

Napredno programiranje i programski jezici

06 Java

Fakultet tehničkih nauka, Novi Sad

23-24/Z

Dunja Vrbaški

Java PL:

- James Gosling
- 1995
- Sun Microsystems → Oracle
- sintaksa slična kao C++

C++ kompajler → izvršni kod

Write once compile anywhere

Java kompajler → bytecode → JVM interpreter izvršava

Write once run anywhere

Jednom kad prevedemo program napisan u **Java** možemo da ga izvršavamo na bilo kojoj platformi koja podržava **Javu**.

Java - jezik

Java - platforma

C++ kompjajler → izvršni kod

Različiti kompjajleri

Java kompjajler → bytecode → JVM interpreter izvršava

Različiti kompjajleri

Različite JVM

C++ vs Java

| C++ | Java |
|----------------------------------|------------------------------------|
| kompajler | kompajler + interpreter |
| proceduralno + OO | OO |
| x | jedinstvena hijerarhija klasa |
| preklapanje operatora | x |
| virtual deklaracija | sve (nestatičke) metode su virtual |
| x | GC |
| pass by value, pass by reference | pass by value |

C++ vs Java

| C++ | Java |
|---------------------------------|------------------------------|
| private, protected nasleđivanje | x |
| višestruko nasleđivanje | x |
| const | x |
| x | interface |
| x | abstract kw |
| .hpp, .cpp | .java (isto kao naziv klase) |
| ... | |

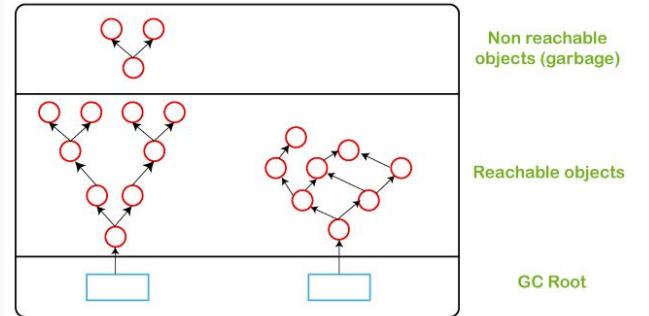
filozofija?

Garbage Collection (GC)

Automatsko upravljanje memorijom

- | | |
|------|----------------------|
| C++ | - new/ delete |
| Java | - new |

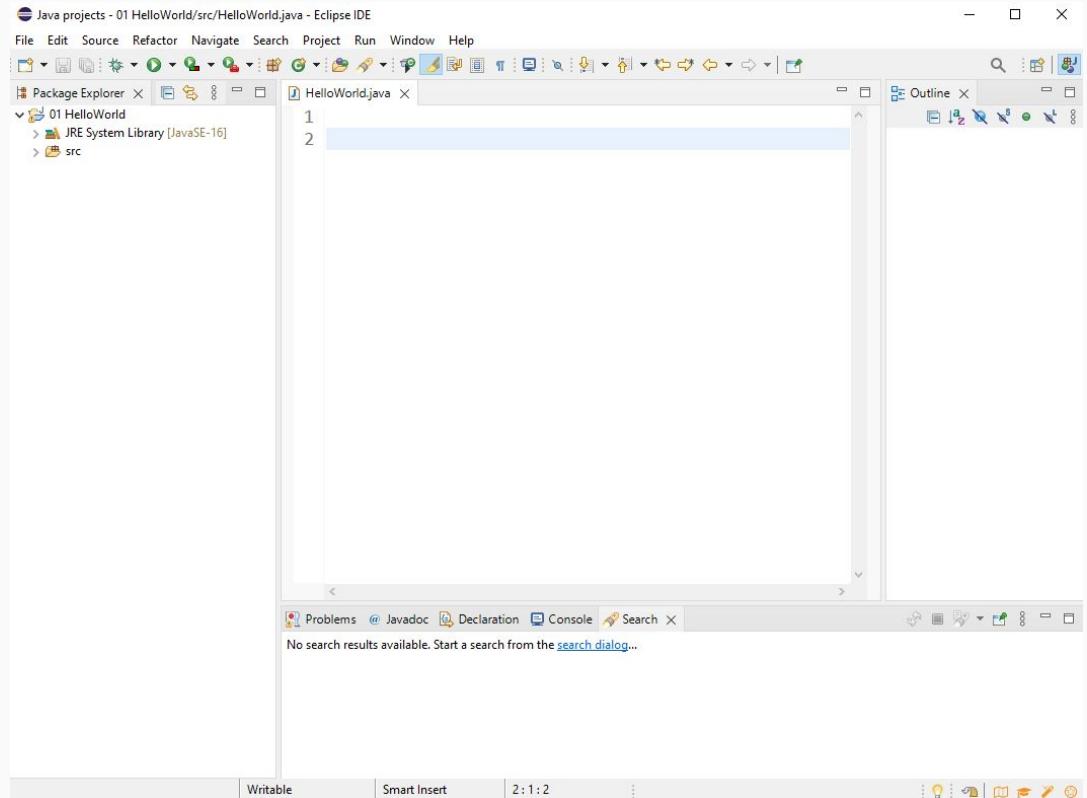
- programeri oslobođeni odgovornosti
- ne znamo trenutak
- različiti algoritmi



