



Eclipse

Avviare un progetto e compilare un semplice programma

Descrizione di Eclipse

- Eclipse è un ambiente di sviluppo che facilita la scrittura ed il debug di programmi java
- Permette di:
 - Scrivere il codice delle classi
 - Compilare le classi
 - Eseguire il codice java
- Fornisce strumenti per la ricerca degli errori

Come ottenere Eclipse

<https://www.eclipse.org/downloads/packages/release/2018-12/r/eclipse-ide-java-developers>

The screenshot shows the Eclipse IDE for Java Developers download page. A red box highlights the 'Download Links' section, which lists the download links for Windows 64-bit, Mac OS X (Cocoa) 64-bit, and Linux 64-bit. Another red box highlights the text 'Selezione in base al sistema operativo' (Selection based on the operating system), with a line pointing to the download links. The page also includes a 'Package Description' section, a 'Downloaded 385,204 Times' statistic, a 'Bugzilla' section, and a 'Get Eclipse IDE 2018-12' section with a 'Download 64 bit' button.

Eclipse IDE for Java Developers

Package Description

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration

This package includes:

- Detailed features list

Maintained by: Eclipse Packaging Project

Download Links

- Windows 64-bit
- Mac OS X (Cocoa) 64-bit
- Linux 64-bit

Downloaded 385,204 Times

► Checksums...

Bugzilla

- Open Bugs: 35
- Resolved Bugs: 128

File a Bug on this Package

New and Noteworthy

Eclipse Platform

Get **Eclipse IDE 2018-12**

Install your favorite desktop IDE packages.

Download 64 bit

Download Packages | Need Help?

RELATED LINKS

Selezione in base al sistema operativo

FAQ (Frequently Asked Question)

Q. Quando provo ad avviare il programma mi si apre una finestra di errore con scritto:

"A java runtime enviroment (JRE) or java development kit (JDK) must be available to run eclipse. No java virtual machine was found after searching the following locations: C:\Users\Desktop\eclipse\jre\bin\javaw.exe javaw.exe in your current PATH". Cosa devo fare?

A. Eclipse ha bisogno di un ambiente Java per poter essere eseguito, poichè esso stesso è un programma scritto in Java. Per risolvere il problema si provi ad installare la JDK scaricandola da questo link:

