc Declaration Console

<terminated> ZdravoLM [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 11, 2017, 1:03:00 PM)

Zdravo, danas je:

Mon Dec 11 13:03:01 CET 2017

Writable

Smart Insert

4 : 1

```
C:\Users\MIROSLAV\workspace2\obuka01\src\obuka01\ZdravoLM.java - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
ZdravoLM.java x

1 package obuka01;
2
3 import java.util.*;
4 public class ZdravoLM {
5     public static void main(String[] args) {
6         System.out.println("Zdravo, danas je: ");
7         System.out.println(new Date());
8     }
9 }
10
```

Java source file length : 204 lines : 10

Problems @ Javadoc Declaration Cor

<terminated> ZdravoLM [Java Application] C:\P

Zdravo, danas je:

Mon Dec 11 13:03:01 CET 2017

```
C:\Users\MIROSLAV\workspace2\obuka01\bin\obuka01\ZdravoLM.class - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
ZdravoLM.java x ZdravoLM.class x

1 Êp°¾NULNULNUL4NUL (BELNULSTXSOHNULDLEobuka01/ZdravoLM
2 NULETXNUL FFNULENONULACKSOHNULSTLineNumberTableSOH
3 NULEMNULEESCBEINULSUBSOHNULDC3java/io/PrintStreamFFN
4 NULRSNUL
5 NULEMNUL" FFNULEFSNUL# SOHNULNAK (Ljava/lang/Object;)VS
6 SourceFileSOHNUL
7 ZdravoLM.javaNUL! NULSOHNULETXNULNULNULNULSTXNULS
8 NULNULNULACKNULSOHNULNULNULEOTNULVTNULNULNULEFNULS
9 NULNULNUL NULSONULSINULSOHNULBELNULNULNULHNULETXN
10 NULNULNULSONULETXNULNULNULACKNULBSNULBELNULNAKNULB

Normal text file length : 626 lines : 10 Ln : 1 Col : 1 Sel : 0 | 0 Unix (LF) ANS
```



Java operatori i operandi

- **Operatori** omogućavaju operacije nad podacima
- **operandi** su podaci na koje se primenjuju operatori
- prefiksni, infiksni i postikfni operatori
- operatori dodele i ostali
- aritmetički operatori, sa operandima i separatorima služe za formiranje aritmetičkih izraza
- aritmetičke operatori su: + - * / % ++ --
- relacioni operatori su: == != < > >= <=

uslovni operator

`<logički_operator>?<prvi_izraz>:<drugi_izraz>`

instancnog operatora, da li pripada klasi

`<instancni_operator>::= instanceof`



Java operatori i operandi

- Operatori na nivou bita su logički ili operator pomeranja
 - & bitovna konjukcija(AND)
 - ~ bitovna negacija(NOT)
 - | bitovna disjunkcija(OR)
 - ^ bitovna ekskluzivna disjunkcija(XOR)
 - << pomeranje (šiftovanje) ulevo
 - >> pomeranje (šiftovanje) udesno
 - >>> pomeranje (šiftovanje) udesno sa nulama
 - && konjukcija
 - || disjunkcija
 - ! negacija



Java komentari

- da se objasne pojedina mesta u programu
- **jednoredni** komentari `//` i završavaju se na kraju reda
- **višeredni** komentari `/*` i završavaju se sa `*/`
mogu se prostirati preko više redova, ne ugneždeni
- **za dokumentaciju** `/**` i završavaju se sa `*/`
mogu se prostirati preko više redova, ne ugneždeni
Javadoc je izvršni za kreiranje dokumentacije programa i može biti učitán iz Integrated Development Environments, IDEs, kao na primer Eclipse da omogući softverašima pristup dokumentaciji u okviru IDE
- **Belina** nema grafički prikaz na izlaznom uređaju za međusobno razdvajanje elementarnih konstrukcija



Java komentari

```
// This is an example of a single line comment using two slashes
```

```
/* This is an example of a multiple line comment using the slash and asterisk.
```

```
This type of comment can be used to hold a lot of information or deactivate  
code, but it is very important to remember to close the comment. */
```

```
package fibsandlies;
```

```
import java.util.HashMap;
```

```
/**
```

```
 * This is an example of a Javadoc comment;
```

```
 * Javadoc can compile documentation
```

```
 * from this text. Javadoc comments must immediately precede
```

```
 * the class, method, or field being documented.
```

```
 */
```



Java komentari

```
public class FibCalculator extends Fibonacci implements Calculator {  
    private static Map<Integer, Integer> memoized = new HashMap<Integer, Integer>();  
  
    /*  
    * The main method written as follows is used by the JVM as a starting point for the program.  
    */  
  
    public static void main(String[] args) {  
        memoized.put(1, 1);  
        memoized.put(2, 1);  
        System.out.println(fibonacci(12)); //Get the 12th Fibonacci number and print to console  
    }  
  
    /**  
    * An example of a method written in Java, wrapped in a class.  
    * Given a non-negative number FIBINDEX, returns  
    * the Nth Fibonacci number, where N equals FIBINDEX.  
    * @param fibIndex The index of the Fibonacci number  
    * @return The Fibonacci number  
    */  
}
```



Java klasa, objekat, atribut

- skup objekata sa zajedničkim svojstvima, koji se ponašaju na isti način
- sastoji se od atributa i metoda
- atributi opisuju osobine, a metode ponašanje klase
- paket je grupa sličnih tipova klasa, interfejsa i pod-paketa
- objekat je primerak klase
- svaki objekat ima svoj primerak svih polja atributa, koji je nezavisan od ostalih objekata, a metode su zajedničke
- atributima objekta se pristupa pomoću operatora tačka, Pera.ime
- metode se pozivaju na sledeći način: Pera.jedi()



Java klasa, objekat, atribut

- Izvorni fajl ima isto ime kao klasa a ekstenzija fajla je .java
- Nakon kompajliranja, dobija se bytecode i novi fajl sa ekstenzijom .class
- Tek tada može da se izvrši program, tzv launched
- Java izvorni kod može da sadrži jednu javnu (public) klasu
- Može da sadrži višestruke klase sa drugim u odnosu na public klasu, i prizvoljan broj ugneždenih klasa
- Klasa koja nije deklarisan kao public, može biti u fajlu sa ekstenzijom .java
- Kompajler će generisati class fajl za svaku klasu definisanu u izvornom fajlu



Java klasa – ključne reči

- Ključna reč **public** označava se metod može pozvati iz koda druge klase, ili da se klasa može koristiti od klase izvan ove klase
- Ključna reč **static** ispred metode ukazuje na static metod, koji je pridružen samo sa klasom i ni sa bilo kojom drugom instancom klase
- Samo static metoda može biti pozvana bez referenciranja na objekat
- Static metoda ne može da pristupi bilo kom članu klase koji nije takođe static
- Metode koje nisu projektovane da su static, su slučajne

```
4 public class ZdravoLM {  
5     public static void main(String[] args) {  
6         System.out.println("Zdravo, danas je: ");  
7         System.out.println(new Date());  
8     }  
9 }  
10
```

Java source file length : 204 lines : 10 Ln : 1 Col : 1 VISEB Windows (CR LF) UTF-8 INS³²