mark&swipe

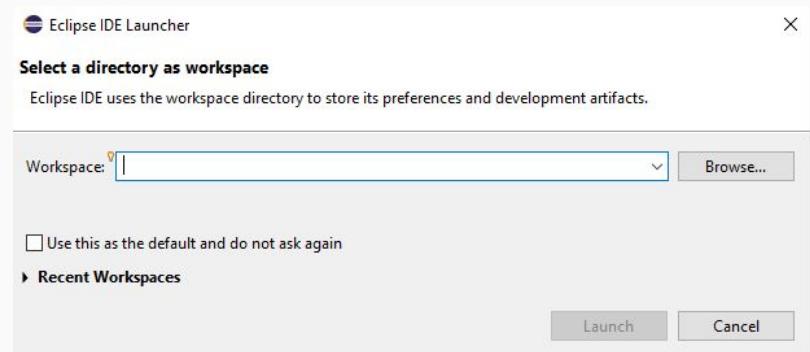
Eclipse IDE

<https://www.eclipse.org/>



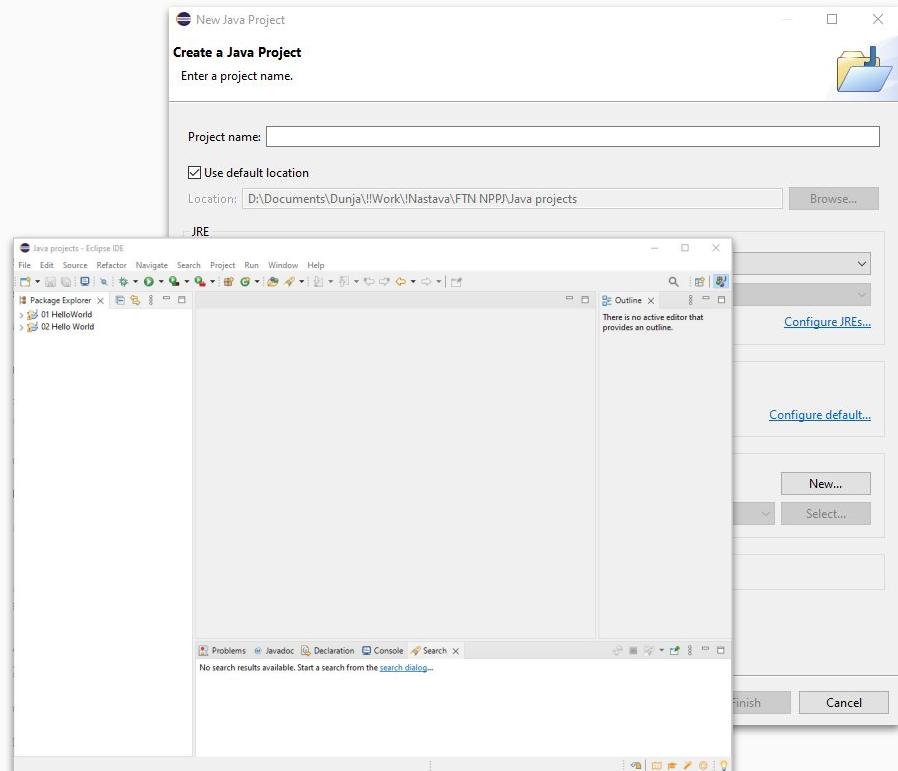
Workspace

projekti + podešavanja, informacije



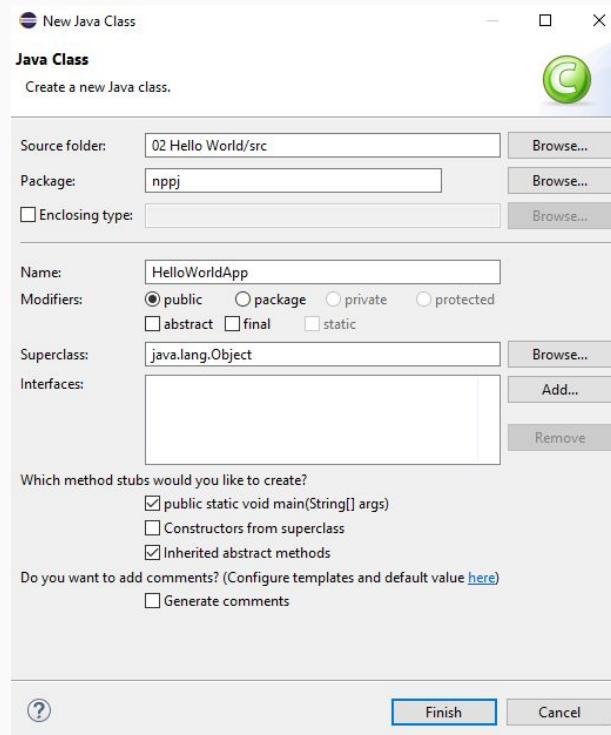
Projekat

Java project



Klasa

*sa main metodom



Java projects - 02 Hello World/src/nppj/HelloWorldApp.java - Eclipse IDE

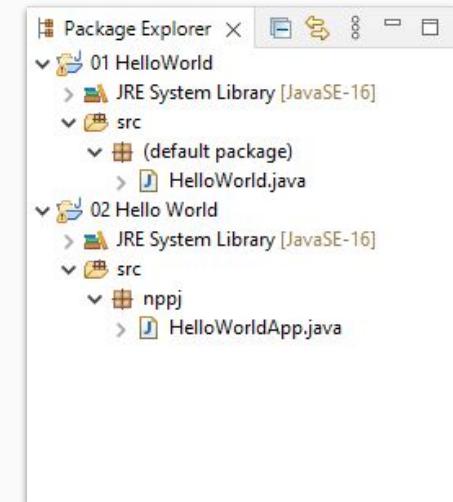
File Edit Source Refactor Navigate Project Run Window Help

The screenshot shows the Eclipse IDE interface. The top bar displays the title "Java projects - 02 Hello World/src/nppj/HelloWorldApp.java - Eclipse IDE" and the menu items: File, Edit, Source, Refactor, Navigate, Project, Run, Window, Help. Below