

Introduzione ad Eclipse

Corso di Metodi Avanzati di Programmazione
2016-2017

Docente: Dr. Annalisa Appice

Sommario

- Ambiente Eclipse

Java: ambiente Eclipse



L'ultima versione disponibile è Eclipse Neon (precedente versione Mars)
distribuito gratuitamente e scaricabile

<http://www.eclipse.org>

Sono disponibili le versioni per Windows, Linux, Solaris, Mac OSX

Eclipse: cosa include?

Eclipse SDK include:

- Eclipse Platform,
- Java development tools,
- Plug-in Development Environment,

Eclipse non include alcun Java runtime environment (JRE).

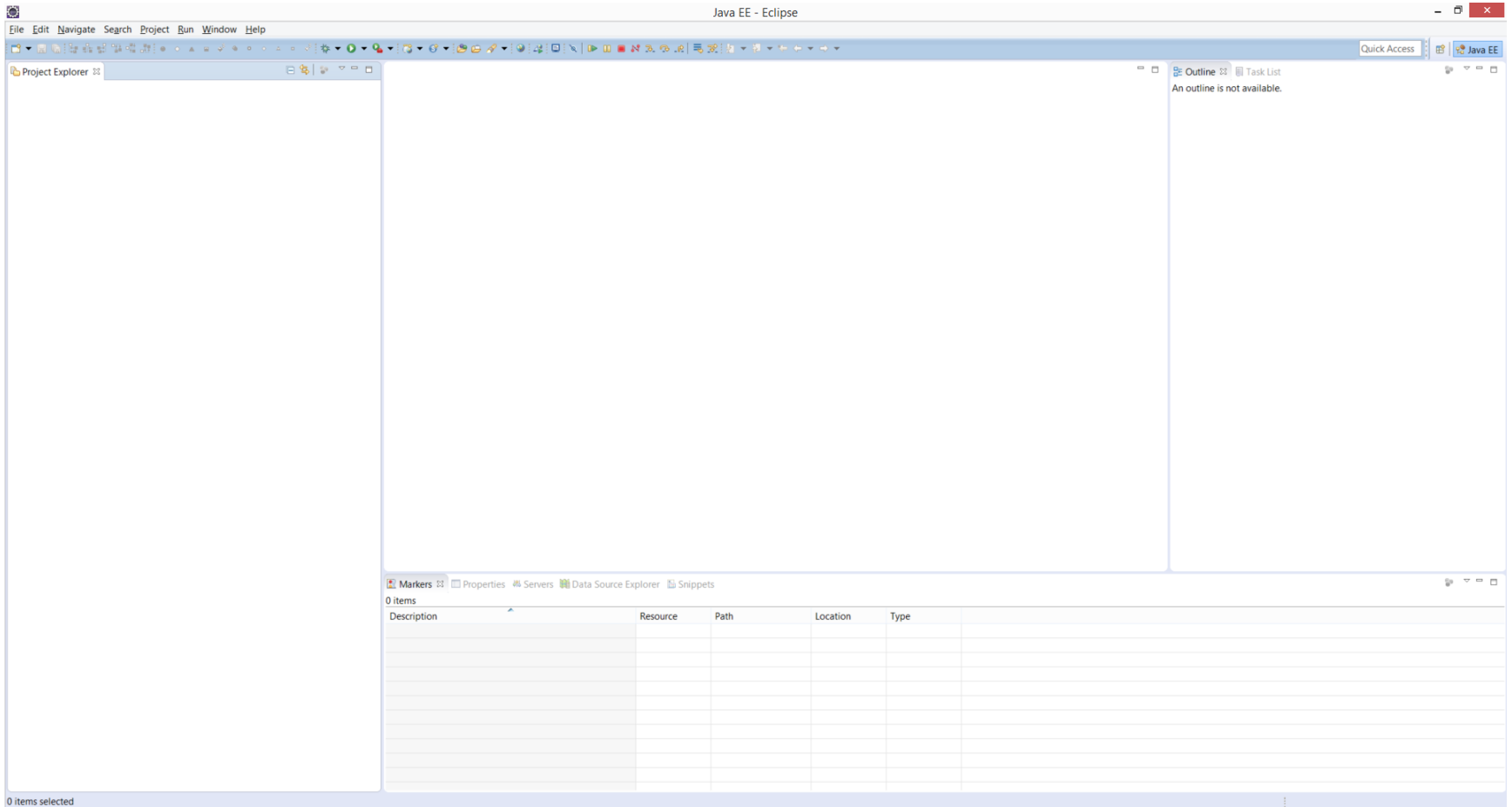
È necessario installare la Java runtime o Java development kit (JDK) in una versione uguale o superiore a 1.4.1. (1.8 in questo laboratorio)

Eclipse

- Creare un progetto
- Eseguire
- Generare la documentazione javadoc
- Debugging
- Creare un archivio .jar

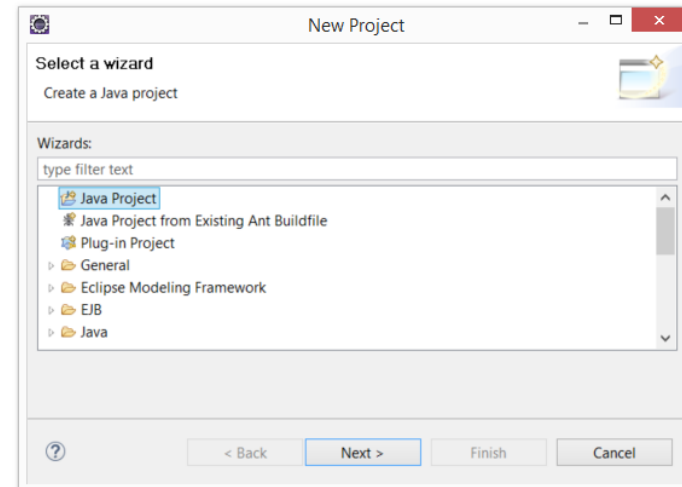
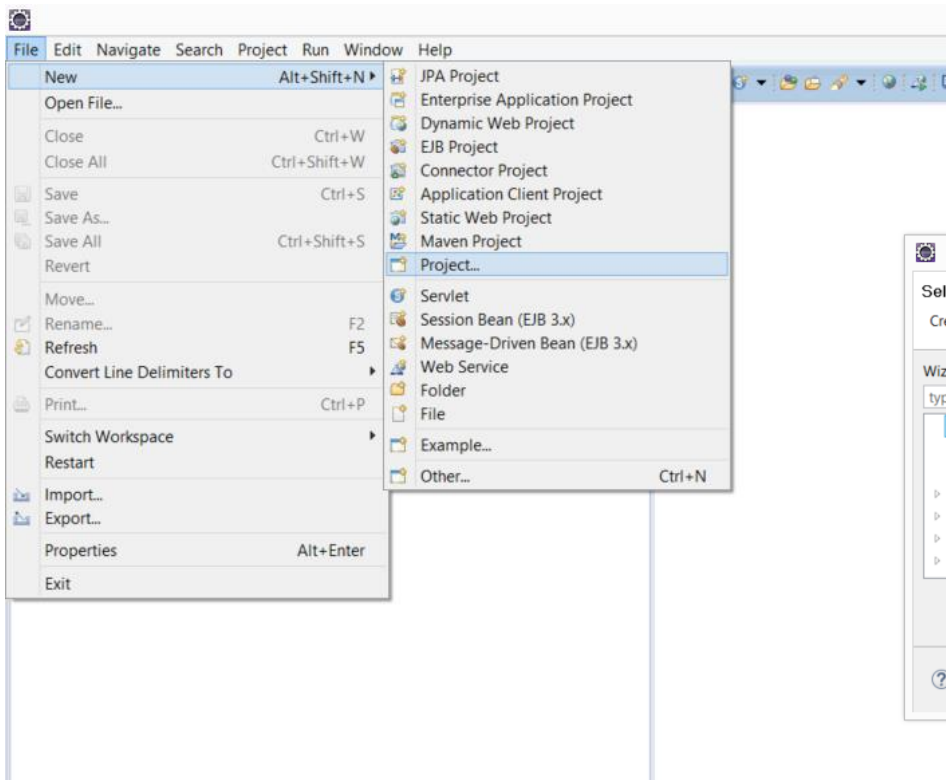
Eclipse: avviare l'ambiente

Avviare eclipse.exe



Eclipse: creare un progetto

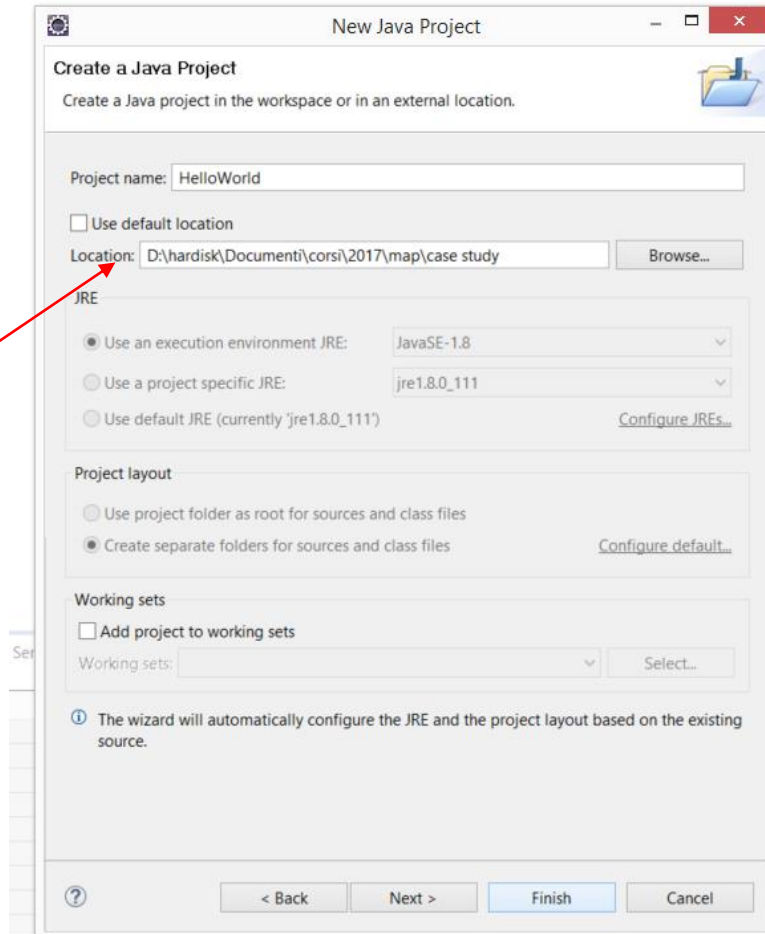
Creazione di un nuovo progetto del tipo Java Project (da Menu principale selezionare File → New → Project)