- Ključna reč **void** označava ukazuje da main metoda ne vraća bilo koju vrednost onome koje pozvao klasu
- Ako Java program treba da izađe sa kodom greškom, mora da se pozove eksplicitno sa **System.exit()**
- Ime metode **main** NIJE ključna reč u Javi
- To je ime metode koju poziva Java da prođe kontrolu programa
- Java klase koje se koriste kao apleti i JavaBeans, ne koriste i ne treba im main() metoda
- Java program može da sadrži višestruke klase, što znači da JVMu treba eksplicitno reći koju klasu treba da pokrene odakle

```
4 public class ZdravoLM {  
5     public static void main(String[] args) {  
6         System.out.println("Zdravo, danas je: ");  
7         System.out.println(new Date());  
8     }  
9 }  
10
```

Java source file length : 204 lines : 10 Ln : 1 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 INS²³



- Metod main prihvata niz kao **String** objekat
- Po konvenciji, referencira se na args, iako bi bilo koje legalno ime identifikatora moglo da se koristi
- Postoji alternativna sintaksa za args
- Pokretač Jave pokreće Javu učitavanjem klase, specificirane u komandnoj liniji ili kao atribut JARa (Java Archive, format paketa fajlova, da bi se agregirali brojni java class fajlovi u jedan fajl za distribuciju)
- Startuje se public static void main(String[]) metoda
- Zasebni programi moraju eksplicitno da deklarišu ovu metodu

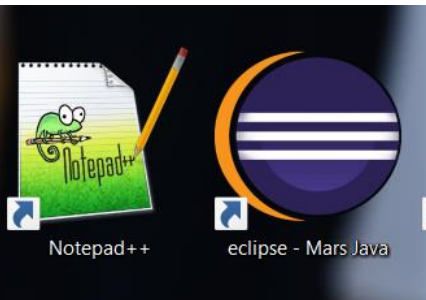
```
4 public class ZdravoLM {  
5     public static void main(String[] args) {  
6         System.out.println("Zdravo, danas je: ");  
7         System.out.println(new Date());  
8     }  
9 }  
10
```

Java source file length : 204 lines : 10 Ln : 1 Col : 1 VISER Sel : 0 | 0 Windows (CR LF) UTF-8 INS²⁴

- Parametar argumenta `String[] args` metoda je niz tipa `String` objekta koji sadrži bilo koje argumente koji se prosleđuju klasi
- Parametri koji se prosleđuju ka `main` se obično prosleđuju preko komandne linije
- `Printing` (ispisivanje) je deo standardne biblioteke
- Klasa `System` definiše `public static` polje koje se naziva `out`
- Objekat `out` je instanca `PrintStream` klase i obezbeđuje brojne metode za ispisivanje podataka na standardni izlaz, uključujući `println(String)`, koji pridružuje novu liniju stringu koji je prosleđen
- Kompajler automatski konvertuje string u string objekat

```
3  import java.util.*;
4  public class ZdravoLM {
5      public static void main(String[] args) {
6          System.out.println("Zdravo, danas je: ");
7          System.out.println(new Date());
8      }
9  }
10
```

Java source file length : 204 lines : 10 Ln : 1 Col : 1 VISEP Set: 0 | 0 Windows (CR LF) UTF-8 INS 35



Startuje se Eclipse

Java - obuka01/src/obuka01/ZdravoLM.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access Java

Package Explorer

- ip1
- obuka01

Klasa.java ZdravoLM.java

```
1 package obuka01;
2
3 import java.util.*;
4 public class ZdravoLM {
5     public static void main(String[] args) {
6         System.out.println("Zdravo, danas je: ");
7         System.out.println(new Date());
8     }
9 }
10
```

Task List

Find All Activate...

Connect Mylyn

Connect to your task and ALM tools or create a local task.

Outline

- obuka01
 - ZdravoLM
 - main(String[]) : void

Problems @ Javadoc Declaration Console

No consoles to display at this time.

VISER

Writable Smart Insert 4 : 1

36



Startuje se novi paket



 New Java Package

Java Package

Create a new Java package.

Creates folders corresponding to packages.

Source folder:

Name:

☐ Create package-info.java





New Java Class

Java Class

⚠ Type name is discouraged. By convention, Java type names usually start with an uppercase letter

Source folder: obuka01/src Browse...

Package: obuka02 Browse...

☐ Enclosing type: Browse...

Name: obuka0201

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

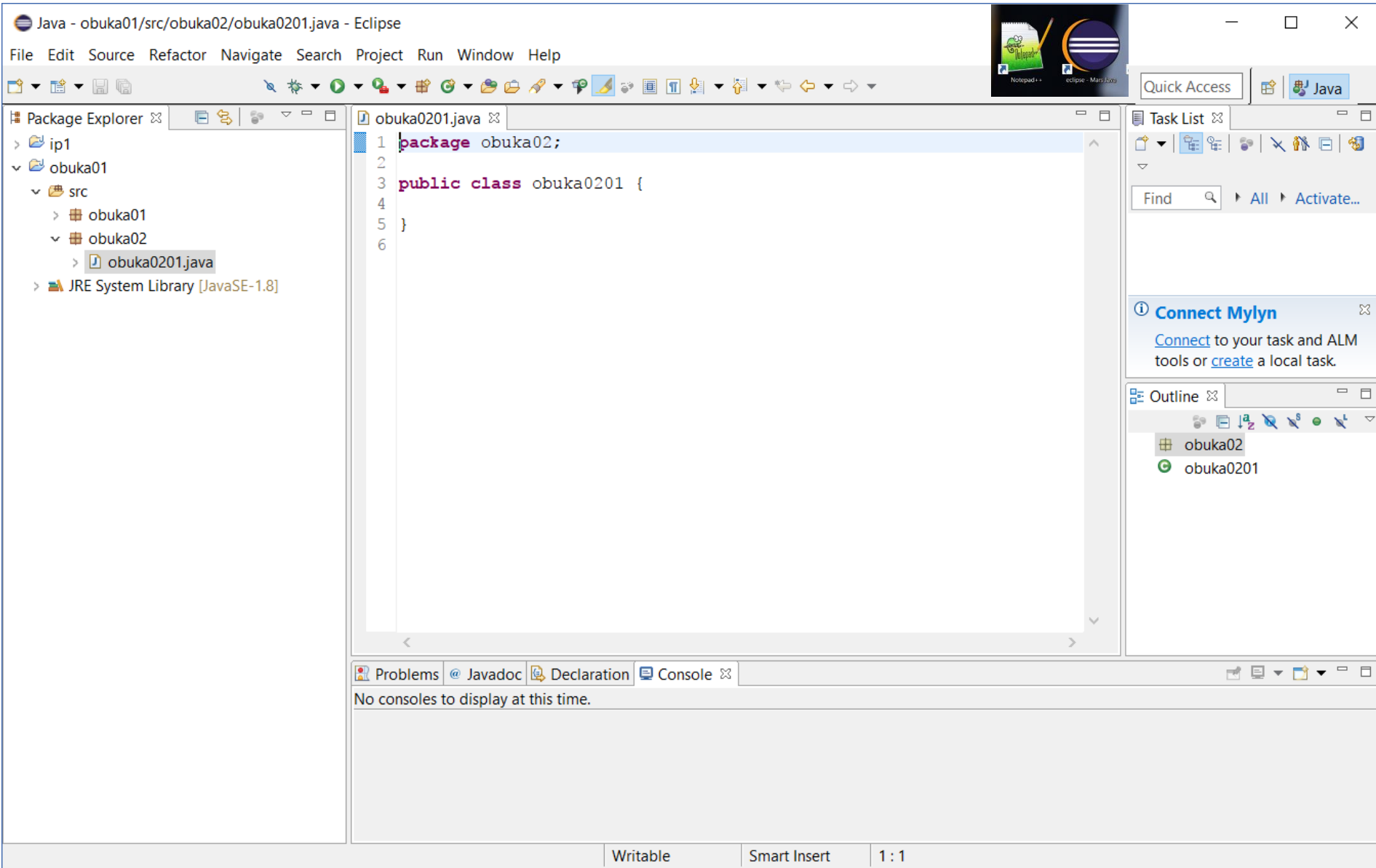
Which method stubs would you like to create?