the menu is a toolbar with various icons. On the left is the "Package Explorer" view, which shows a project structure: 01 HelloWorld, 02 Hello World (selected), JRE System Library [JavaSE-16], src, and nppj (containing HelloWorldApp.java). The main workspace contains a code editor titled "HelloWorldApp.java" with the following content:

```
1 package nppj;
2
3 public class HelloWorldApp {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8     }
9
10 }
11
```

Paket

- modularizacija, organizacija
- imenovanje
- prava pristupa



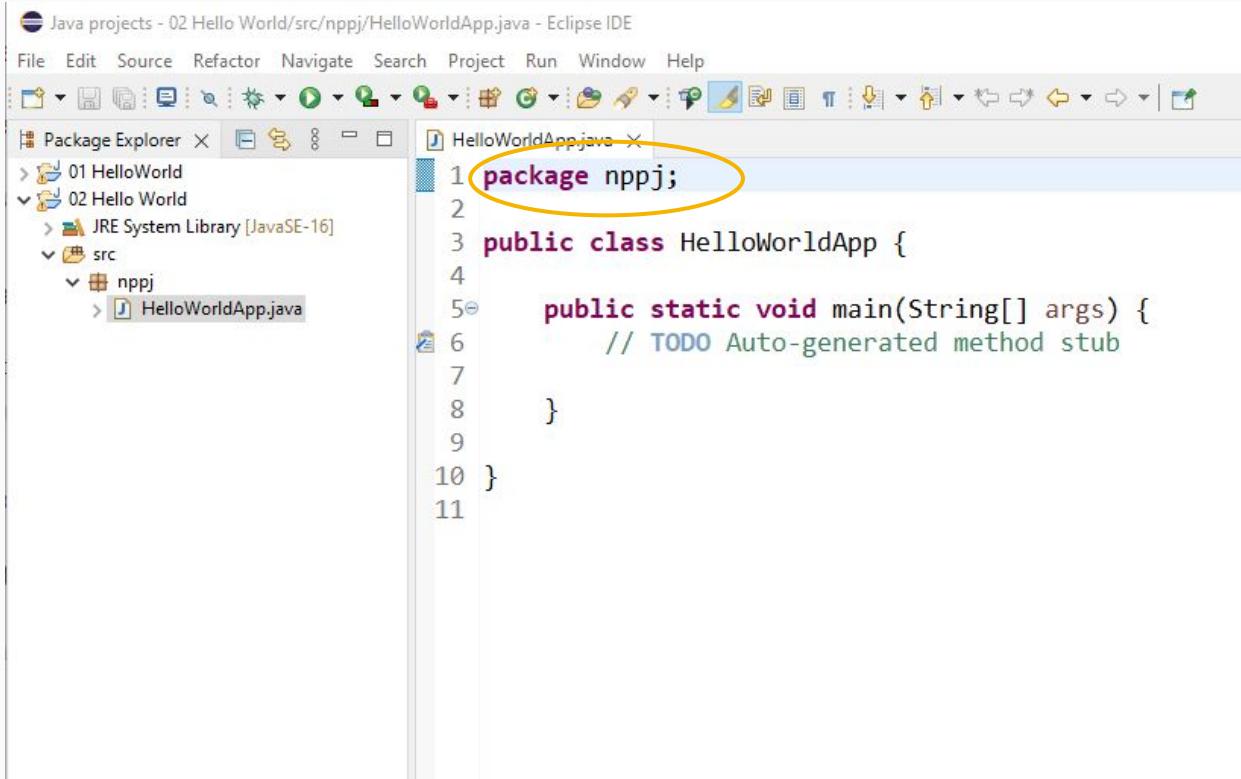
prava pristupa, vidljivost, prevođenje...: c++ vs java