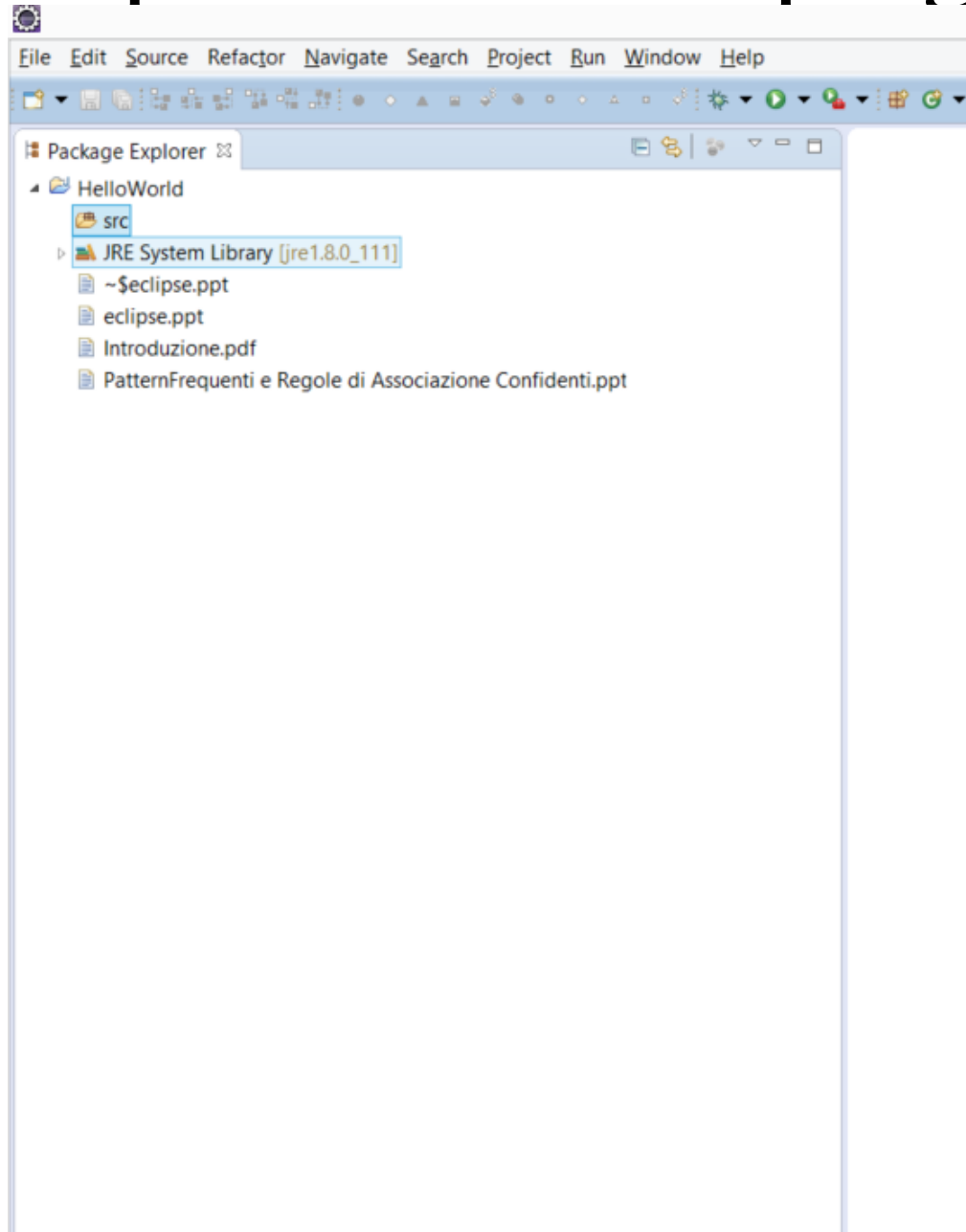
Eclipse: creare un progetto

Per default il progetto è salvato nella stessa directory dove è definite la workspace

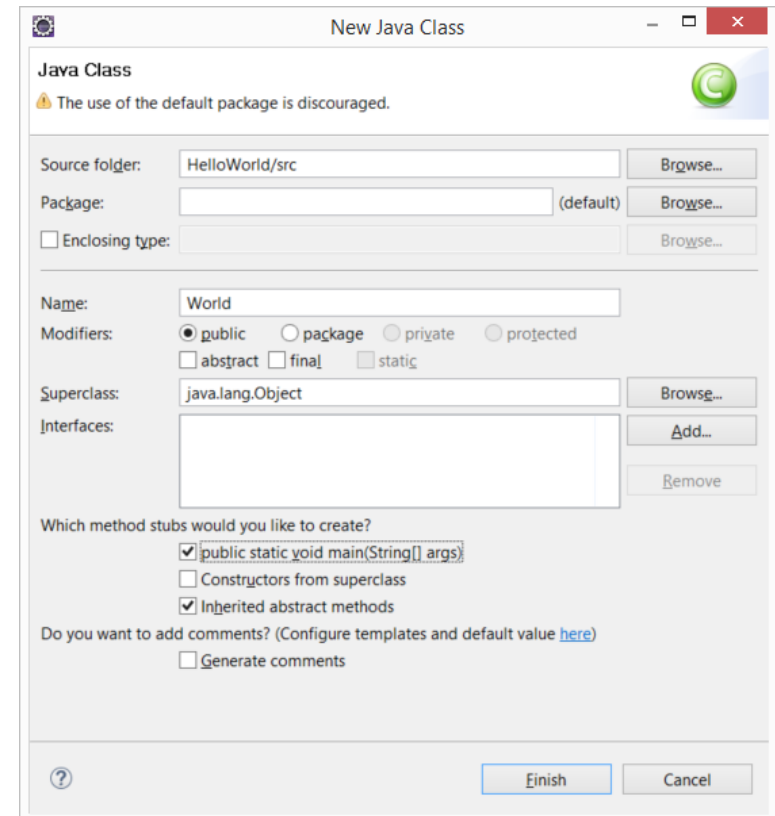
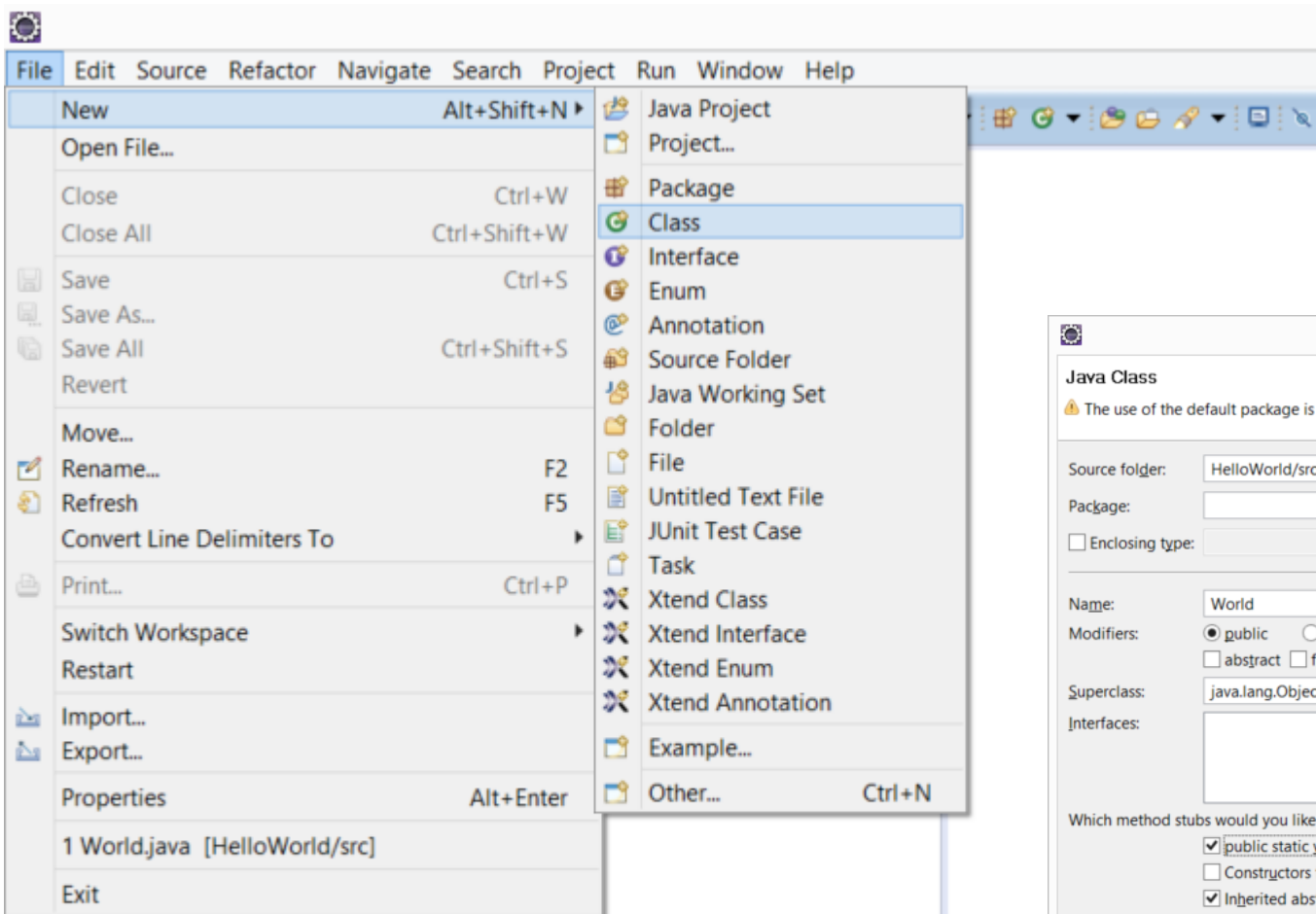
In alternativa va specificata la directory di riferimento