☐ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

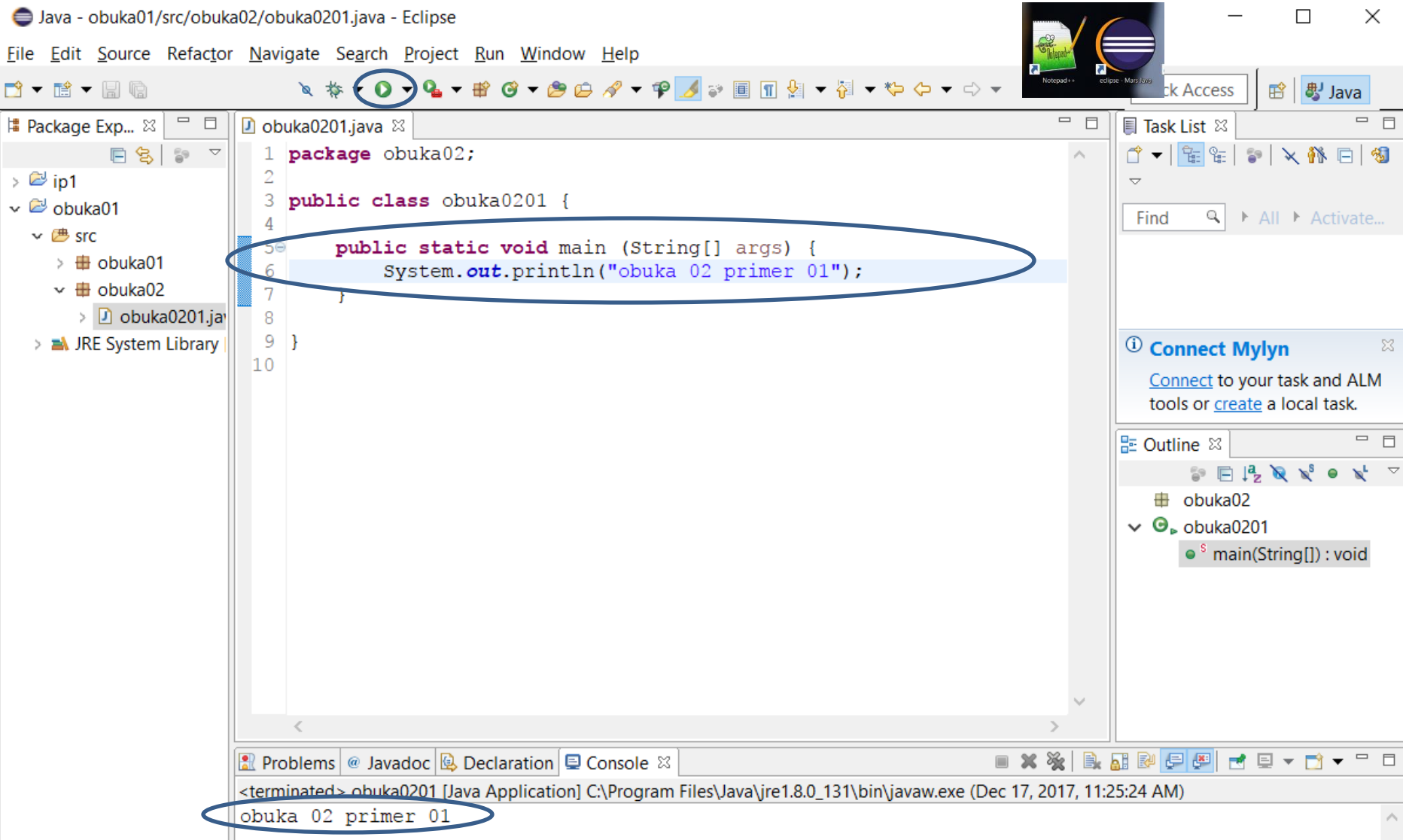
Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

pravi se nova klasa

Finish Cancel



Pripremljen uzorak za klasu



1. unese se tekst, 2. izvrši klasa

```
1 package obuka02;  
2  
3 public class obuka0201 {  
4  
5     public static void main (String[] args) {  
6         System.out.println("obuka 02 primer 01");  
7     }  
8  
9 }  
10
```

public klasa se može koristiti od drugih klasa
static je static metod, pridružen samo sa ovom klasom
void ukazuje da ne vraća vrednost onome ko je pozvao klasu
main je ime metode
metoda main prihvata niz kao **String** objekat
Parametar **args** je niz tipa String objekta koji sadrži
bilo koje argumente koji se prosleđuju klasi
ispisivanje u **out** (sistemski izlaz) je deo standardne biblioteke



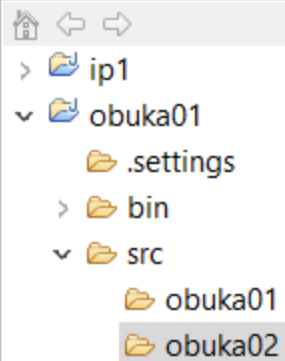
Save As

Save As

Save file to another location.

Enter or select the parent folder:

obuka01/src/obuka02



Nova klasa kopiranjem
postojeće klase

File name: obuka0202.java



OK

Cancel

Java - obuka01/src/obuka02/obuka0202.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- ip1
- obuka01
 - src
 - obuka01
 - obuka02
 - obuka0201.java
 - obuka0202.java
- JRE System Library [JavaSE-1.8]

obuka0202.java

```
1 package obuka02;  
2  
3 import java.util.*;  
4 public class obuka0202 {  
5  
6     public static void main (String[] args) {  
7         System.out.println("obuka 02 primer 02");  
8         System.out.println(new Date());  
9     }  
10 }  
11  
12 }  
13
```

Task List

Find All Activate...

Connect Mylyn

Connect to your task and ALM tools or create a local task.