Java projects - 02 Hello World/src/nppj/HelloWorldApp.java - Eclipse IDE

File Edit Source Refactor Navigate Project Run Window Help

Package Explorer X HelloWorldApp.java

```
1 package nppj;
2
3 public class HelloWorldApp {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8     }
9
10 }
11
```



Java projects - 02 Hello World/src/nppj/HelloWorldApp.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X HelloWorldApp.java X

The screenshot shows the Eclipse IDE interface. The left pane displays the 'Package Explorer' with two projects: '01 HelloWorld' and '02 Hello World'. The '02 Hello World' project contains a 'src' folder which includes a package named 'nppj' containing a file named 'HelloWorldApp.java'. The right pane shows the content of 'HelloWorldApp.java'.

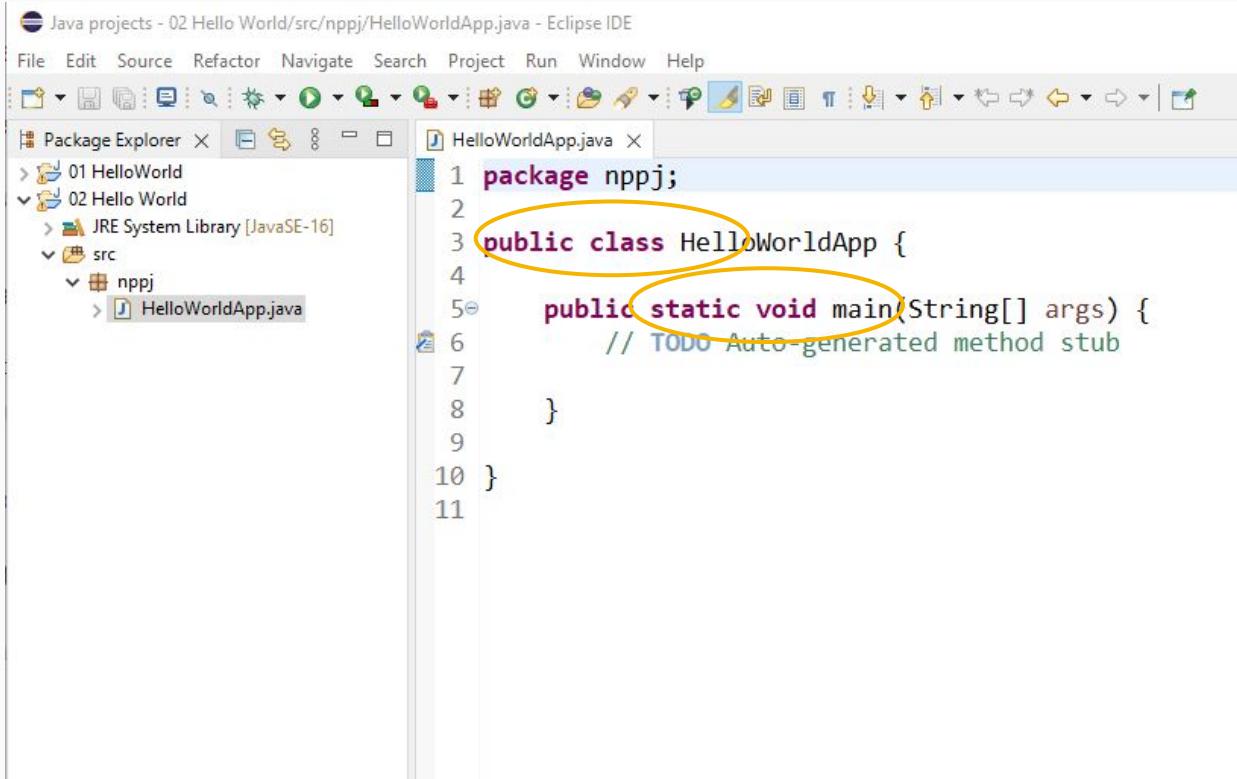
```
1 package nppj;
2
3 import java.util.*;
4 import java.util.LinkedList;
5
6 public class HelloWorldApp {
7
8     public static void main(String[] args)
9     {
10 }
11}
12
```