Eclipse: creare un progetto



Eclipse: aggiungere una classe

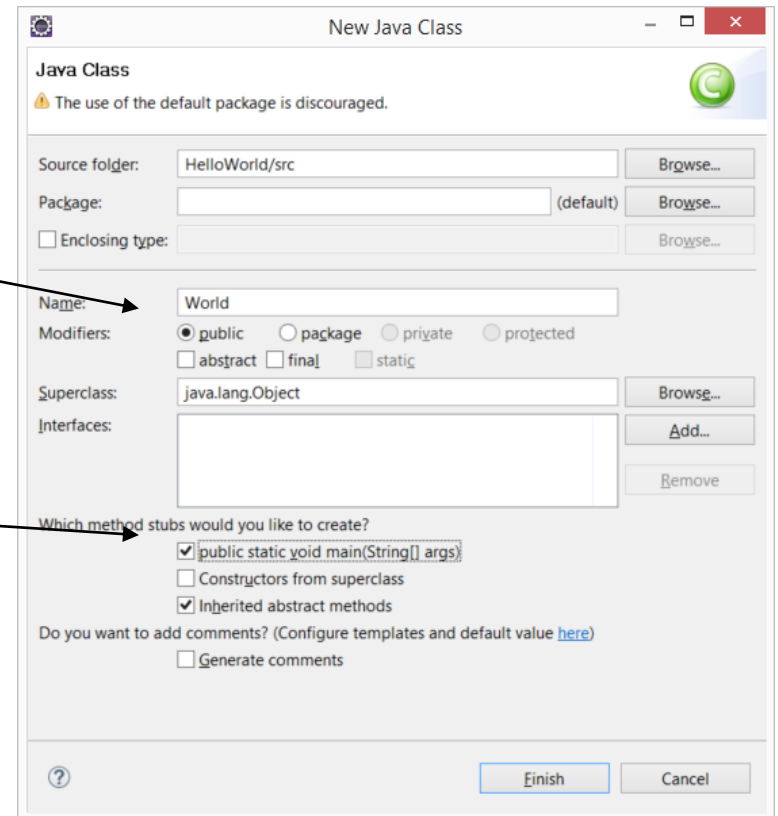


Eclipse: aggiungere una classe

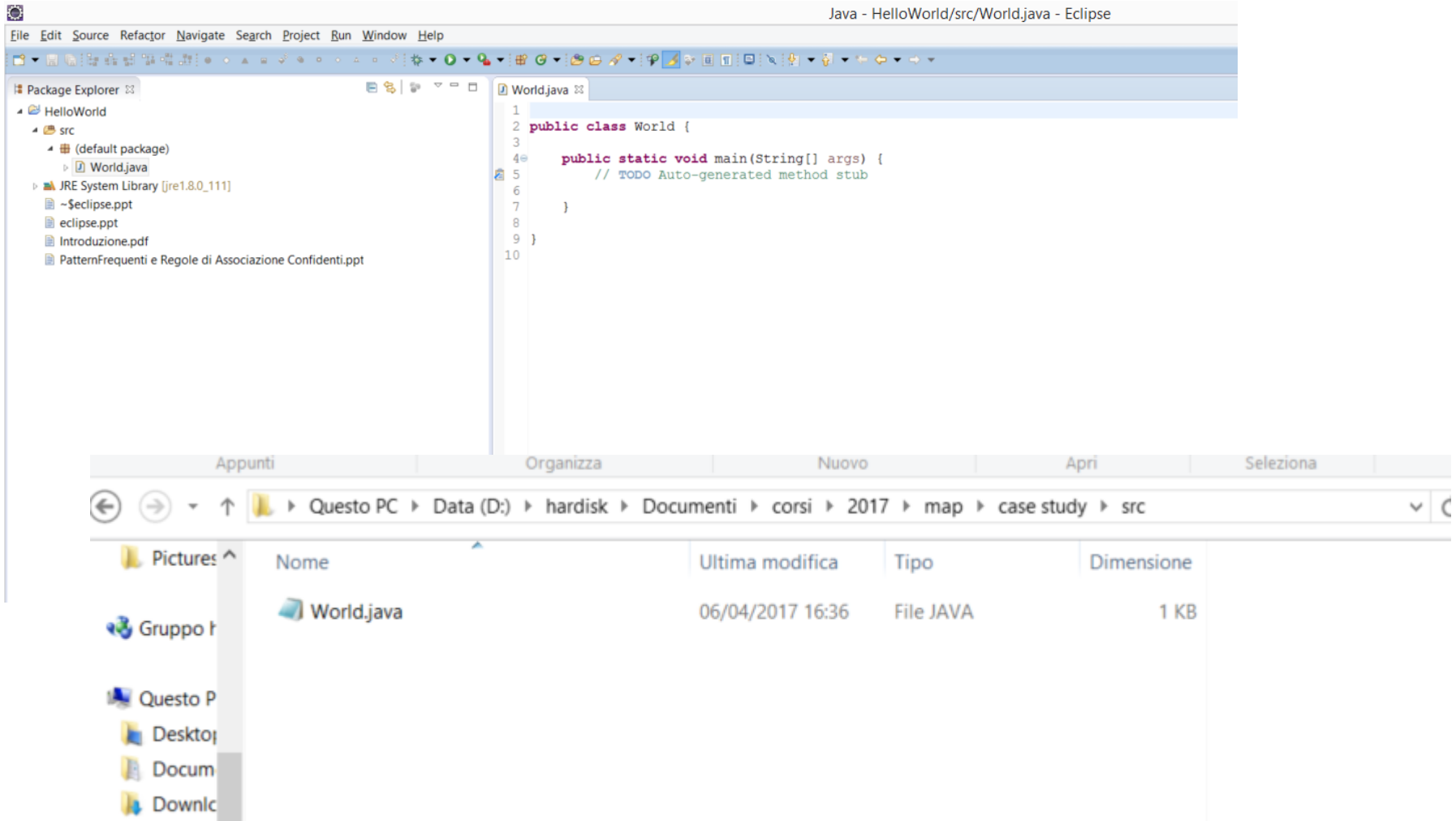
Creazione di una classe

Creazione della classe con
visibilità public World
all'interno del file World.java

La classe contiene un
metodo main

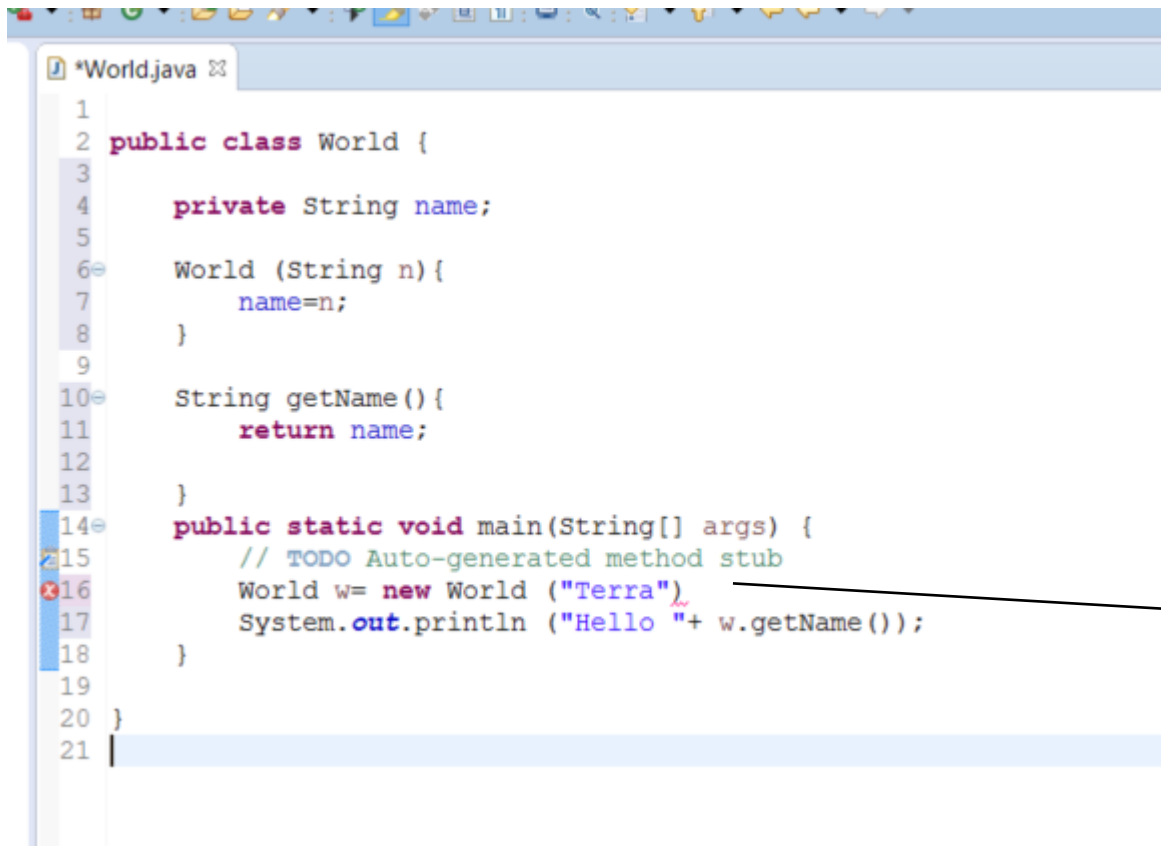


Eclipse: aggiungere una classe



Eclipse: errori di sintassi

Che succede se si introducono semplici errori di sintassi?



```
1
2 public class World {
3
4     private String name;
5
6     World (String n){
7         name=n;
8     }
9
10    String getName(){
11        return name;
12    }
13
14    public static void main(String[] args) {
15        // TODO Auto-generated method stub
16        World w= new World ("Terra");
17        System.out.println ("Hello "+ w.getName());
18    }
19
20 }
21
```

Manca il “.”

Eclipse: errori di sintassi

L'editor visualizza la presenza di un errore sintattico

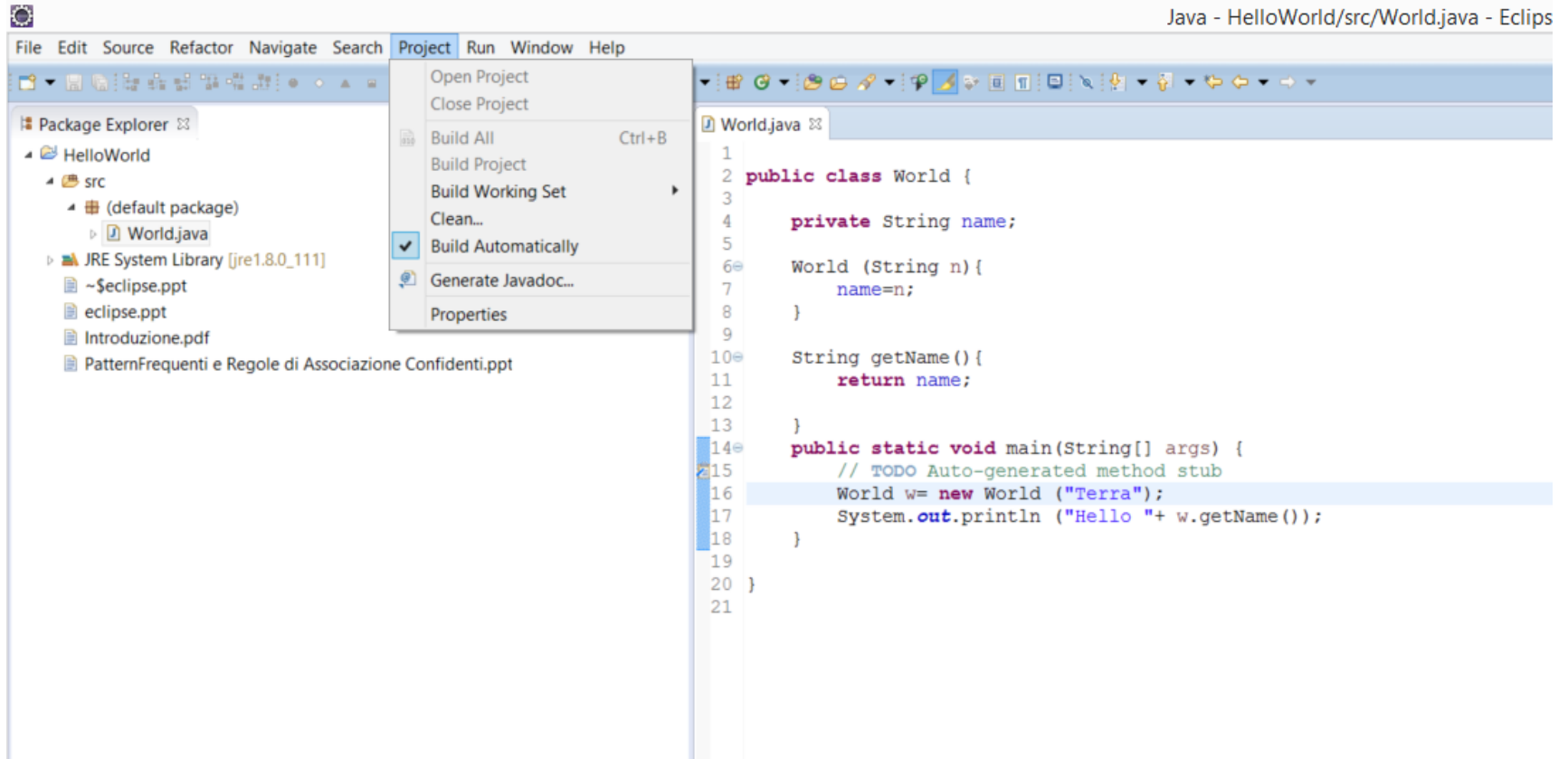


```
1
2 public class World {
3
4     private String name;
5
6     World (String n){
7         name=n;
8     }
9
10    String getName(){
11        return name;
12    }
13
14    public static void main(String[] args) {
15        // TODO Auto-generated method stub
16        System.out.println ("Hello " + w.getName());
17    }
18
19
20
21
```

L'ambiente riconosce l'assenza del “;”

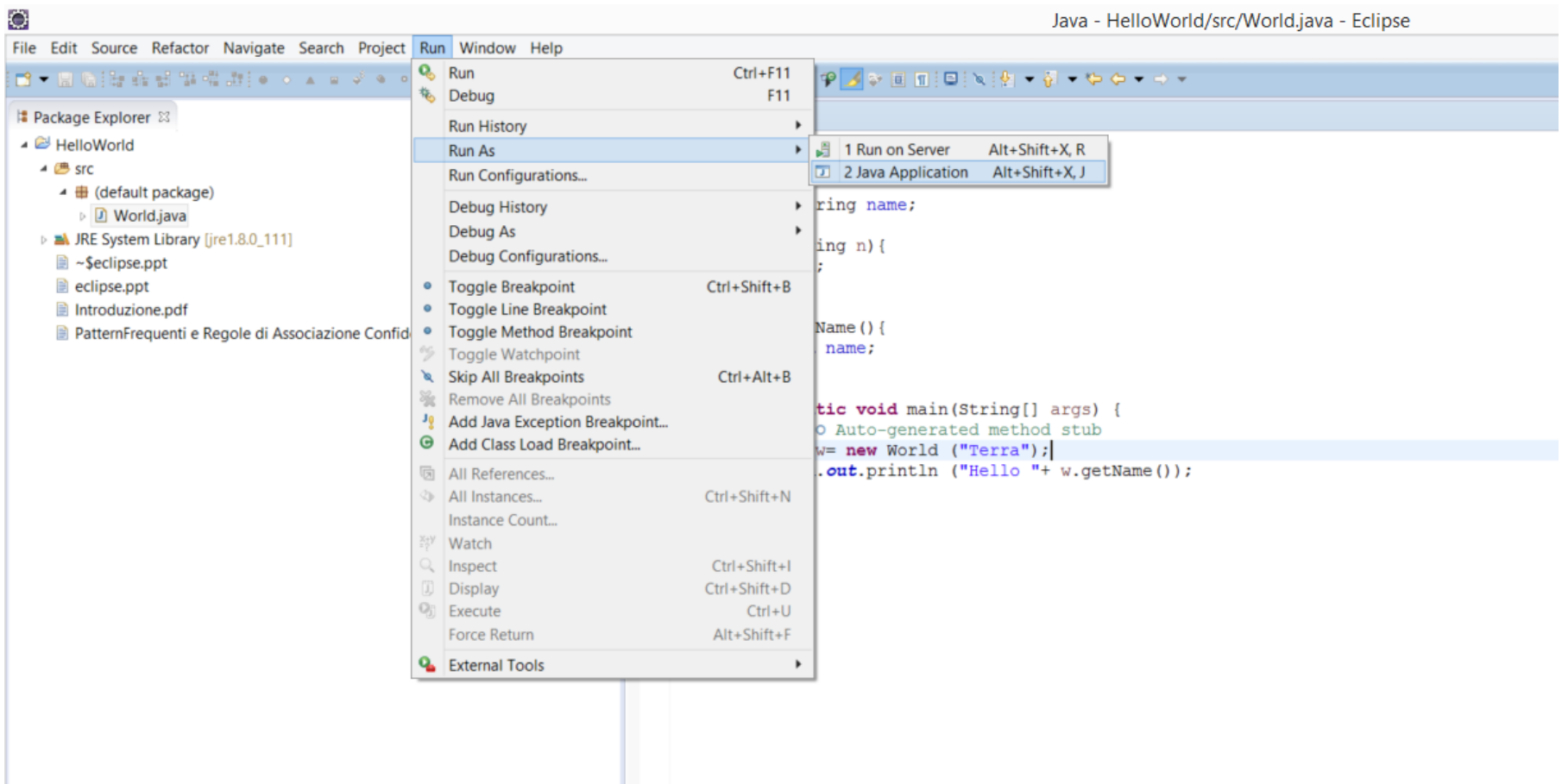
Eclipse: eseguire una applicazione

Quando si crea un progetto l'opzione "Build Automatically" è abilitata di default.