Outline

- obuka02
 - obuka0202
 - main(String[]) : void

Problems @ Javadoc Declaration Console

<terminated> obuka0202 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 17, 2017, 12:40:08 PM)

obuka 02 primer 02

Sun Dec 17 12:40:09 CET 2017

Dodaje se new ... iz java.util.*

Java - obuka01/src/obuka02/obuka0203.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Java

Package ...

ip1

obuka01

src

obuka01

obuka02

obuka0201

obuka0202

obuka0203

JRE System Libra

obuka0203.java

```
1 package obuka02;
2 import java.util.*;
3 public class obuka0203 {
4     public static void main (String[] args) {
5         System.out.println("obuka 02 primer 03");
6         System.out.println("primitivne vrednosti\n");
7         System.out.println("Celobrojni tip podataka\n'int a = 2;');
8         int a = 2;
9         System.out.println("celobrojna a = " +a +"\n");
10        System.out.println("Racionalni tip podataka\n'double b = 3.14;');
11        double b = 3.14;
12        System.out.println("racionalni b = " +b);
13        b = a;
14        System.out.println("konverzija int u double 'b = a;' b = " +b);
15    }
```

Task List

Connect Mylyn

Connect to your task and ALM tools or create a local task.

Outline

obuka02

obuka0203

main(String[]): vo

Problems

Javadoc

Declaration

Console

<terminated> obuka0203 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 17, 2017, 1:34:56 PM)

obuka 02 primer 03

primitivne vrednosti

Celobrojni tip podataka

'int a = 2;'

celobrojna a = 2

Racionalni tip podataka

'double b = 3.14;'

racionalni b = 3.14

konverzija int u double 'b = a;' b = 2.0

int a = 2;

double b = 3.14;

b = a;

Primitivne celobrojne i racionalne vrednosti



Quick Access



Task List



Find

All Activ...

Connect Mylyn

Connect to your task and ALM tools or create a local task.

Outline



obuka02

obuka0204

main(String[]) : void

obuka0204.java

```
1 package obuka02;
2 import java.util.*;
3 public class obuka0204 {
4     public static void main (String[] args) {
5         System.out.println("obuka 02 primer 04");
6         System.out.println("primitivne vrednosti");
7         double a = 1.23456;
8         System.out.println("\n'a = 1.23456;");
9         System.out.println(" a = " + a);
10        double b = -3.21e-45; // -3.21E-45
11        System.out.println("\n'double b = -3.21e-45;");
12        System.out.println(" b = " + b);
13        double c = 1d/3; // 0.3333333333333333
14        System.out.println("\n'double c = 1d/3;");
15        System.out.println(" c = " + c);
16    }
17 }
18
```

Problems @ Javadoc Declaration Console

<terminated> obuka0204 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 17, 2017, 2:01:59 PM)

obuka 02 primer 04
primitivne vrednosti

'a = 1.23456;'
a = 1.23456

'double b = -3.21e-45;'
b = -3.21E-45

'double c = 1d/3;'
c = 0.3333333333333333

```
double a = 1.23456;
double b = -3.21e-45;
double c = 1d/3;
```

Primitivne racionalne vrednosti

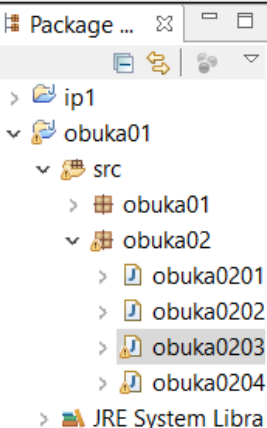
Writable

Smart Insert

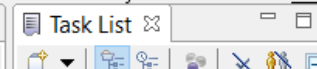
13 : 47



Quick Access

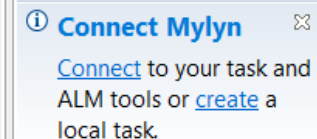


```
1 package obuka02;
2 import java.util.*;
3 public class obuka0204 {
4     public static void main (String[] args) {
5         System.out.println("obuka 02 primer 04");
6         System.out.println("primitivne vrednosti");
7         double a = 1.23456;
8         System.out.println("\n'a = 1.23456;");
9         System.out.println(" a = " + a);
10        double b = -3.21e-45; // -3.21E-45
11        System.out.println("\n'double b = -3.21e-45;");
12        System.out.println(" b = " + b);
13        double c = 1d/3; // 0.3333333333333333
14        System.out.println("\n'double c = 1d/3;");
15        System.out.println(" c = " + c);
16    }
17 }
18
```

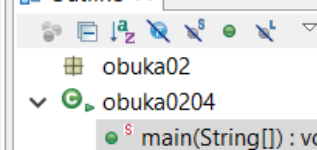


Find

All Activ...



Outline



Problems @ Javadoc Declaration Console

<terminated> obuka0204 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 17, 2017, 2:01:59 PM)

obuka 02 primer 04
primitivne vrednosti'a = 1.23456;'
a = 1.23456'double b = -3.21e-45;'
b = -3.21E-45'double c = 1d/3;'
c = 0.3333333333333333

```
double a = 1.23456;
double b = -3.21e-45;
double c = 1d/3;
```

Primitivne racionalne vrednosti

Writable

Smart Insert

13 : 47

Primitivne racionalne vrednosti

Java - obuka01/src/obuka02/obuka0205.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help



Quick Access

Java

Package ...
ip1
obuka01
src
obuka01
obuka02
obuka0201
obuka0202
obuka0203
obuka0204
obuka0205
JRE System Libra

```
obuka0205.java
1 package obuka02;
2 import java.util.*;
3 public class obuka0205 {
4     public static void main (String[] args) {
5         System.out.println("obuka 02 primer 05");
6         System.out.println("primitivne vrednosti");
7         double a = 1/3;
8         System.out.println("a = 1/3; a = " + a);
9         double d = 1d/3;
10        System.out.println("d = 1d/3; d = " + d);
11        float b = 3.14159365358979323846f;
12        System.out.println("b = 3.14159365358979323846f; b = " + b);
13        double c = 3.14159365358979323846; //precizniji od float
14        System.out.println("c = 3.14159365358979323846; c = " + c);
15    }
16 }
17
```

Connect Mylyn
Connect to your task and ALM tools or create a local task.

Outline
obuka02
obuka0205
main(String[]) : vo

Problems Javadoc Declaration Console

<terminated> obuka0205 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 17, 2017, 2:18:16 PM)

```
obuka 02 primer 05
primitivne vrednosti
a = 1/3; a = 0.0
d = 1d/3; d = 0.3333333333333333
b = 3.14159365358979323846f; b = 3.1415937
c = 3.14159365358979323846; c = 3.1415936535897933
```

```
double a = 1/3;
double d = 1d/3;
float b = 3.14159365358979323846f;
double c = 3.14159365358979323846;
//precizniji od float
```

byte 8-bit, -128 do 127
short 16-bit, -2^{15} do $(2^{15}-1)$
int 32-bit, -2^{31} do $(2^{31}-1)$
long 64-bit, -2^{63} do $(2^{63}-1)$

The screenshot shows the Eclipse IDE with a Java project named 'obuka01'. The source code file 'obuka0206.java' is open, displaying a program that prints the values of various primitive data types. The code is as follows:

```
1 package obuka02;
2 import java.util.*;
3 public class obuka0206 {
4     public static void main (String[] args) {
5         System.out.println("obuka 02 primer 06");
6         System.out.println("primitivne vrednosti\n");
7         int a = 100;
8         System.out.println("'int a = 100;'  a = " + a);
9         byte b = (byte) a;
10        System.out.println("'byte b = (byte) a;'  b = " + b);
11        byte c = 0x55;
12        System.out.println("'byte c = 0x55;'  c = " + c);
13        short d = 0x5a6b;
14        System.out.println("'short d = 0x5a6b;'  d = " + d);
15        long v = -1234567890;
16        System.out.println("'long v = -1234567890;'  v = " + v);
17    }
18 }
19
```