<https://www.java.com/it/download/manual.jsp>

Avviare Eclipse

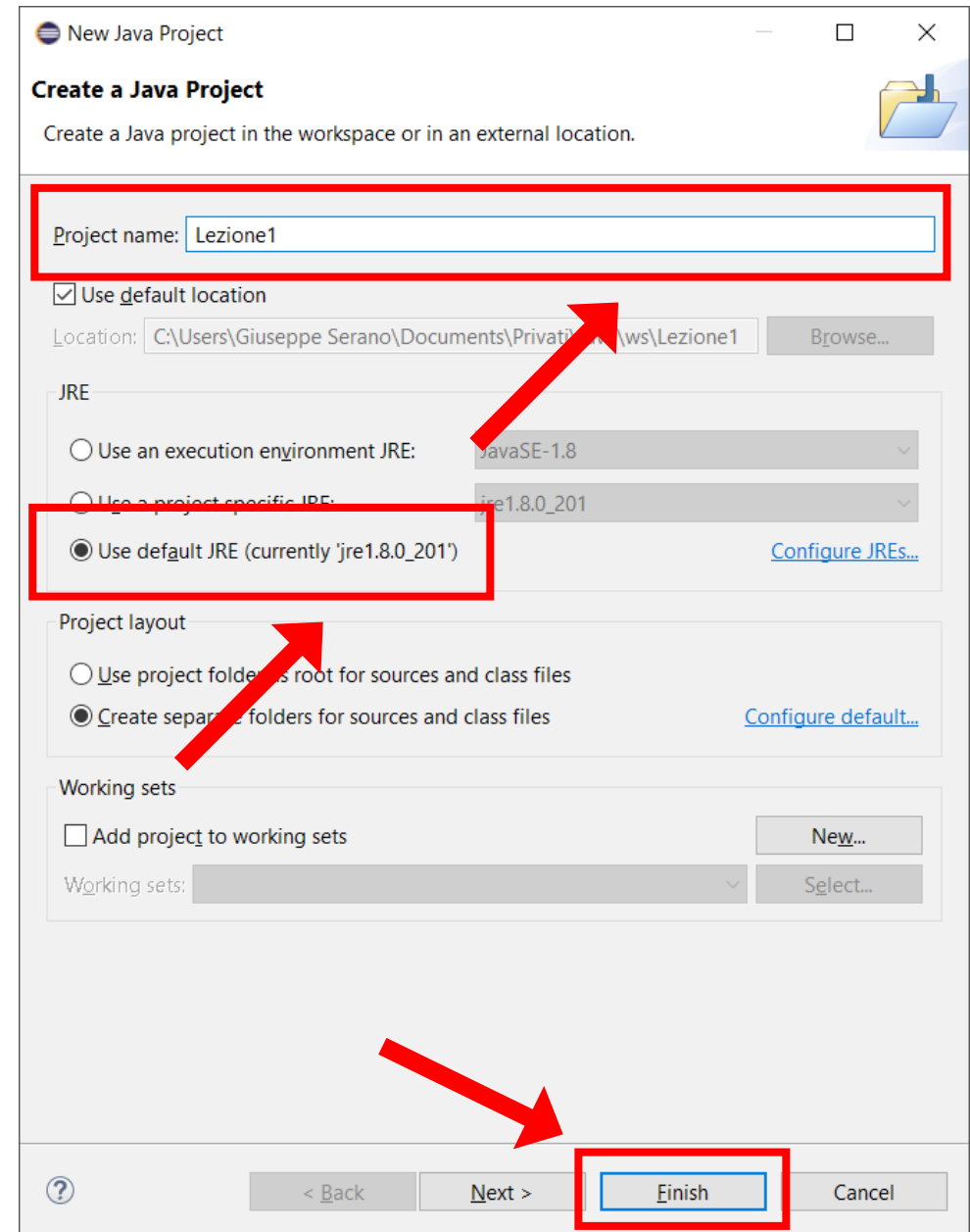
- Eclipse è una cartella compressa
- Va decompressa nel File-System
- Si ottiene di fatto una nuova cartella “eclipse”
- Al cui interno contiene il file “eclipse.exe”
- Per avviare Eclipse bisogna lanciare il file “eclipse.exe”

Operazioni di base

- Creare un nuovo progetto
- Creare una classe (nuovo file java)
- Scrivere il codice
- Compilare il codice
- Eseguire il programma

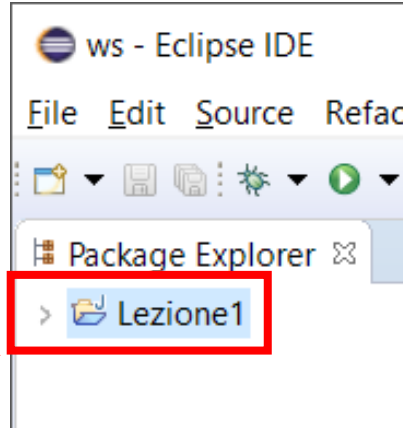
Creare un progetto

- Selezionare dal menù:
File → New → Project
- Scegliere “Java Project”
- Inserire il nome del progetto: “Lezione1”
- Selezionare “Use default JRE (...)”
- Cliccare “Finish”