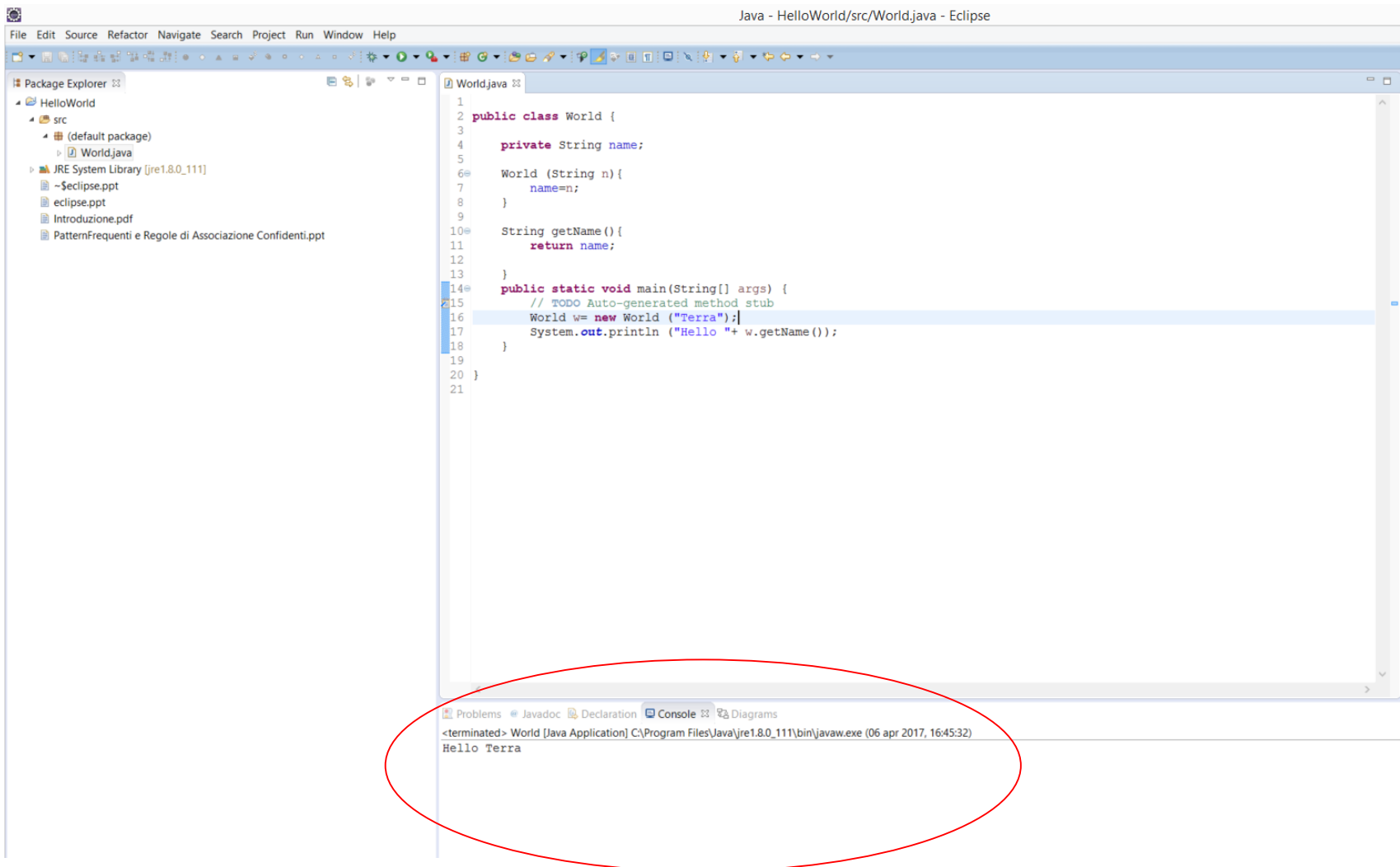


Eclipse: eseguire una applicazione

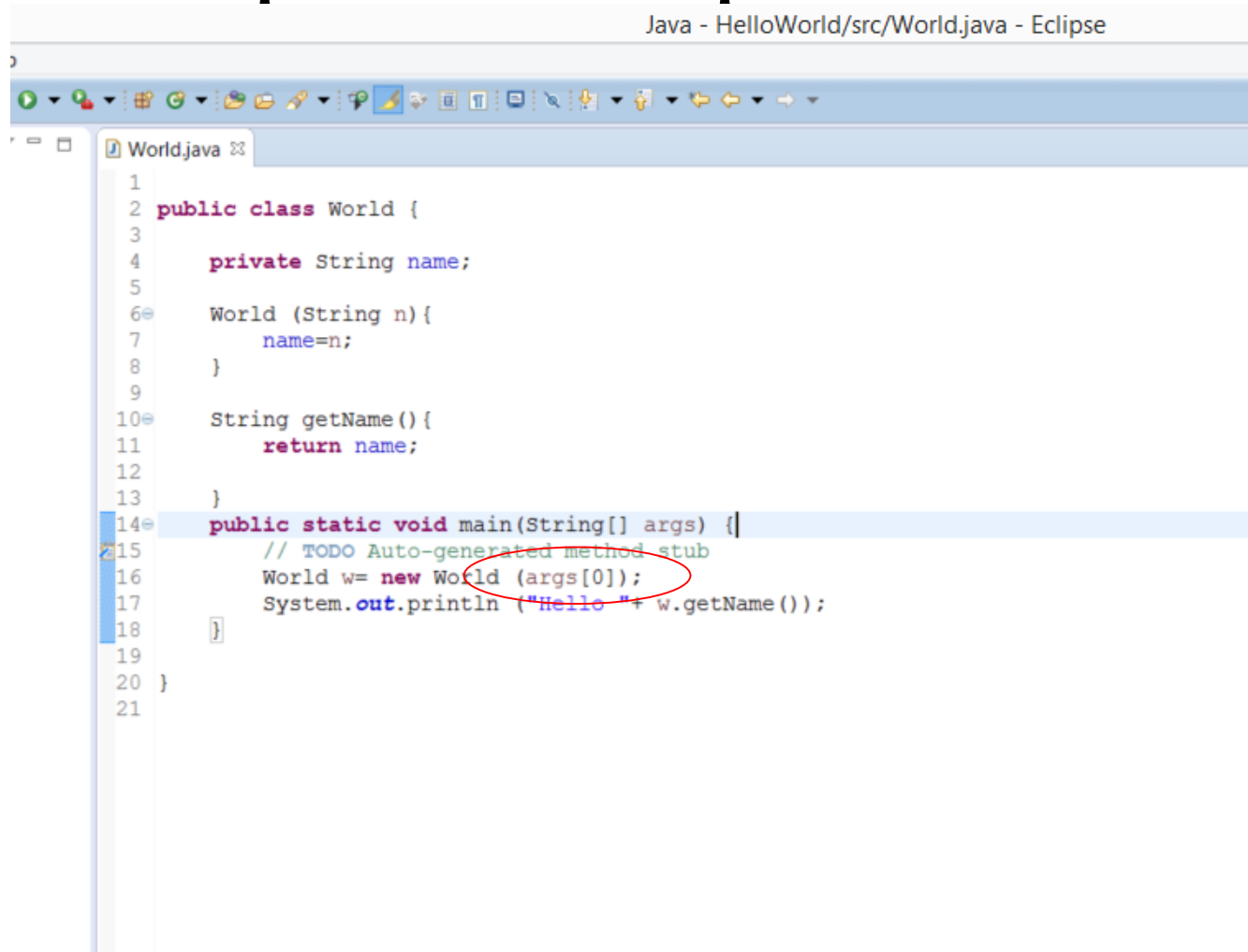
Selezionare la voce “Run”



Eclipse: eseguire una applicazione



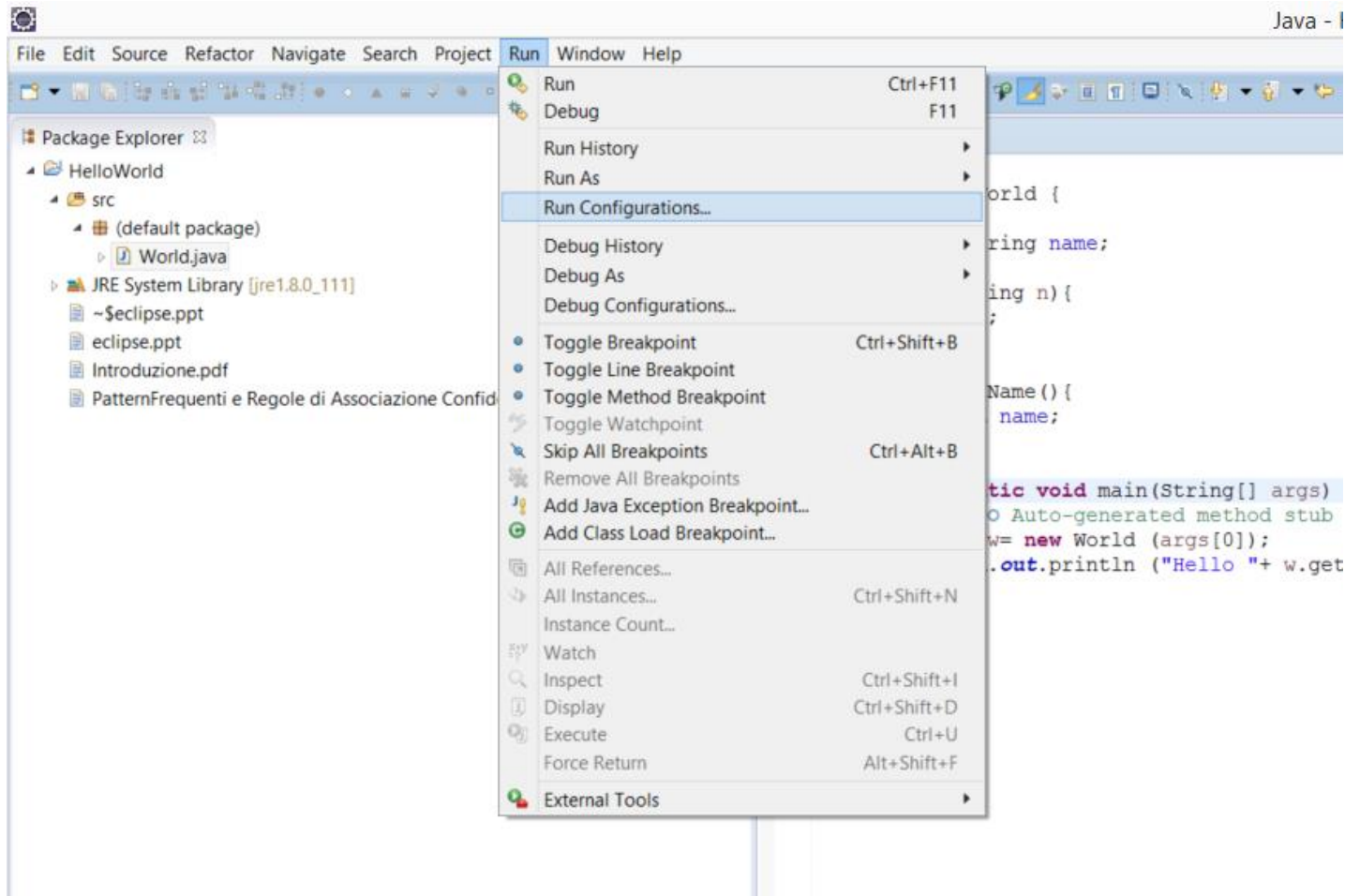
Eclipse: eseguire una applicazione e specificare parametri



```
Java - HelloWorld/src/World.java - Eclipse

World.java
1
2 public class World {
3
4     private String name;
5
6     World (String n){
7         name=n;
8     }
9
10    String getName(){
11        return name;
12    }
13
14    public static void main(String[] args) {
15        // TODO Auto-generated method stub
16        World w= new World (args[0]);
17        System.out.println ("Hello " + w.getName());
18    }
19
20 }
21
```