The 'Console' view at the bottom shows the output of the program:

```
<terminated> obuka0206 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin
obuka 02 primer 06
primitivne vrednosti

'int a = 100;'  a = 100
'byte b = (byte) a;'  b = 100
'byte c = 0x55;'  c = 85
'short d = 0x5a6b;'  d = 23147
'long v = -1234567890;'  v = -1234567890
```

int a = 100;
byte b = (byte) a;
byte c = 0x55;
short d = 0x5a6b;
long v = -1234567890;

Primitivne celobrojne vrednosti



Boolean 1 bit String

```

1 package obuka02;
2 import java.util.*;
3 public class obuka0207 {
4     public static void main (String[] args) {
5         System.out.println("obuka 02 primer 07");
6         System.out.println("primitivne vrednosti\n");
7         String tekst1 = "Zdravo ";
8         System.out.println("tekst1 =" + tekst1);
9         String tekst2 = "svima!";
10        System.out.println("tekst2 =" + tekst2);
11        System.out.println("tekst1 + tekst1 = " + tekst1 + tekst2);
12        boolean tacno = true;
13        System.out.println("\ntacno == " + tacno);
14        boolean pogresno = false;
15        System.out.println("pogresno = " + pogresno);
16    }
17 }

```

Problems @ Javadoc Declaration Console

<terminated> obuka0207 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 17, 2017 4:26:08 PM)

```

obuka 02 primer 07
primitivne vrednosti

tekst1 =Zdravo
tekst2 =svima!
tekst1 + tekst1 = Zdravo svima!

tacno == true
pogresno = false

```

```

String tekst1 = "Zdravo ";
String tekst2 = "svima!";
boolean tacno = true;
boolean pogresno = false;

```

Primitivne logički i string tip

char 16-bit, Unicode 0 do Unicode $2^{16}-1$

Primitivne karakter tip

```
obuka0208.java
1 package obuka02;
2 import java.util.*;
3 public class obuka0208 {
4     public static void main (String[] args) {
5         System.out.println("obuka 02 primer 08");
6         System.out.println("primitivne vrednosti");
7         char znak88 = '\n';
8         System.out.println("znak88 = " + znak88);
9         char slovoM = 'M';
10        System.out.println("slovoM = " + slovoM);
11        char brojTri = '3';
12        System.out.println("brojTri = " + brojTri);
13        char charE = 0x45;
14        System.out.println("'charE = 0x45;' charE = " + charE);
15        char znakPitanja = '?';
16        System.out.println("znakPitanja = " + znakPitanja);
17        int tri = 3;
18        char cetiri = '4';
19        char broj4plus3 = (char) (cetiri + tri);
20        System.out.println("broj4plus3 = " + broj4plus3);
21    }
22 }
```

Problems Javadoc Declaration Console
<terminated> obuka0208 [Java Application] C:\
obuka 02 primer 08
primitivne vrednosti
znak88 =

slovoM = M
brojTri = 3
'charE = 0x45;' charE = E
znakPitanja = ?
broj4plus3 = 7

```
char slovoM = 'M';
char brojTri = '3';
char charE = 0x45;
char znakPitanja = '?';
char znak88 = '\n';
int tri = 3;
char cetiri = '4';
char broj4plus3 = (char) (cetiri + tri);
```

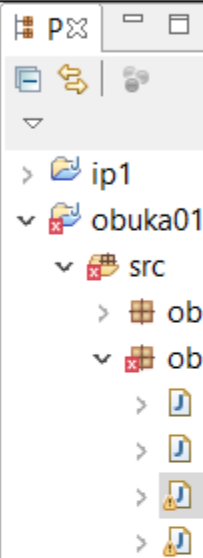
Operatori

File Edit Source Refactor Navigate Search Project

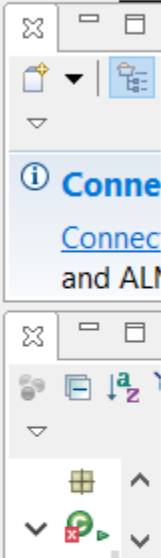


Quick Access

Java



```
obuka0210.java x
1 package obuka02;
2 import java.util.*;
3 public class obuka0210 {
4     public static void main (String[] args) {
5         System.out.println("obuka 02 primer 10");
6         System.out.println("primitivne vrednosti");
7         int m = 4;
8         System.out.println("m = " + m);
9         int m = 3;
10        System.out.println("m = " + m);
11    }
12 }
13
```



Ne mogu dva puta da se definiše ista promenljiva

```
<terminated> obuka0210 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 17, 2017, 6:13:24 PM)
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Duplicate local variable m

    at obuka02.obuka0210.main(obuka0210.java:9)
```

Operatori za celobrojne i racionalne tipove

The screenshot shows an IDE with a Java file named `obuka0211.java`. The code defines a package `obuka02` and a public class `obuka0211` with a `main` method. The code performs several arithmetic operations on integer and double variables and prints the results. The console output shows the execution of the program, displaying the values of the variables and the results of the operations.

```
1 package obuka02;
2 public class obuka0211 {
3     public static void main (String[] args) {
4         System.out.println("obuka 02 primer 11, operatori\n");
5         int m = 63;
6         System.out.println("m = " + m);
7         int n = 35;
8         System.out.println("n = " + n);
9         int mn = m/n;
10        System.out.println("m/n = " + mn);
11        double u = 63;
12        System.out.println("\nu = " + u);
13        double v = 35;
14        System.out.println("v = " + v);
15        double uv = u/v;
16        System.out.println("u/v = " + uv);
17    }
18 }
```

Console Output:

```
<terminated> obuka0211 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (
obuka 02 primer 11, operatori

m = 63
n = 35
m/n = 1

u = 63.0
v = 35.0
u/v = 1.8
```

PEMDAS

Parentheses

Exponents

Multiplication

Division

Addition

Subtraction

 obuka0212.java

```
1 package obuka02;
2 public class obuka0212 {
3     public static void main (String[] args) {
4         System.out.println("obuka 02 primer 12, operatori\n");
5         int m = 63;
6         System.out.println("m = " + m);
7         int n = 35;
8         System.out.println("n = " + n);
9         int mn = m+n;
10        System.out.println("m+n = " + mn);
11        System.out.println("++mn = " + ++mn);
12        System.out.println("mn++ = " + mn++);
13        System.out.println("mn = " + mn);
14        System.out.println("--m = " + --m);
15        System.out.println("m-- = " + m--);
16        System.out.println("m = " + m);
17    }
18 }
```

Find All

Connect to your t

and ALM tools or

a local task.

Outli...