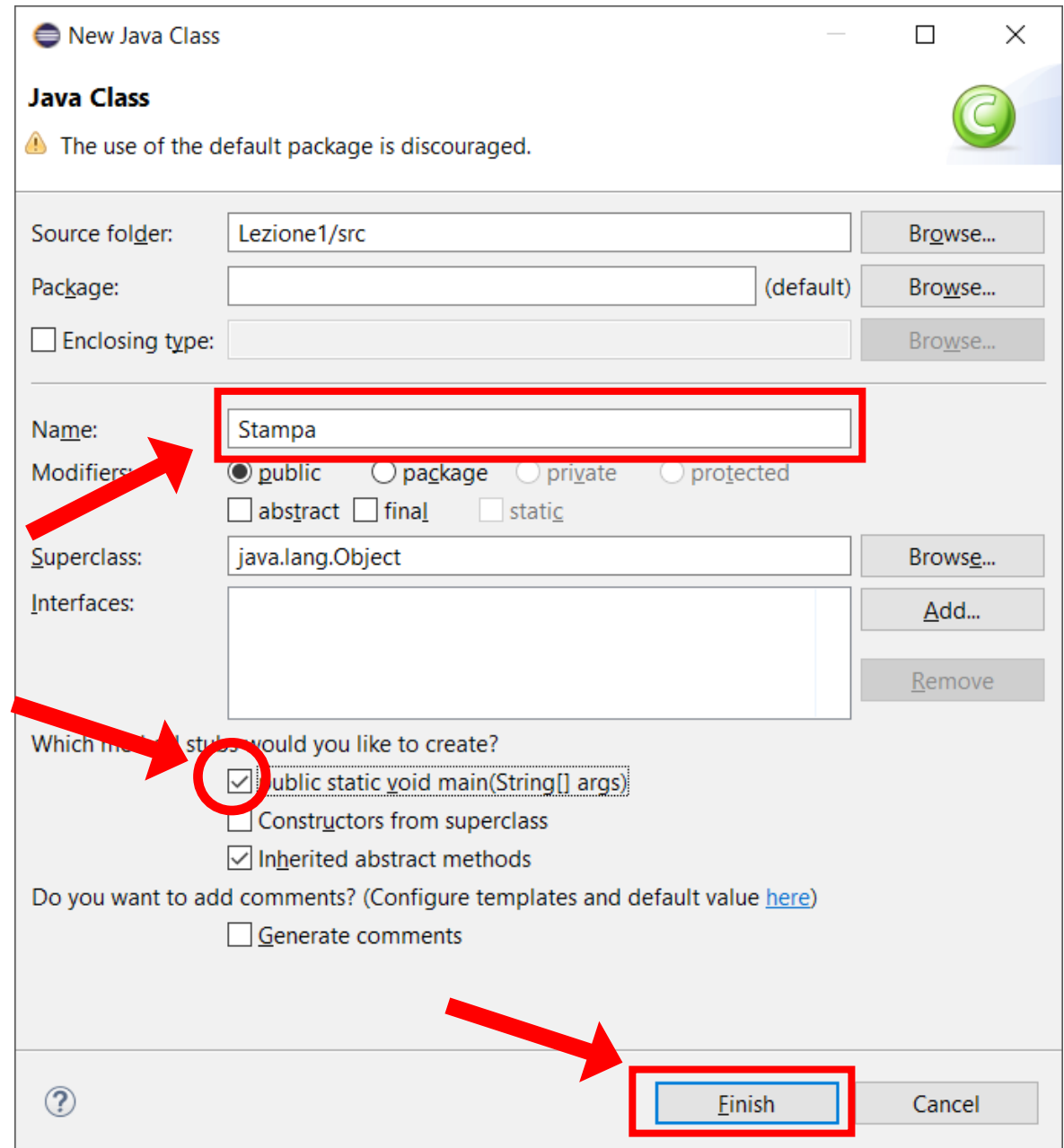


Creare una classe

- Selezionare il progetto dal tab “Package Explorer”



- Selezionare dal menù:
File → New → Class
- Inserire il nome della classe: “Stampa”
- Spuntare la casella di creazione del main
- Cliccare “Finish”