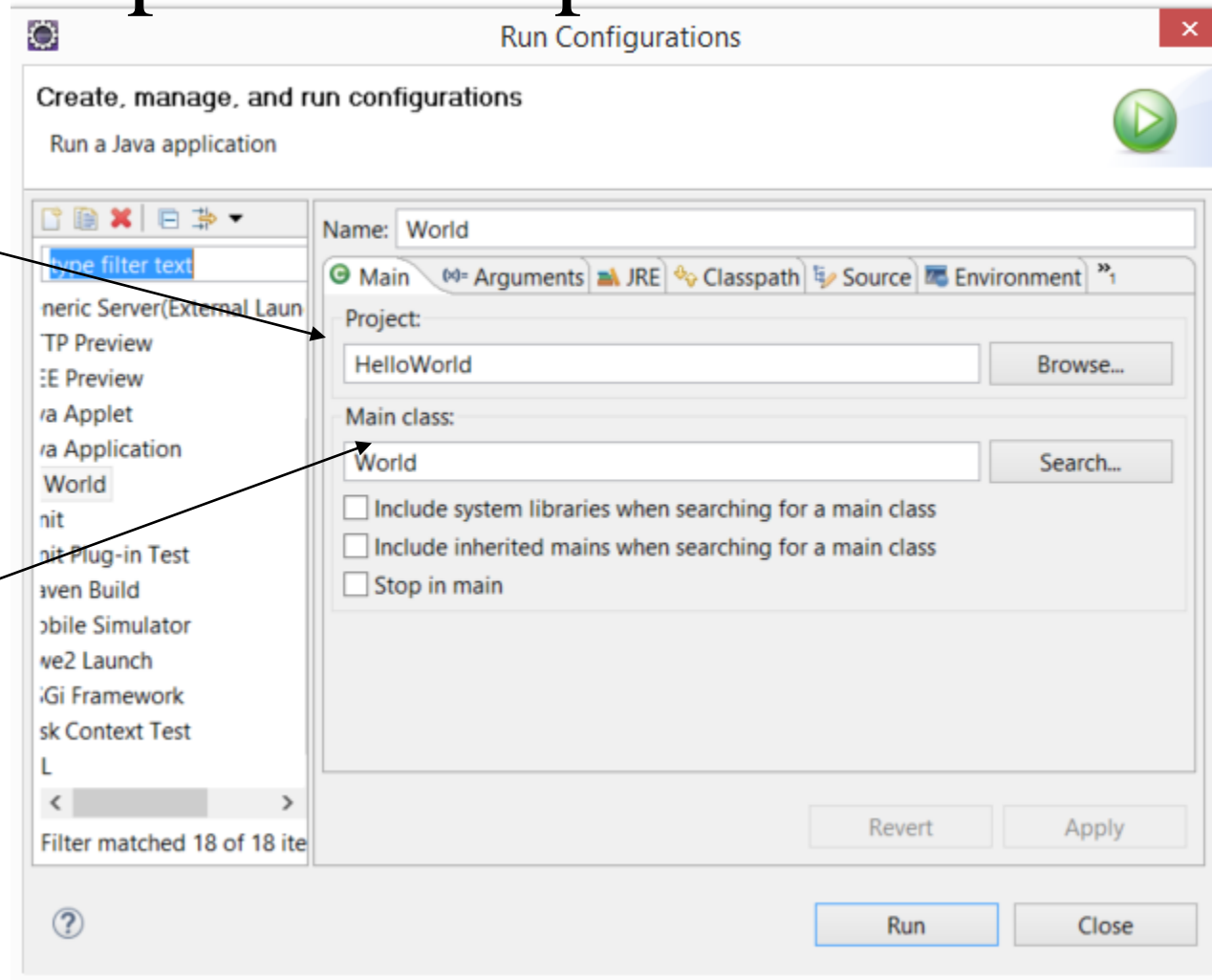
Eclipse: eseguire una applicazione e specificare parametri



Eclipse: eseguire una applicazione e specificare parametri

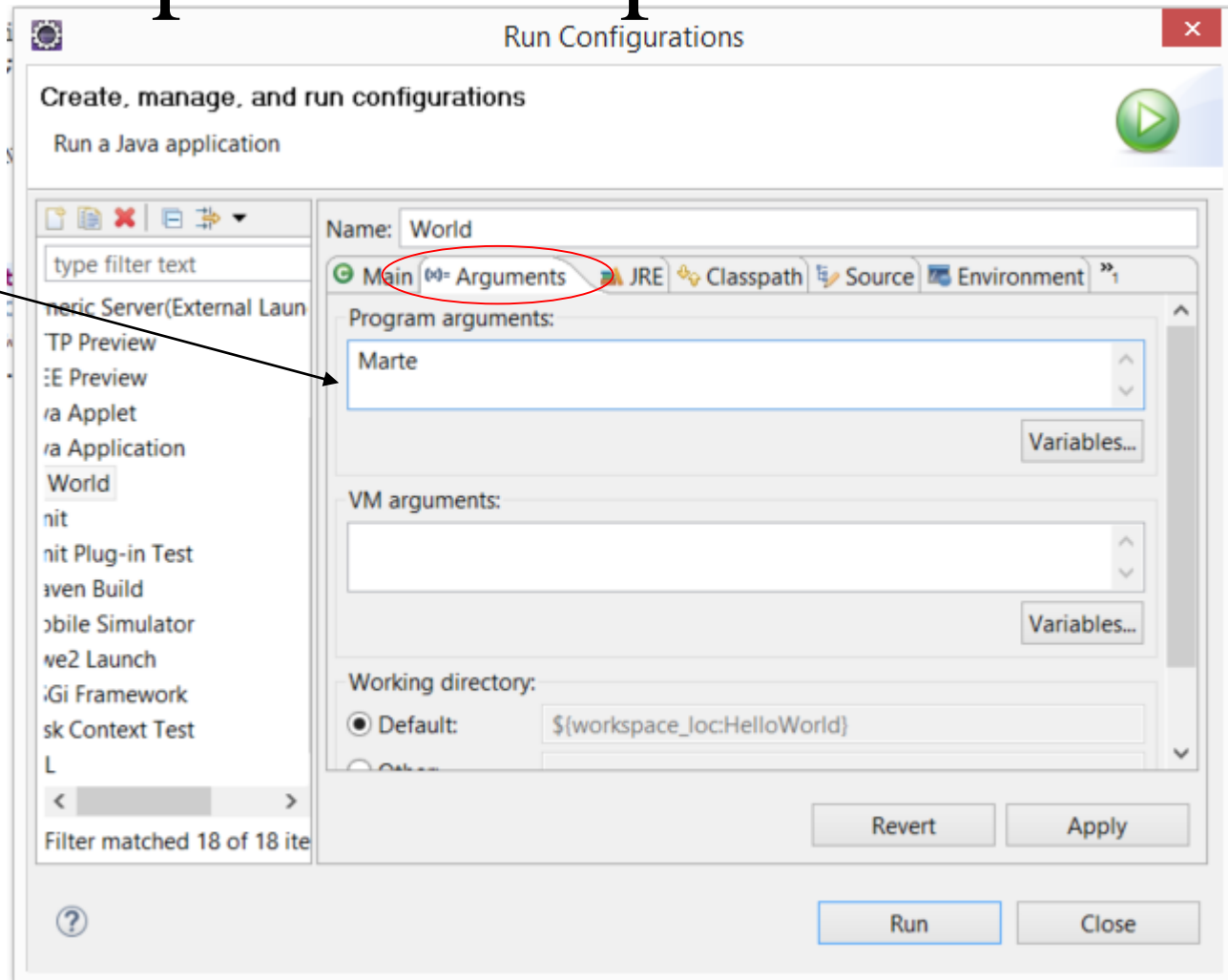
Nome del
progetto

Main class

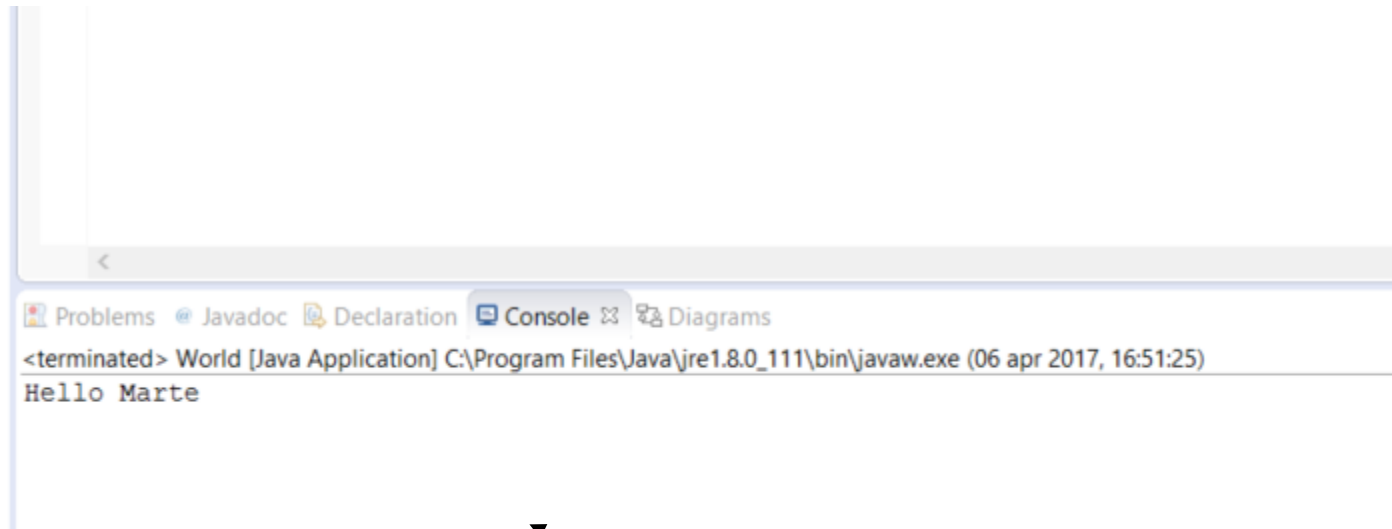


Eclipse: eseguire una applicazione e specificare parametri

argomenti



Eclipse:eseguire una applicazione



Console

Eclipse:javadoc

Javadoc è usato per realizzare una documentazione della applicazione fruibile all'utente

1. Descrizione della classe

```
/**
```

```
* <p>Title: HelloWorld</p>
```

```
* <p>Description: esempio di uso dell'ambiente ECLIPSE </p>
```

```
* <p>Copyright: Copyright (c) 2017</p>
```

```
* <p>Company: Dipartimento di Informatica, Università degli studi di Bari</p>
```

```
* <p>Class description: hello world
```

```
* Hello World per gli studenti di Metodi Vvanzati di Programmazione.
```

```
* @author Metodi Avanzati di Programmazione Group
```

```
* @version 1.0
```

```
*/
```