Java projects - 02 Hello World/src/nppj/HelloWorldApp.java - Eclipse IDE

File Edit Source Refactor Navigate Project Run Window Help

Package Explorer X HelloWorldApp.java X

```
1 package nppj;
2
3 public class HelloWorldApp {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8     }
9
10 }
11
```



The screenshot shows the Eclipse IDE interface with the title bar "Java projects - 02 Hello World/src/nppj/HelloWorldApp.java - Eclipse IDE". The menu bar includes File, Edit, Source, Refactor, Navigate, Project, Run, Window, and Help. Below the menu is a toolbar with various icons. On the left is the "Package Explorer" view, which shows a project structure with two packages: "01 HelloWorld" and "02 Hello World". Under "02 Hello World", there is a "src" folder containing a "nppj" folder, which contains the file "HelloWorldApp.java". The "HelloWorldApp.java" file is open in the central editor area. The code is as follows:

```
1 package nppj;
2
3 public class HelloWorldApp {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8     }
9
10 }
11
```

The first two lines ("package nppj;" and "public class HelloWorldApp {") are circled in yellow.

| C++ | Java |
|-----------------------------|-------------------------------|
| x | jedinstvena hijerarhija klasa |
| proceduralno + OO | OO |
| main() je slobodna funkcija | main() je metod neke klase |

Java - sve klase su nasleđene od jedne zajedničke klase Object

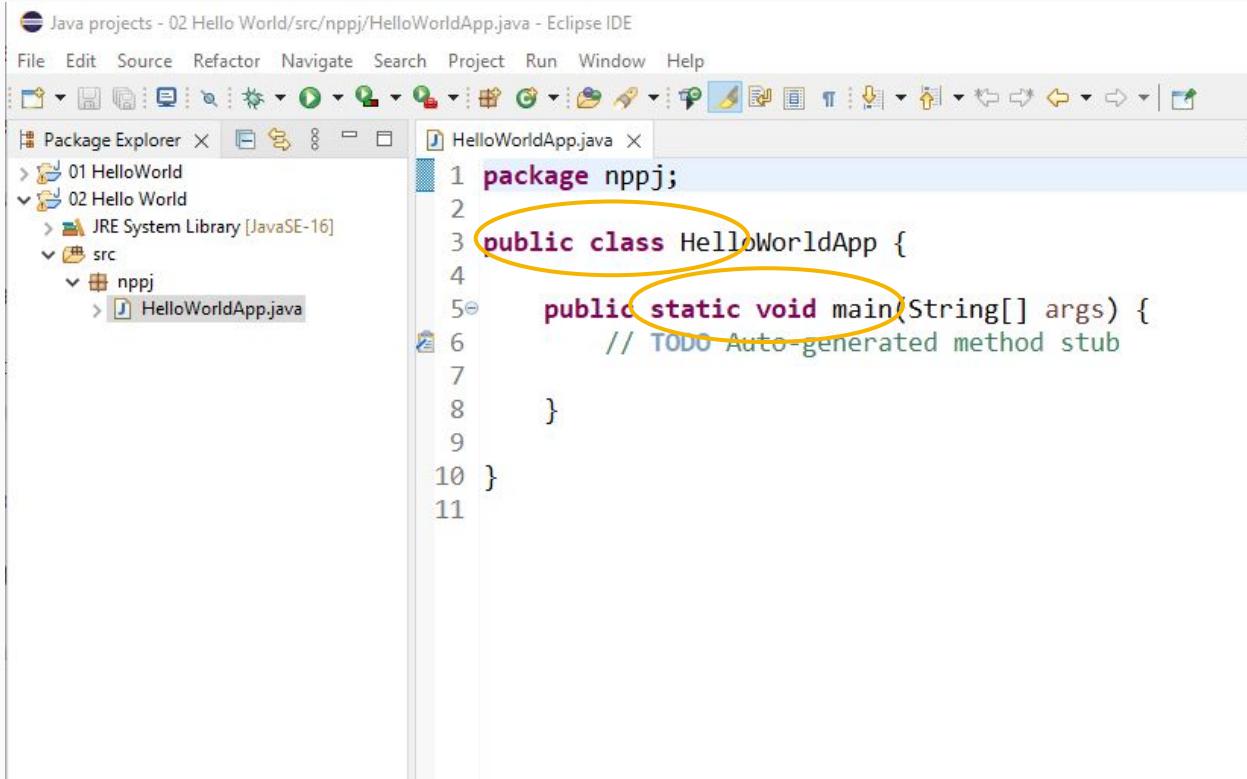
Nema slobodnih funkcija

Java projects - 02 Hello World/src/nppj/HelloWorldApp.java - Eclipse IDE

File Edit Source Refactor Navigate Project Run Window Help

Package Explorer X HelloWorldApp.java X

```
1 package nppj;
2
3 public class HelloWorldApp {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8     }
9
10 }
11
```