10/10/2014

Obuka02

main/Stri

main(Bn

Problems @ Javadoc Declaration Console

```
<terminated> obuka0212 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 17, 2017, 7:42:45 PM)
```

obuka 02 primer 12, operatori

```
m = 63
n = 35
m+n = 98
++mn = 99
mn++ = 99
mn = 100
--m = 62
m-- = 62
m = 61
```

Postfix forma $x++$, $x--$

Operatori za racionalne, ++x, x++

The screenshot shows an IDE with a Java file named `obuka0213.java`. The code defines a package `obuka02` and a public class `obuka0213` with a `main` method. The code demonstrates various arithmetic and increment/decrement operations:

```
1 package obuka02;
2 public class obuka0213 {
3     public static void main (String[] args) {
4         System.out.println("obuka 02 primer 13, operatori\n");
5         int m = 63;
6         System.out.println("m = " + m);
7         int n = 35;
8         System.out.println("n = " + n);
9         double mn = m+n;
10        System.out.println("'mn = m+n' = " + mn);
11        System.out.println("++mn = " + ++mn);
12        System.out.println("mn++ = " + mn++);
13        System.out.println("mn = " + mn);
14        int v = m++ + n;
15        System.out.println("m++ + n = " + v);
16        int u = ++n + m;
17        System.out.println("++n + m= " + u);
18    }
19 }
```

The console output shows the results of these operations:

```
<terminated> obuka0213 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 17, 2017, 7:54:58 PM)
obuka 02 primer 13, operatori

m = 63
n = 35
'mn = m+n' = 98.0
++mn = 99.0
mn++ = 99.0
mn = 100.0
m++ + n = 98
++n + m= 100
```

On the right side of the IDE, there is a sidebar with a search bar, a "Connect Mylyn" button, and an "Outlines" view showing the class structure:

- obuka02
 - obuka0213
 - main(Stri

inkrement dekrement
Prefix forma ++x, --x
Postfix forma x ++, x --

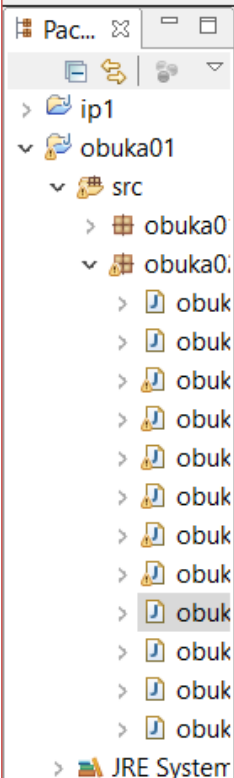
Relazioni operatori

File Edit Source Refactor Navigate Search



Quick Access

Java



```

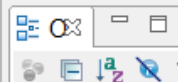
1 package obuka02;
2 public class obuka0214 {
3     public static void main (String[] args) {
4         System.out.println("obuka 02 primer 14, operatori\n");
5         int m = 63;
6         int n = 35;
7         boolean mn = (m==n);
8         boolean notmn = (m!=n);
9         System.out.println("m = " + m + ", n = " + n);
10        System.out.println("(m==n) = " + mn);
11        System.out.println("(m!=n) = " + notmn);
12        String u = "63";
13        String v = "63";
14        boolean uv = (u==v);
15        boolean notuv = (u!=v);
16        System.out.println("u = " + u + ", v = " + v);
17        System.out.println("(u==v) = " + uv);
18        System.out.println("(u!=v) = " + notuv);
19    }
20 }

```



Find

Connect I
[Connect to](#)
 and ALM to
 a local task



obuka0214
 obuka0214
 ma

Problems Javadoc Declaration Console

<terminated> obuka0214 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 17, 2017, 9:32:21 PM)
 obuka 02 primer 14, operatori

```

m = 63, n = 35
(m==n) = false
(m!=n) = true
u = 63, v = 63
(u==v) = true
(u!=v) = false

```

==, !=

Samo za brojeve >, <, >=, <=



Logički operatori

```
1 package obuka02;
2 public class obuka0215 {
3     public static void main (String[] args) {
4         System.out.println("obuka 02 primer 15, operatori\n");
5         boolean m = false;
6         boolean n = true;
7         boolean maan = (m&& n);
8         boolean man = (m& n);
9         boolean moon = (m|| n);
10        boolean mon = (m| n);
11        System.out.println("m = " + m + ", n = " + n);
12        System.out.println("(m&&n) = " + maan);
13        System.out.println("(m&n) = " + man);
14        System.out.println("(m||n) = " + moon);
15        System.out.println("(m|n) = " + mon);
16    }
```

```
<terminated> obuka0215 [Java Application] C:\Program Files\Java\jre1.8.0
obuka 02 primer 15, operatori

m = false, n = true
(m&&n) = false
(m&n) = false
(m||n) = true
(m|n) = true
```

&&, kratko AND
&, logičko AND
||, kratko OR
|, logičko OR

Kontrola toka

The screenshot shows an IDE with the following components:

- File Explorer:** Shows a project structure with folders 'ip1', 'obuka01', and 'src'. Under 'src', there are folders 'obuka01' and 'obuka02'. The 'obuka02' folder contains several Java files, including 'Kubovij.java' and 'obuka0216.java'.
- Code Editor:** Displays the code for 'obuka0216.java'. The code is as follows:

```
1 package obuka02;  
2 public class obuka0216 {  
3     public static void main (String[] args) {  
4         System.out.println("obuka 02 primer 16, kontrole toka\n")  
5         int a = 65;  
6         int b = 79;  
7         int c = 79;  
8         if (a<b) {  
9             System.out.println("a = " + a + ", b = " + b);  
10            System.out.println("a < b");  
11        }  
12        if (b>a) {  
13            System.out.println("\na = " + a + ", b = " + b);  
14            System.out.println("b > a");  
15        }  
16    }  
17 }
```
- Console:** Shows the output of the program execution. The output is:

```
<terminated> obuka0216 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 18, 2017, 4:42:37 PM)  
obuka 02 primer 16, kontrole toka  
  
a = 65, b = 79  
a < b  
  
a = 65, b = 79  
b > a
```
- Right Panel:** Contains a 'Connect' button and a 'Find' search bar.