Eclipse:javadoc

2. Descrizione di un metodo

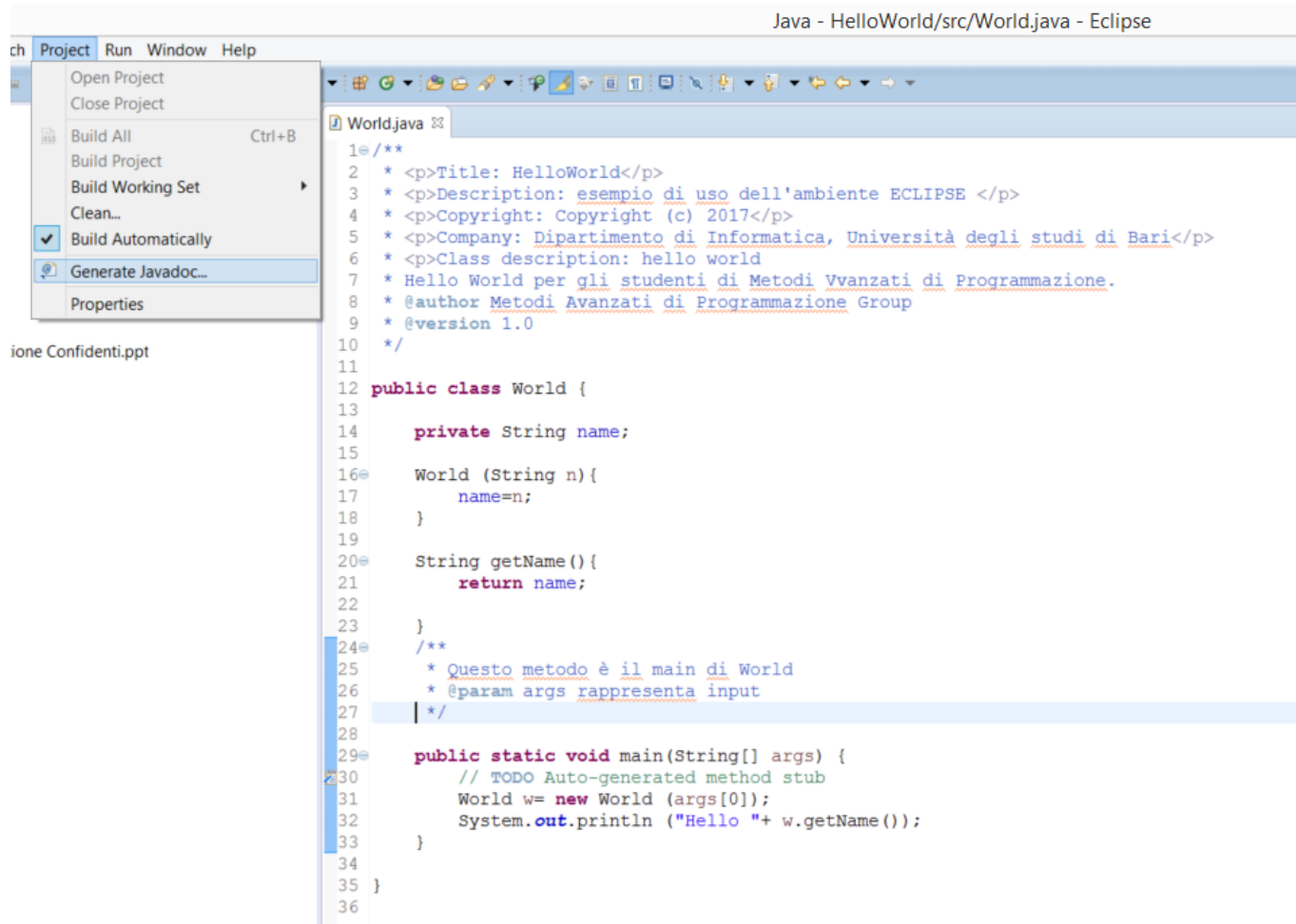
```
/**  
 * Questo metodo è il main di Hello World  
 * @param args rappresenta input  
 */
```

allo scopo di descrivere parametri e obiettivo

Eclipse: generare javadoc

```
World.java
1 /**
2  * <p>Title: HelloWorld</p>
3  * <p>Description: esempio di uso dell'ambiente ECLIPSE </p>
4  * <p>Copyright: Copyright (c) 2017</p>
5  * <p>Company: Dipartimento di Informatica, Università degli studi di Bari</p>
6  * <p>Class description: hello world
7  * Hello World per gli studenti di Metodi Vanzati di Programmazione.
8  * @author Metodi Avanzati di Programmazione Group
9  * @version 1.0
10 */
11
12 public class World {
13
14     private String name;
15
16     World (String n){
17         name=n;
18     }
19
20     String getName(){
21         return name;
22     }
23
24     /**
25      * Questo metodo è il main di World
26      * @param args rappresenta input
27      */
28
29     public static void main(String[] args) {
30         // TODO Auto-generated method stub
31         World w= new World (args[0]);
32         System.out.println ("Hello " + w.getName());
33     }
34 }
35
36
```

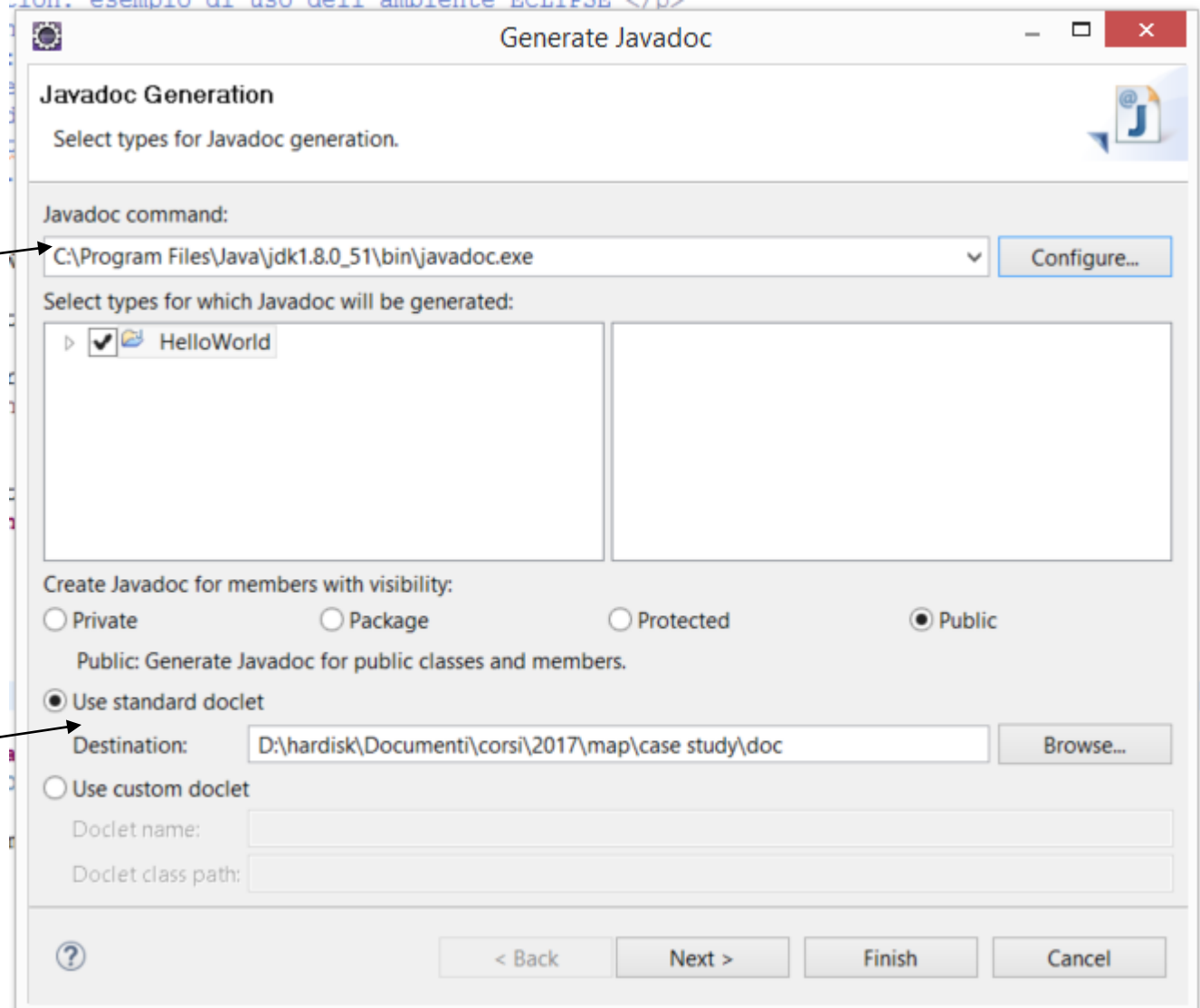
Eclipse: generare javadoc



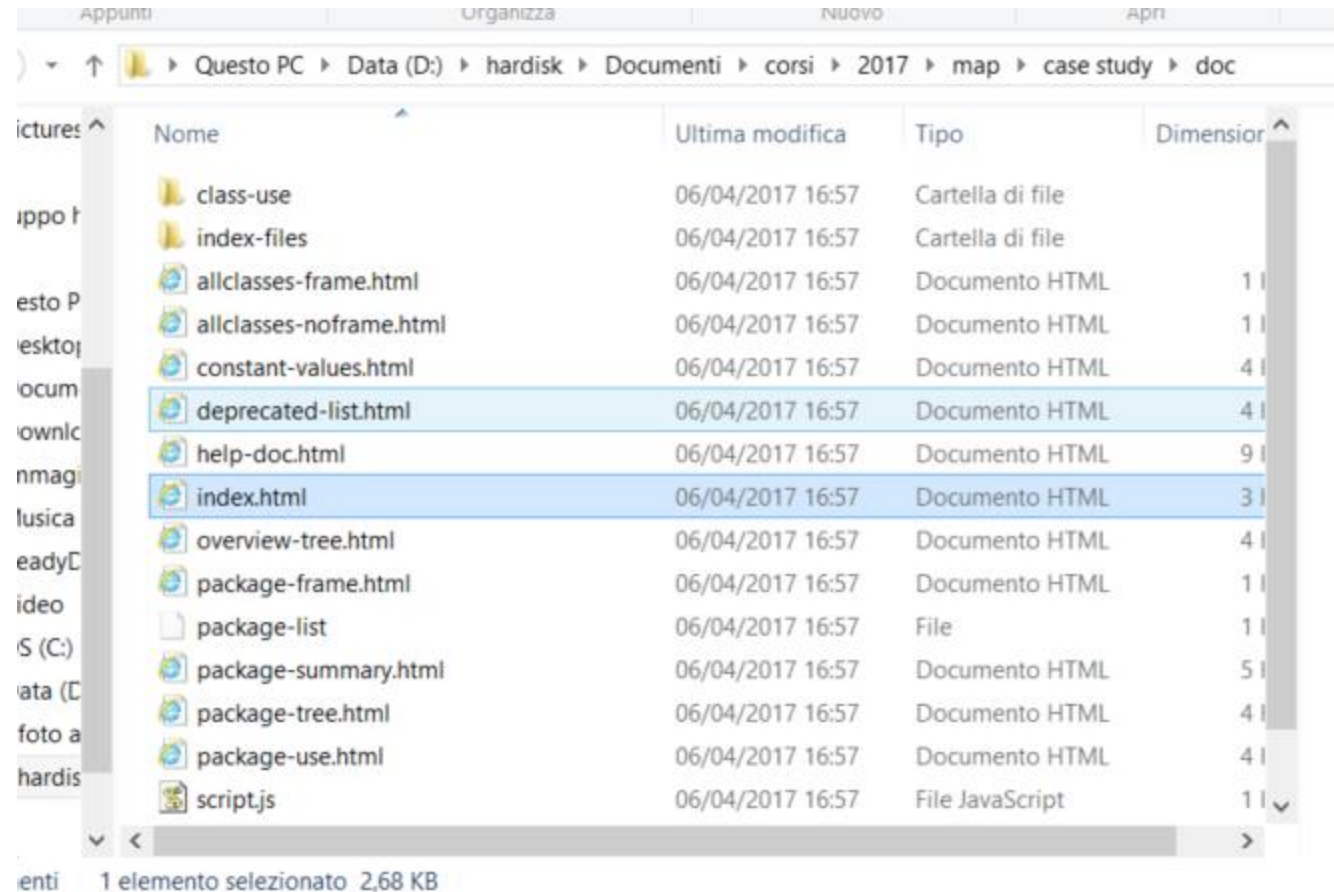
Eclipse: generare javadoc

Selezionare
javadoc.exe da
jdk

Selezionare dove
salvare java doc



Eclipse: generare javadoc



Eclipse: generare javadoc

The screenshot displays the Eclipse IDE interface with the 'Generated Documentation' window open. The browser tabs at the top include 'D:\hardisk\Documen', 'Apriori algorithm - Wikipedia', 'Eclipse Downloads', and 'Generated Documentation (... x)'. The address bar shows a Google search bar with the text 'Effettua la ricerca' and 'Condividi'.