The screenshot shows the Eclipse IDE interface with a Java project named '02 Hello World'. The 'src' folder contains a package named 'nppj' which holds the file 'HelloWorldApp.java'. The code editor displays the following Java code:

```
1 package nppj;
2
3 public class HelloWorldApp {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8     }
9
10 }
11
```

The words 'nppj' in the package declaration and 'public static void main(String[] args)' in the main method are highlighted with yellow circles.

static - u nastavku...

Java projects - 02 Hello World/src/nppj/HelloWorldApp.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

The screenshot shows the Eclipse IDE interface. The title bar reads "Java projects - 02 Hello World/src/nppj/HelloWorldApp.java - Eclipse IDE". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. Below the menu is a toolbar with various icons. On the left is the "Package Explorer" view, which lists two projects: "01 HelloWorld" and "02 Hello World". Each project contains a "src" folder with a "HelloWorld.java" file. The "02 Hello World" project also contains a "nppj" folder. The main workspace shows the content of "HelloWorldApp.java". The code is as follows:

```
1 package nppj;
2
3 public class HelloWorldApp {
4
5     public static void main(String[] args) {
6         System.out.println("Hello World!");
7     }
8
9 }
10
```

Java projects - 02 Hello World/src/nppj/HelloWorldApp.java - Eclipse IDE

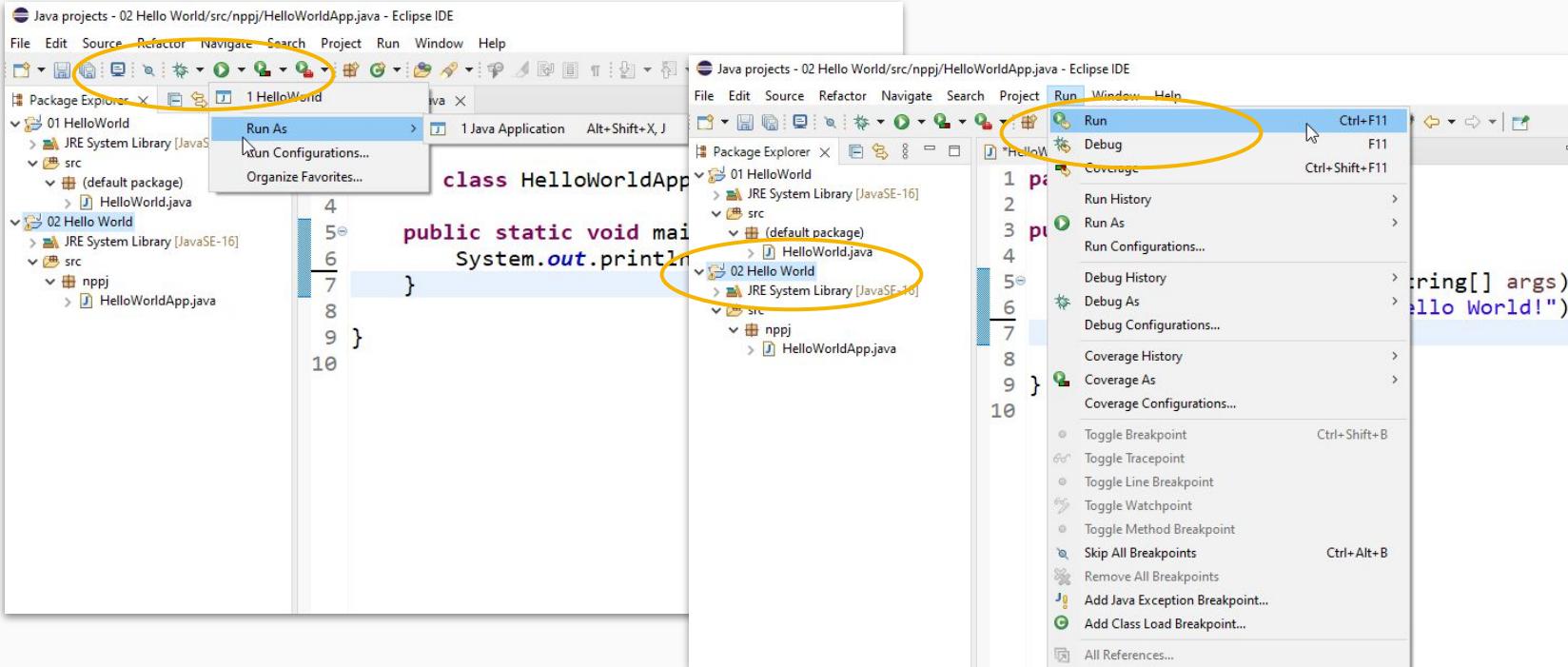
File Edit Source Refactor Navigate Search Project Run Window Help