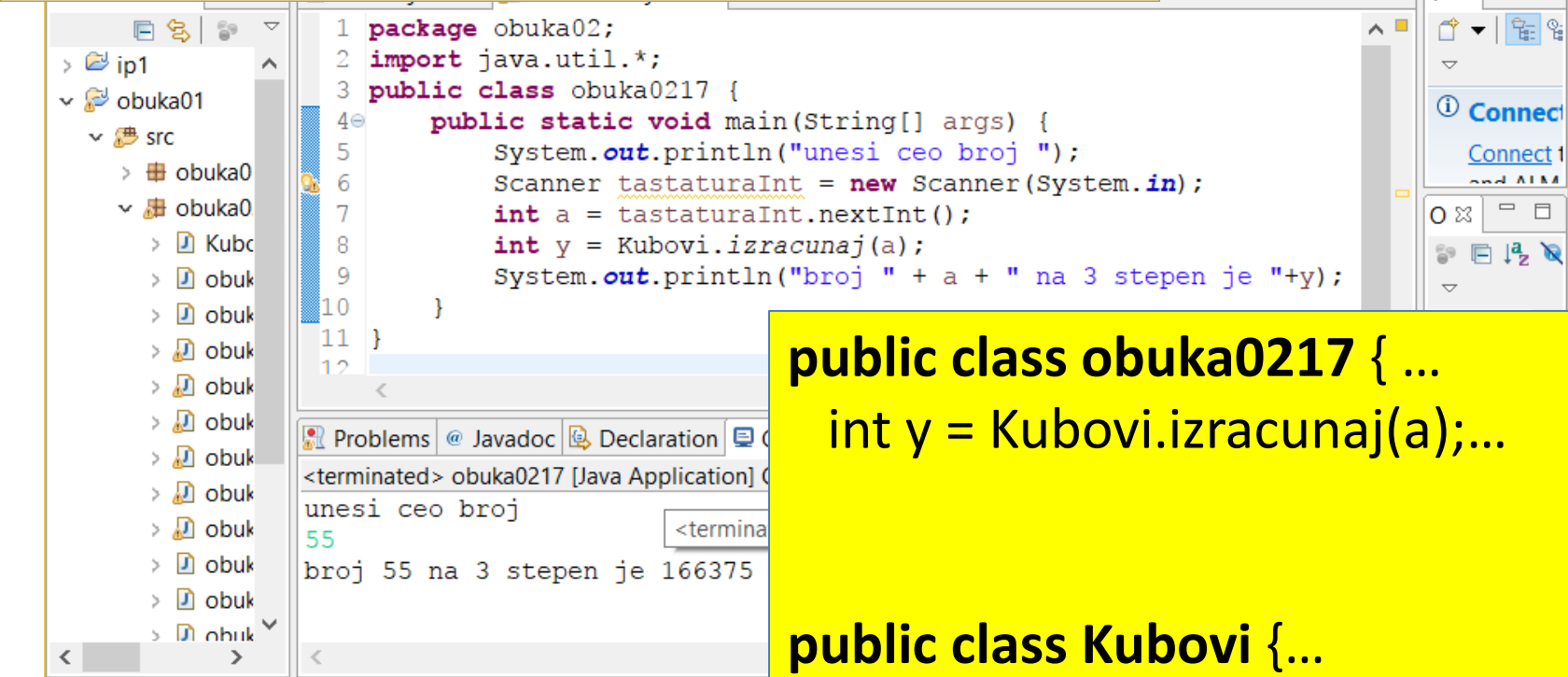
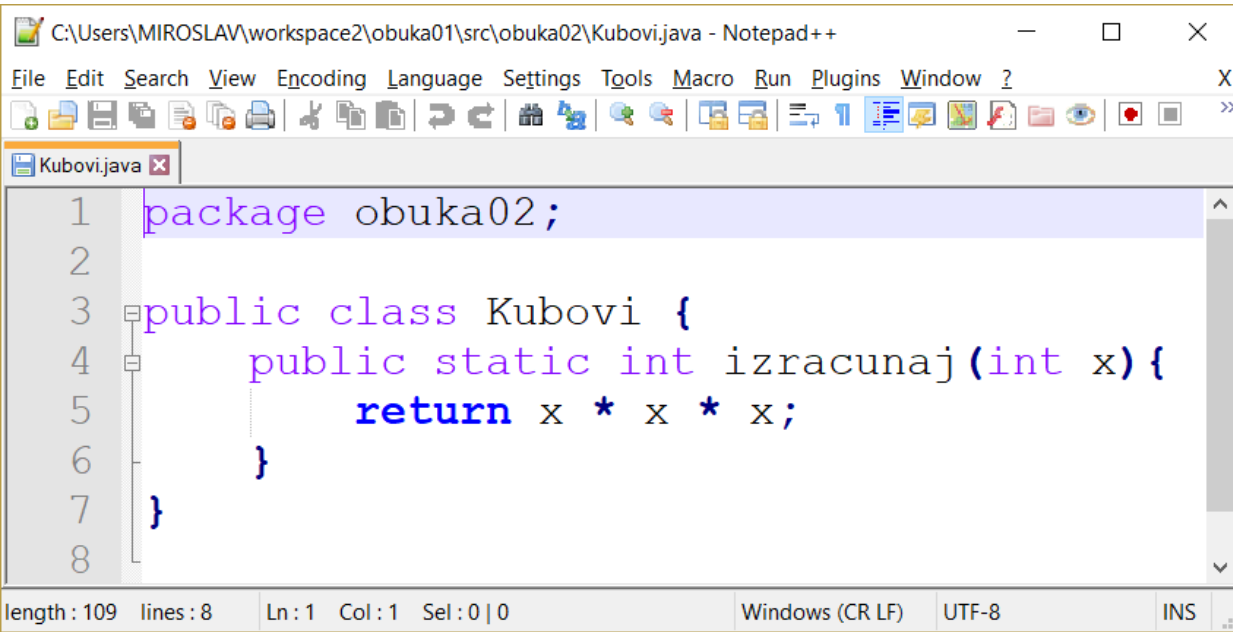
The code in the editor is a Java program that checks if a number is even or odd. It uses the modulo operator (%) to check if a number is divisible by 2. If the remainder is 0, the number is even; otherwise, it is odd. The program prints the result to the console.

The console output shows the program execution results. The output is:

```
<terminated> obuka0216 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 18, 2017, 4:42:37 PM)  
obuka 02 primer 16, kontrole toka  
  
a = 65, b = 79  
a < b  
  
a = 65, b = 79  
b > a
```

If (izraz) {
naredba;
}

Kontrola toka



```
public class obuka0217 { ...  
    int y = Kubovi.izracunaj(a);...  
  
public class Kubovi {...
```

```
public class obuka0217 { ...  
    int y = Kubovi.izracunaj(a);...  
  
public class Kubovi {...
```

Kontrola toka

```

1 package obuka02;
2 import java.util.*;
3 public class obuka0217 {
4     public static void main(String[] args) {
5         System.out.println("unesi ceo broj ");
6         Scanner tastaturaInt = new Scanner(System.in);
7         int a = tastaturaInt.nextInt();
8         int y = Kubovi.izracunaj(a);
9         System.out.println("broj " + a + " na 3 stepen je "+y);
10    }
11 }
12

```

unesi ceo broj

```

1 package obuka02;
2
3 public class Kubovi {
4     public static int izracunaj(int x){
5         return x * x * x;
6     }
7 }
8

```

unesi ceo broj

23

<terminated> obuka0217 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 18, 2017, 5:29:43 PM)

unesi ceo broj

23

broj 23 na 3 stepen je 12167

File Edit Source Refactor Navigate Search Project Run Window Help



Quick Access



Pack...