The left sidebar shows the 'All Classes' view with 'World' selected. The main content area displays the Javadoc for the 'World' class, which is a subclass of 'java.lang.Object'. The class includes a title 'HelloWorld', a description 'esempio di uso dell'ambiente ECLIPSE', and a copyright notice 'Copyright (c) 2017'. The company is listed as 'Dipartimento di Informatica, Università degli studi di Bari'.

The 'Method Summary' section shows a table of methods:

Modifier and Type	Method and Description
static void	main(java.lang.String[] args) Questo metodo è il main di World

Below the table, it lists 'Methods inherited from class java.lang.Object': equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait.

The 'Method Detail' section provides the signature and description for the 'main' method:

```
public static void main(java.lang.String[] args)
```

Questo metodo è il main di World

Parameters:
args - rappresenta input

```

/**
 * <p>Title: HelloWorld</p>
 * <p>Description: esempio di uso dell'ambiente ECLIPSE </p>
 * <p>Copyright: Copyright (c) 2017</p>
 * <p>Company: Dipartimento di Informatica, Università degli studi
 * <p>Class description: hello world
 * Hello World per gli studenti di Metodi Vvanzati di Programmazio
 * @author Metodi Avanzati di Programmazione Group
 * @version 1.0
 */

public class World {

    private String name;

    World (String n){
        name=n;
    }

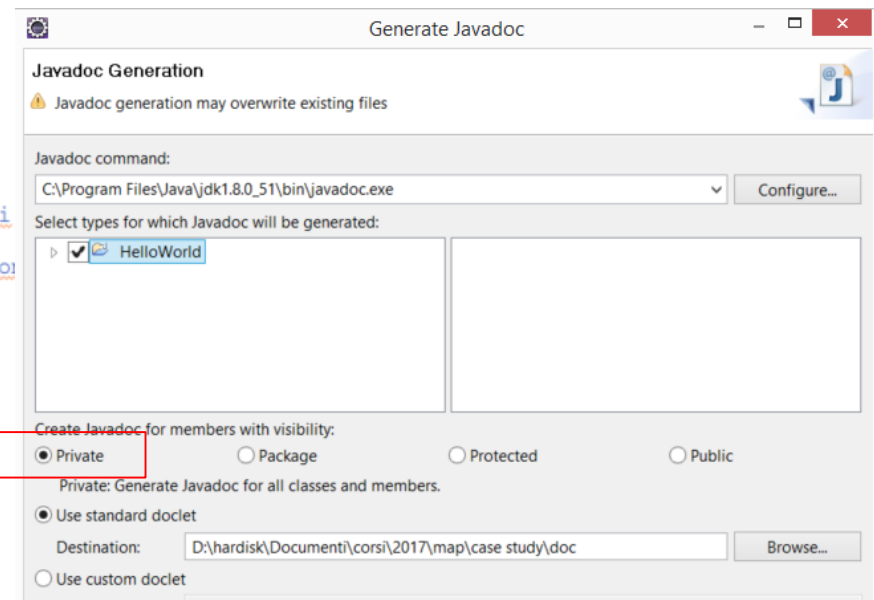
    /**
     * Questo metodo è il metodo che restituisce il nome
     * @return nome
     */

    private String getName(){
        return name;
    }

    /**
     * Questo metodo è il main di World
     * @param args rappresenta input
     */

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        World w= new World (args[0]);
        System.out.println ("Hello " + w.getName());
    }
}

```



World(java.lang.String n)

Method Summary

All Methods | Static Methods | Instance Methods | Concrete Methods

Modifier and Type	Method and Description
private java.lang.String	getName() Questo metodo è il metodo che restituisce il nome
static void	main(java.lang.String[] args) Questo metodo è il main di World

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

name

private java.lang.String name

Constructor Detail

Eclipse debugging

- Per inserire/rimuovere un breakpoint:
 - Selezionare il punto di interesse
 - Selezionare “Toggle Line Breakpoint” da “Run”
- Per eseguire Debug
 - Lanciare tramite “Debug” di “Run”
 - F5 = Step into
 - F6 = Step Over
 - Ctrl+R = per spostarsi nel punto dove è posizionato il cursore

Eclipse:debugging

Debug - HelloWorld/src/World.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Debug World [Java Application]
World at localhost:56009
Thread [main] (Suspended (breakpoint at line 37 in World))
World.main(String[]) line: 37
C:\Program Files\Java\jre1.8.0_111\bin\javaw.exe (06 apr 2017, 17:07:39)

World.java

```
21 * Questo metodo è il metodo che restituisce il nome
22 * @return nome
23 */
24
25 private String getName(){
26     return name;
27 }
28
29 /**
30 * Questo metodo è il main di World
31 * @param args rappresenta input
32 */
33
34 public static void main(String[] args) {
35     // TODO Auto-generated method stub
36     World w= new World (args[0]);
37     System.out.println ("Hello "+ w.getName());
38 }
39
40 }
41
```

Variables Breakpoints

Name	Value
args	String[1] (id=16)
w	World (id=17)
name	"Marte" (id=21)
hash	0
value	(id=25)

Marte

Outline

- World
 - name : String
 - World(String)
 - getName() : String
 - main(String[]) : void

Console

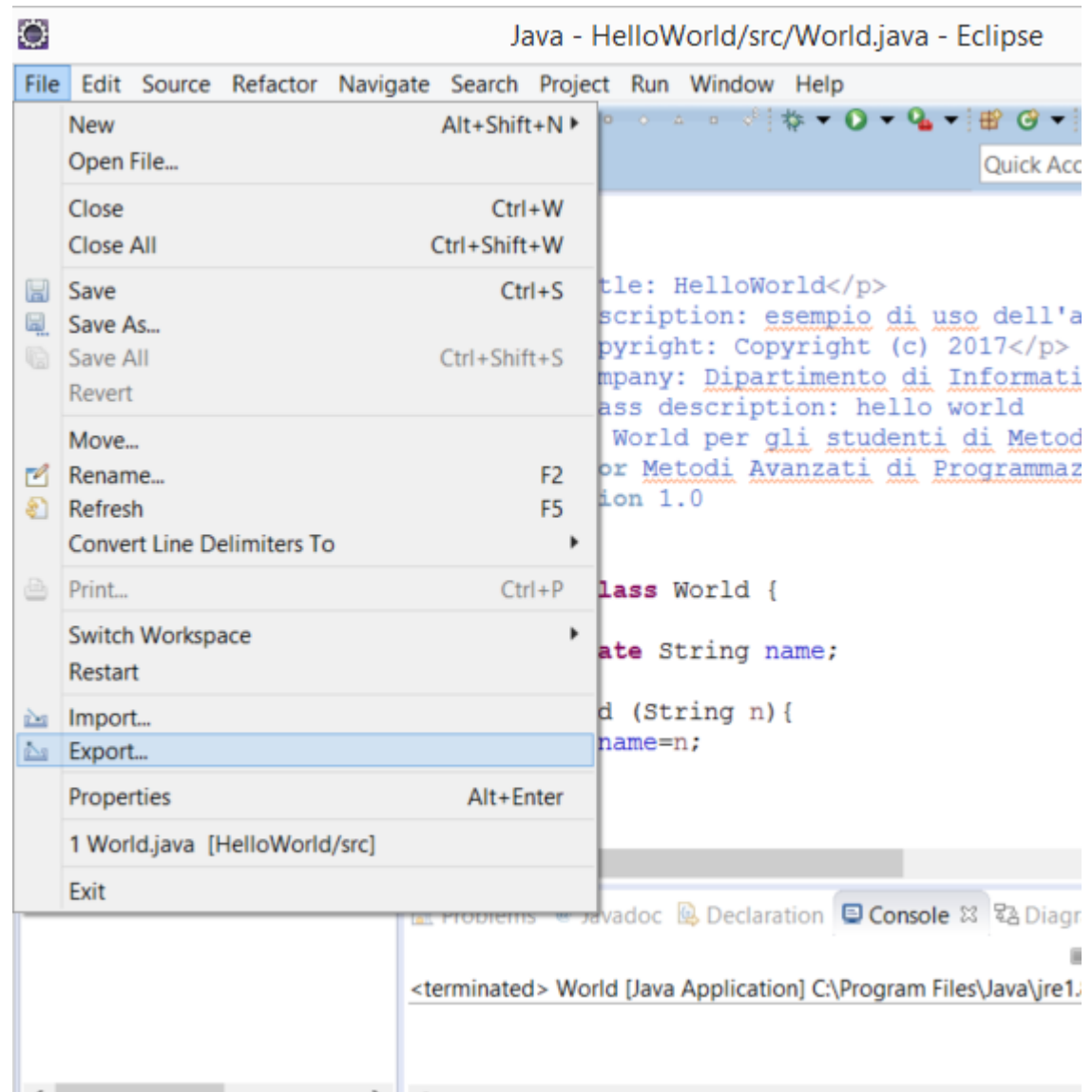
World [Java Application] C:\Program Files\Java\jre1.8.0_111\bin\javaw.exe (06 apr 2017, 17:07:39)