The screenshot shows the Eclipse IDE interface with the title bar "Java projects - 02 Hello World/src/nppj/HelloWorldApp.java - Eclipse IDE". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. Below the menu is a toolbar with various icons. On the left is the Package Explorer view showing two projects: "01 HelloWorld" and "02 Hello World". The "02 Hello World" project has a "src" folder containing a "nppj" package with a file named "HelloWorldApp.java". The main editor window displays the following Java code:

```
1 package nppj;
2
3 public class HelloWorldApp {
4
5     public static void main(String[] args)
6         System.out.println("Hello World!");
7         sysout
8     }
9
10 }
11
```

The cursor is at line 7, position 10, where the word "sysout" is highlighted in blue, indicating it is a suggestion from the code completion feature. A tooltip or dropdown menu is visible above the cursor, showing the suggestion "sysout". The status bar at the bottom right shows the text "ctrl + space".

ctrl + space



F11

Java projects - 02 Hello World/src/nppj/HelloWorldApp.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X HelloWorldApp.java X

01 HelloWorld JRE System Library [JavaSE-16] src (default package) HelloWorld.java

02 Hello World JRE System Library [JavaSE-16] src nppj HelloWorldApp.java

```
1 package nppj;
2
3 public class HelloWorldApp {
4
5     public static void main(String[] args)
6         System.out.println("Hello World!");
7     }
8
9 }
10
```

Outline X

nppj HelloWorldApp main(String[]): void

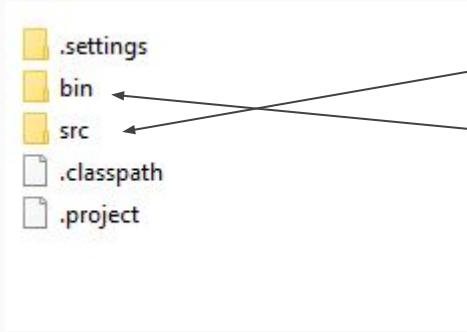
Problems Declaration Console X Search

<terminated> HelloWorldApp [Java Application] C:\Users\Dušan\p2\poch\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211

Hello World!

Writable Smart Insert 10 : 1 : 140

A screenshot of the Eclipse IDE interface. The central part shows the code editor with a Java file named 'HelloWorldApp.java'. The code prints 'Hello World!' to the console. To the right of the code editor is the 'Outline' view, which shows the class structure. Below the code editor is the 'Console' view, which displays the output 'Hello World!', with the word 'Hello' highlighted by a yellow oval. The status bar at the bottom shows the file path and current line number.



HelloWorldApp.java

HelloWorldApp.class

Java projects - 02 Hello World/src/nppj/HelloWorldApp.java - Eclipse IDE

File Edit Source Refactor Navigate Project Run Window Help

Package Explorer X "HelloWorldApp.java" Outline X

```
1 package nppj;
2
3 public class HelloWorldApp {
4
5     public static void main(String[] args)
6         System.out.println("Hello World!");
7         kgh;
8     }
9 }
```

kgh cannot be resolved
4 quick fixes available:

- ① Create local variable 'kgh'
- ② Create field 'kgh'
- ③ Create parameter 'kgh'
- ④ Create constant 'kgh'

Problems @ Javadoc Declaration Console X Search

<terminated> HelloWorldApp [Java Application] C:\Users\Dunja\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v2021

Hello World!

The screenshot shows the Eclipse IDE interface with a Java project named '02 Hello World'. In the 'HelloWorldApp.java' file, there is a syntax error at line 7, column 10, where the identifier 'kgh' is used without being declared. A tooltip box appears over the 'kgh' identifier, displaying the error message 'kgh cannot be resolved' and four 'quick fixes available': 'Create local variable 'kgh'', 'Create field 'kgh'', 'Create parameter 'kgh'', and 'Create constant 'kgh''. The 'Outline' view on the right shows the class structure with the method 'main' highlighted. The 'Console' tab at the bottom shows the output 'Hello World!' from the application.

```
package nppj;

public class HelloWorldApp {

    public static void main(String[] args) {
        int x = 5;
        int y = 3;
        int z = x + y;

        System.out.println("zbir je: " + z);
        System.out.println("zbir je: " + (x + y));
    }
}
```