DeljivoAsaB.java obuka0218.java

```
1 package obuka02;
2 import java.util.*;
3 public class obuka0218 {
4     public static void main(String[] args) {
5         System.out.println("unesi prvi ceo broj ");
6         Scanner tastaturaInt = new Scanner(System.in);
7         int a = tastaturaInt.nextInt();
8         System.out.println("unesi delilac ceo broj ");
9         int b = tastaturaInt.nextInt();
10        int y = DeljivoAsaB.izracunaj(a,b);
11        System.out.println("ostatak deljenja " + a + " sa " + b + " je "+y);
12    }
13 }
```

Problems @ Javadoc Declaration Console

<terminated> obuka0218 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 18, 2017, 5:57:29 PM)

```
unesi prvi ceo broj
36
unesi delilac ceo broj
5
ostatak deljenja 36 sa 5 je 1
```

Writable Smart Insert 13:2

DeljivoAsaB.java

```
1 package obuka02;
2
3 public class DeljivoAsaB {
4     public static int izracunaj(int x, int y){
5         return x % y;
6     }
7 }
```

VISER 60

Java src length: 117 lines: 8 Ln: 1 Col: 1 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

File Edit Source Refactor Navigate Search Project Run Window Help



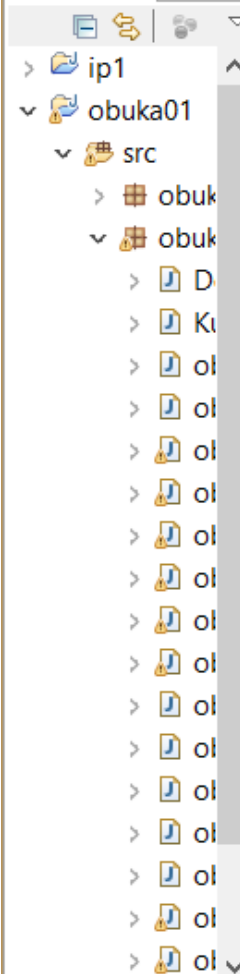
Quick Access

Java

Pac...

obuka0219.java

T



```
1 package obuka02;
2 public class obuka0219 {
3     public static void main (String[] args) {
4         System.out.println("obuka 02 primer 19, kontrola toka\n");
5         int ocena = 8;
6         if (ocena == 10)
7             System.out.println("student je dobio 10.");
8         else if (ocena == 9)
9             System.out.println("student je dobio 9.");
10        else if (ocena == 8)
11            System.out.println("student je dobio 8.");
12        else if (ocena == 7)
13            System.out.println("student je dobio 7.");
14        else if (ocena == 6)
15            System.out.println("student je dobio 6.");
16        else
17            System.out.println("student je pao na ispitu.");
18    }
19 }
```

Problems Javadoc Declaration Console

```
<terminated> obuka0219 [Java Application] C:\Program Files\Java\
obuka 02 primer 19, kontrola toka

student je dobio 8.
```

Writable

Smart Insert

19 : 2

VISER

```
if (logicki_izraz) {
    naredba1
}
else {
    naredba2
}
```

File Edit Source Refactor Navigate Search Project Run Window Help



Quick Access

Java

Packag...

obuka0220.java

```
1 package obuka02;
2 public class obuka0220 {
3     public static void main (String[] args) {
4         System.out.println("obuka 02 primer 20, kontrola toka\n");
5         int ocena = 8;
6         switch (ocena){
7             case 10:
8                 System.out.println("student je dobio 10."); break;
9             case 9:
10                System.out.println("student je dobio 9."); break;
11             case 8:
12                System.out.println("student je dobio 8."); break;
13             case 7:
14                System.out.println("student je dobio 7."); break;
15             case 6:
16                System.out.println("student je dobio 6."); break;
17             default:
18                System.out.println("student je dobio 8.");
19         }
20     }
```

Problems Javadoc Declaration Console

<terminated> obuka0220 [Java Application] C:\Program Files\J
obuka 02 primer 20, kontrola toka

student je dobio 8.

Writable

Smart Insert

VISER 4:47

```
switch (int_promenljiva) {
case 1: naredba1; break;
case 2: naredba2; break;
...
default: naredbaOstalo;
}
```

File Edit Source Refactor Navigate Search Project Run Window Help



Quick Access

Java

Packag...

obuka0221.java

```
1 package obuka02;
2 public class obuka0221 {
3     public static void main (String[] args) {
4         System.out.println("obuka 02 primer 21, petlje\n");
5         int istop = 5;
6         int istart = 1;
7         int fakt = 1;
8         for (int i=istart;i<istop;i++){
9             fakt *= i;
10            System.out.println("i = " + i + ", fakt = " + fakt);
11        }
12    }
13 }
```

Problems Javadoc Declaration Console

<terminated> obuka0221 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 19, 2017, 6:12:37 AM)

obuka 02 primer 21, petlje

```
i = 1, fakt = 1
i = 2, fakt = 2
i = 3, fakt = 6
i = 4, fakt = 24
```

```
for (inicijalizacija;uslov;inkrement) {
    naredbe;
    ...
}
```

Writable

Smart Insert

13 : 2

File Edit Source Refactor Navigate Search Project Run Window Help



Quick Access

Java

Packag...

obuka0222.java

```
1 package obuka02;
2 public class obuka0222 {
3     public static void main (String[] args) {
4         System.out.println("obuka 02 primer 22, petlje\n");
5         int istop = 5;
6         int i = 1;
7         int fakt = 1;
8         while (i<istop){
9             fakt *= i;
10            i += 1;
11            System.out.println("i = " + i + ", fakt = " + fakt);
12        }
13    }
```

Stop rada

Problems Javadoc Declaration Console

<terminated> obuka0222 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 19, 2017, 6:24:07 AM)

obuka 02 primer 22, petlje

```
i = 2, fakt = 1
i = 3, fakt = 2
i = 4, fakt = 6
i = 5, fakt = 24
```

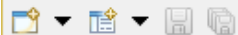
while (uslov) {
inkrement;
naredbe;

...
}

Writable

Smart Insert

10:19



Quick Access

Java

Packag...

obuka0224.java

```
1 package obuka02;
2 public class obuka0224 {
3     public static void main (String[] args) {
4         System.out.println("obuka 02 primer 24, petlje\n");
5         int istop = 5+1;
6         int i = 0;
7         int fNep = 1;
8         while (++i<istop){
9             if (i%2 == 0)
10                continue;
11             System.out.println("i = " + (i-1) + ", parni");
12             fNep *= i;
13             System.out.println("i = " + i + ", fakt = " + fNep);
14         }
15     }
16 }
```

T



Connect

Connect to
and ALM
a local tas

O



▼

obuka

obuka

s m

< >

Problems @ Javadoc Declaration Console



<terminated> obuka0224 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 19, 2017, 6:46:53 AM)

obuka 02 primer 24, petlje

```
i = 0, parni
i = 1, fakt = 1
i = 2, parni
i = 3, fakt = 3
i = 4, parni
i = 5, fakt = 15
```

while sa inkrementom



Quick Access

Java

Pack...

obuka0225.java

T

obuka

Del

Kut

obu

obu

obu

obu

obu

obu

obu

obu

obu

obu

obu

obu

obu

obu

obu

obu

obu

obu

```
1 package obuka02;
2 public class obuka0225 {
3     public static void main (String[] args) {
4         System.out.println("obuka 02 primer 25, nizovi\n");
5         String[] str = {"Marko", "Jovan", "Milan", "Ana"};
6         int nP = str.length;
7         for (int i=0; i<nP; i++){
8             System.out.println((i+1) + ". | " + str[i]);
9         }
10    }
11 }
```

Problems Javadoc Declaration Console

<terminated> obuka0225 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Dec 19, 2017)

obuka 02 primer 25, nizovi

```
1. Marko
2. Jovan
3. Milan
4. Ana
```

Open Console

String[]

Writable

Smart Insert

8 : 42