Stato delle
variabili

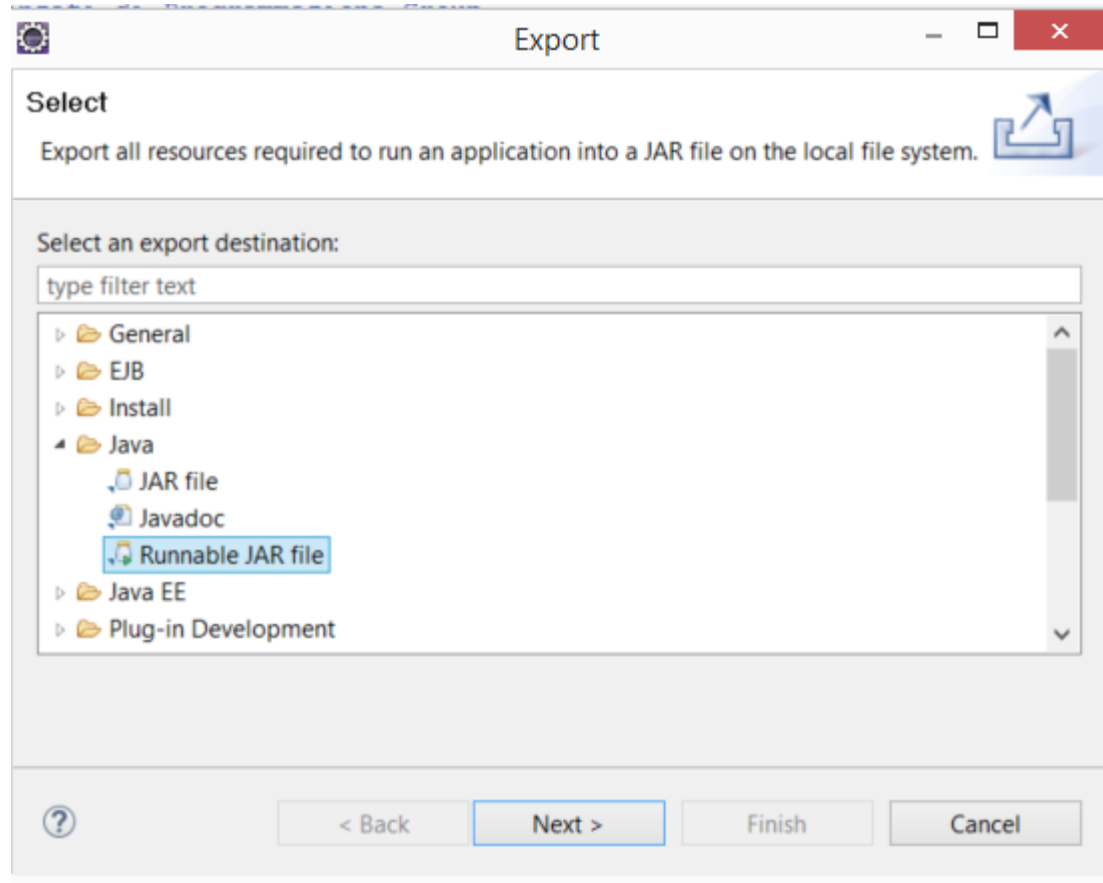
breakpoint

Eclipse: creare un archivio .jar

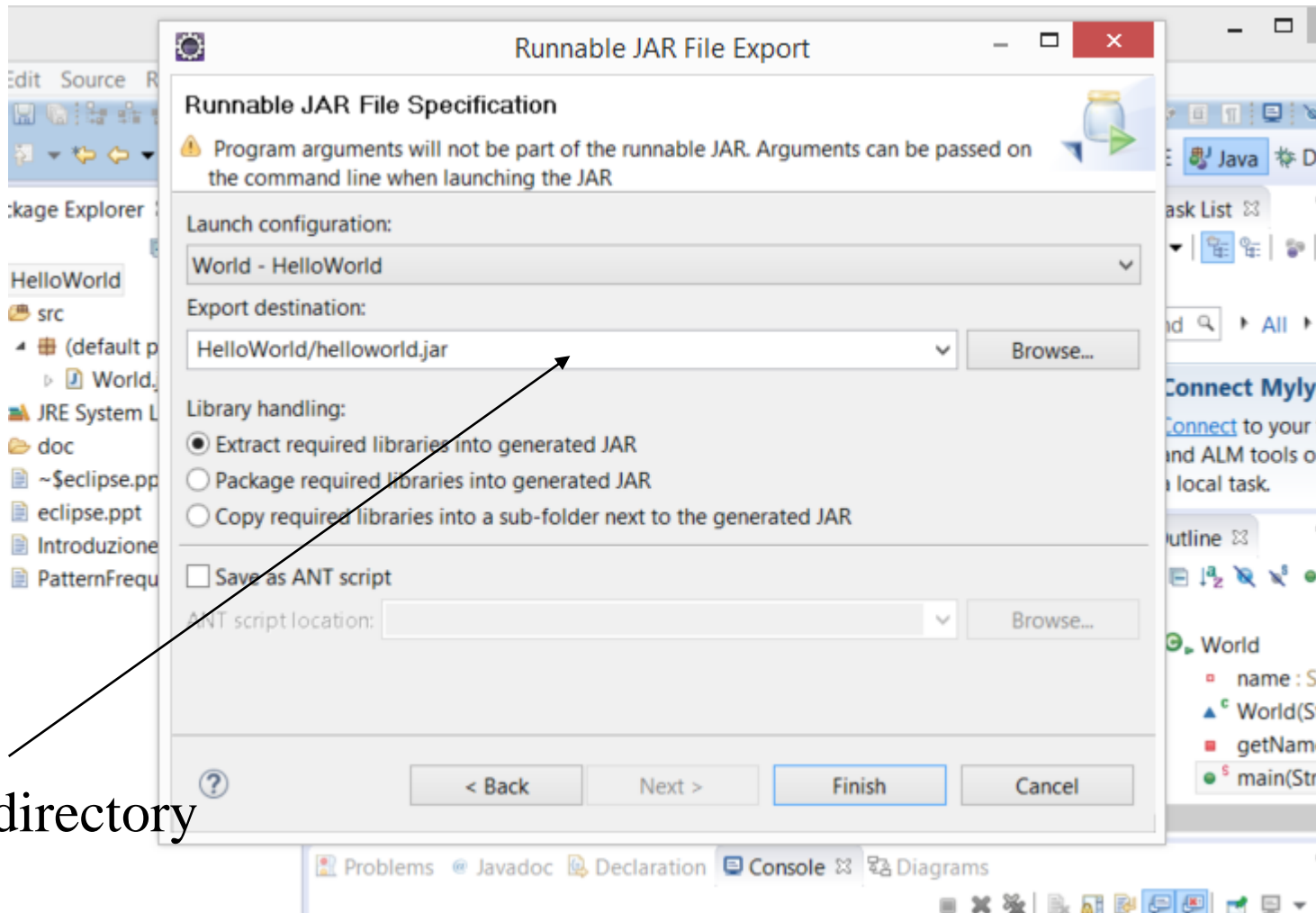
Selezionare
“Export” da
menu “File”



Eclipse: creare un archivio .jar

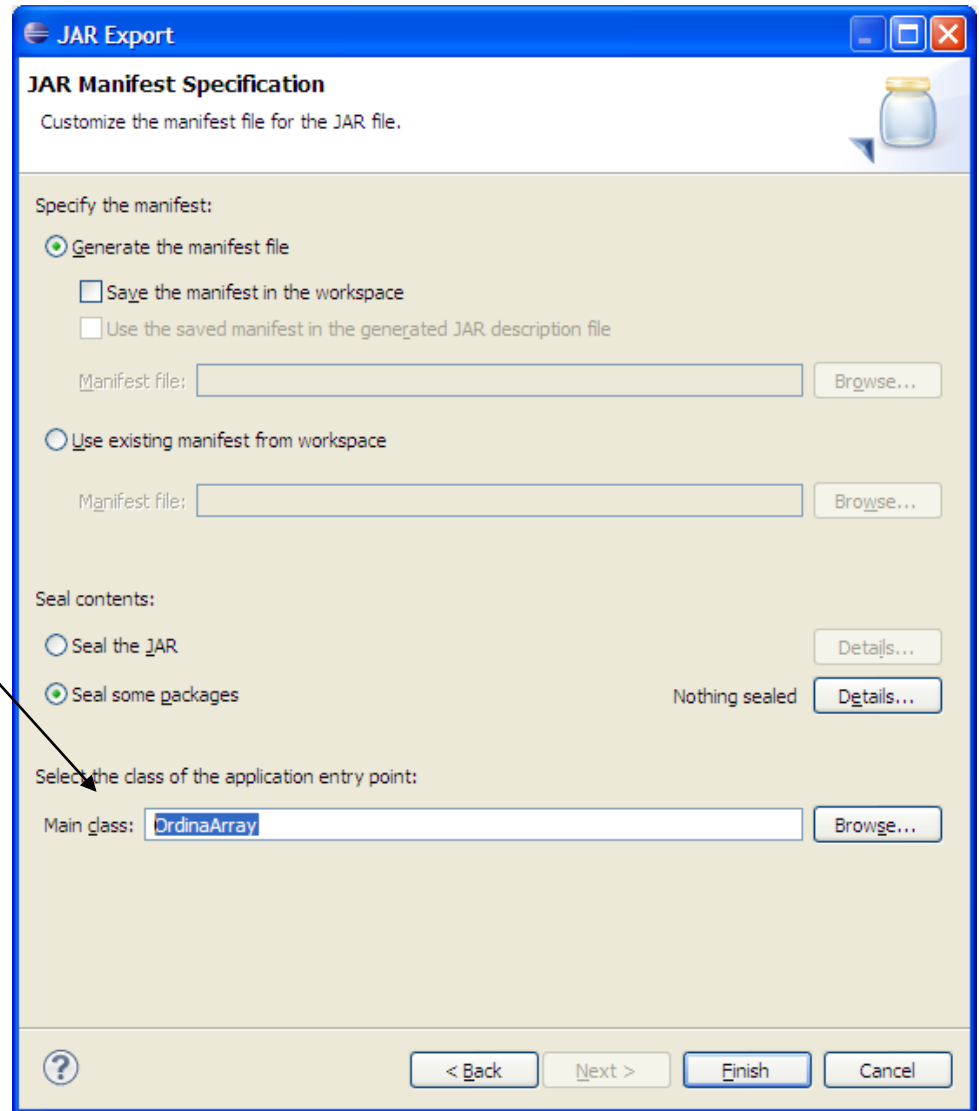


Eclipse: creare un archivio .jar

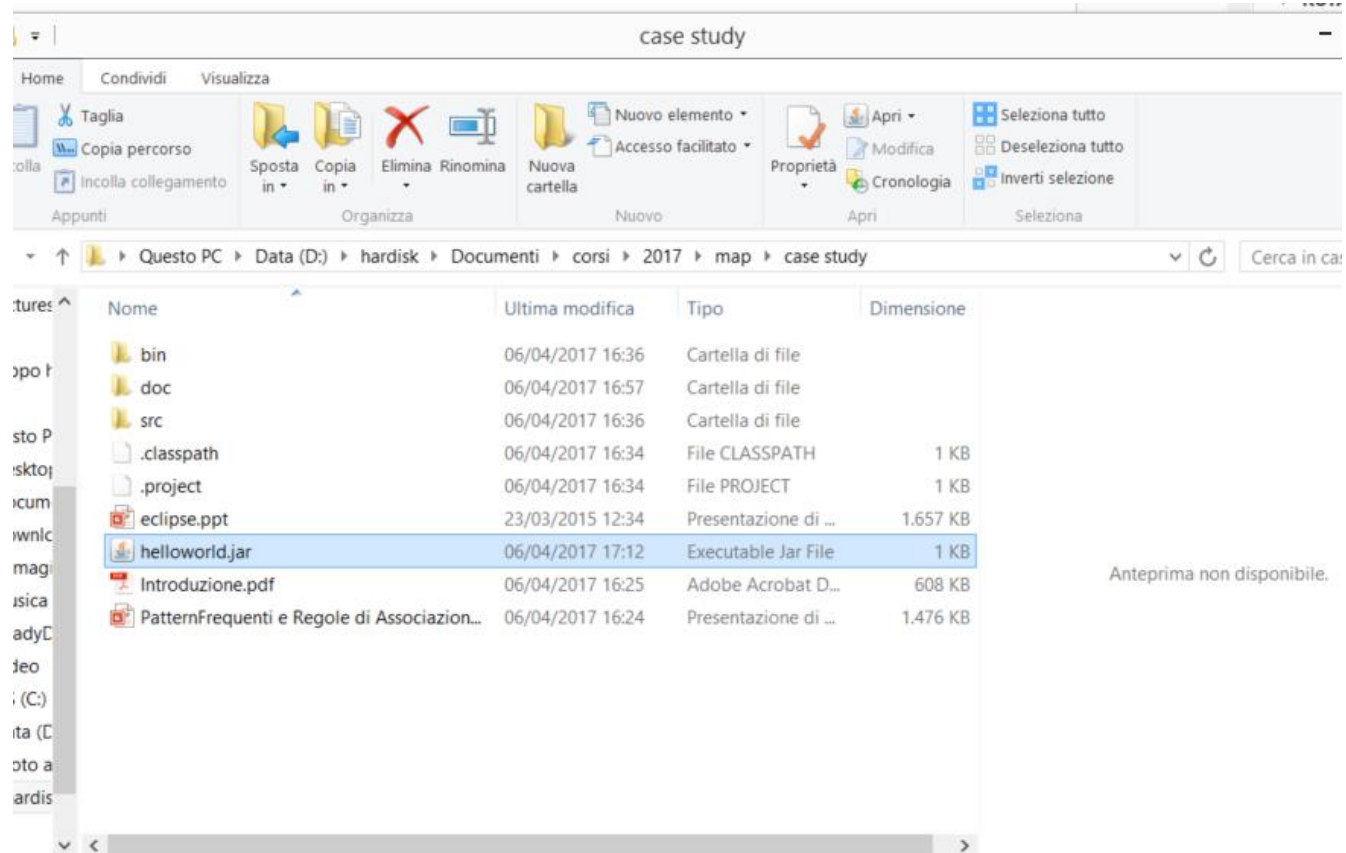


Eclipse: creare un archivio .jar


Main class



Esempio di jar



Eclipse: eseguire un archivio .jar



```
C:\Users\Annalisa>D:
D:\hardisk\Documenti\corsi\2017\map\case study>java -jar helloworld.jar Luna
Hello Luna
D:\hardisk\Documenti\corsi\2017\map\case study>_
```