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows two projects: "01 HelloWorld" and "02 Hello World".
- Editor:** Displays the file "HelloWorldApp.java" containing Java code:

```
1 package nppj;
2
3 import java.util.Iterator;
4
5 public class HelloWorldApp {
6
7     public static void main(String[] args) {
8         int x = 5;
9         int y = 3;
10        int z = x + y;
11        System.out.println("zbir je: " + z);
12        System.out.println("zbir je: " + (x + y));
13
14        z = 0;
15        for|
```
- Code Completion:** A dropdown menu is open at the cursor position (line 15, character 5), listing several 'for' loop variants:
 - for - iterate over collection
 - for - use index on array
 - for - use index on array with temporary variable
 - foreach - iterate over an array or Iterable
 - formap - iterate over map
- Output:** The terminal window shows the output of the program:

```
zbir je: 8
zbir je: 8
```
- Status Bar:** Shows "Writable", "Smart Insert", and the current time "15:12:282".

ctrl + space
tab

```
package nppj;

public class HelloWorldApp {

    public static void main(String[] args) {
        int x = 5;
        int y = 3;
        int z = x + y;
        System.out.println("zbir je: " + z);
        System.out.println("zbir je: " + (x + y));

        z = 0;
        for (int i = 0; i < 10; i++) {
            z += i;
        }
        System.out.println("zbir je: " + z);
    }
}
```

The screenshot shows the Eclipse IDE interface. The title bar reads "Java projects - 03 Klase/src/nppj/Pravougaonik.java - Eclipse IDE". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help. The toolbar has various icons for file operations. The Package Explorer view on the left shows a project structure with nodes like "01 HelloWorld", "02 Hello World", "03 Klase" (expanded), "JRE System Library [JavaSE-16]", "src" (expanded), and "nppj" (expanded). Inside "nppj", there are files "KlaseTestApp.java" and "Pravougaonik.java". The central editor area displays the following Java code:

```
1 package nppj;
2
3 public class Pravougaonik {
4
5 }
6
```

fajl se zove isto kao klasa

Java projects - 03 Klase/src/nppj/Pravougaonik.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X KlaseTestApp.java Pravougaonik.java

```
1 package nppj;
2
3 public class Pravougaonik {
4     public double a;
5     public double b;
6
7     //TODO metode
8
9 }
10
```

Problems Javadoc Declaration Console Search Tasks

1 items

| | Description | Resource | Path | Location |
|---|-------------|-----------------|--------------------|----------|
| ! | TODO metode | Pravougaonik... | /03 Klase/src/nppj | line 7 |

The screenshot shows the Eclipse IDE interface with a Java project named '03 Klase'. The 'src' folder contains a package 'nppj' which has a class 'Pravougaonik'. The code for 'Pravougaonik.java' is displayed in the editor. A yellow circle highlights the TODO comment 'TODO metode' at line 7. The 'Problems' view at the bottom shows one item: 'TODO metode' located at line 7 of the 'Pravougaonik' resource. The status bar at the bottom right indicates '29'.

Java projects - 03 Klase/src/nppj/KlaseTestApp.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X KlaseTestApp.java X Pravougaonik.java

```
1 package nppj;
2
3 public class KlaseTestApp {
4
5     public static void main(String[] args) {
6
7         Pravougaonik p = new Pravougaonik();
8
9     }
10
11 }
12
```

Problems Javadoc Declaration Console Search Tasks

| ! | Description | Resource | Path | Location |
|---|-------------|-----------------|--------------------|----------|
| | TODO metode | Pravougaonik... | /03 Klase/src/nppj | line 7 |

```
package nppj;

public class Pravougaonik {
    public double a;
    public double b;

    //TODO metode
}
```

```
package nppj;

public class KlaseTestApp {
    public static void main(String[] args) {
        Pravougaonik p = new Pravougaonik();
    }
}
```

```
package nppj;

public class Pravougaonik {
    public double a;
    public double b;

    //TODO metode
}
```

```
package nppj;

public class KlaseTestApp {

    public static void main(String[] args) {

        Pravougaonik p = new Pravougaonik();
        p.a = 5;
        p.b = 3;
    }
}
```

```
package nppj;

public class Pravougaonik {
    private double a;
    private double b;

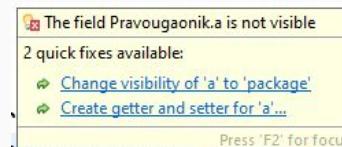
    //TODO metode
}
```

```
package nppj;

public class KlaseTestApp {

    public static void main(String[] args) {

        Pravougaonik p = new Pravougaonik();
        p.a = 5;
        p.b = 3;
    }
}
```



- private → samo u klasi
- protected → sve klase u paketu + sve nasleđene klase u drugom paketu
- public → sve klase
- default → sve klase u paketu

*kad se ne navede