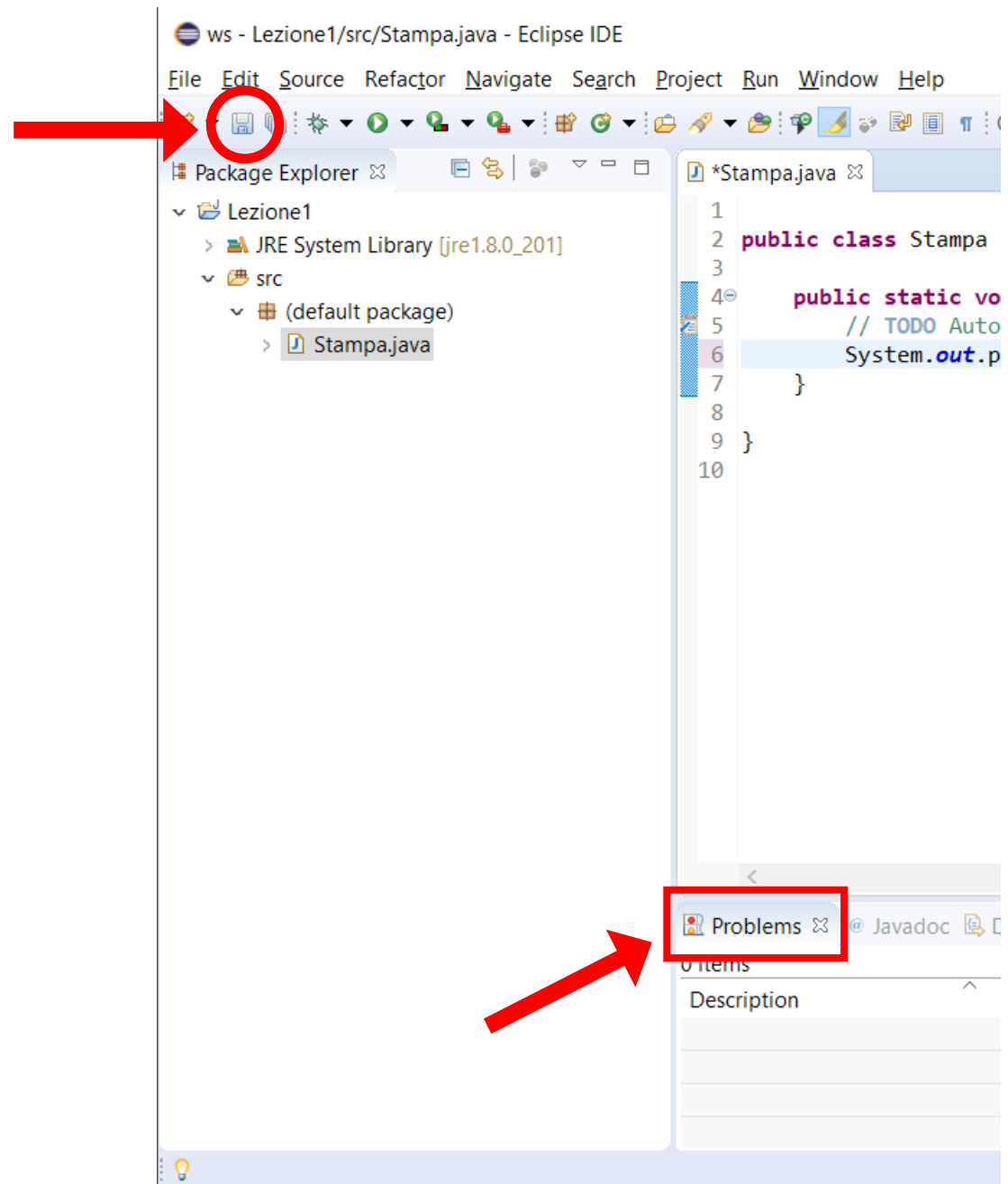
Scrivere il codice

- Eclipse genera automaticamente lo scheletro del programma.
- Le istruzioni del programma vanno inserite nel corpo del metodo main (attenzione alle parentesi!)
- Aggiungere il codice per stampare su schermo il messaggio "Hello, world!"

```
public class Stampa {  
  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        System.out.println("Hello, world!");  
    }  
  
}
```

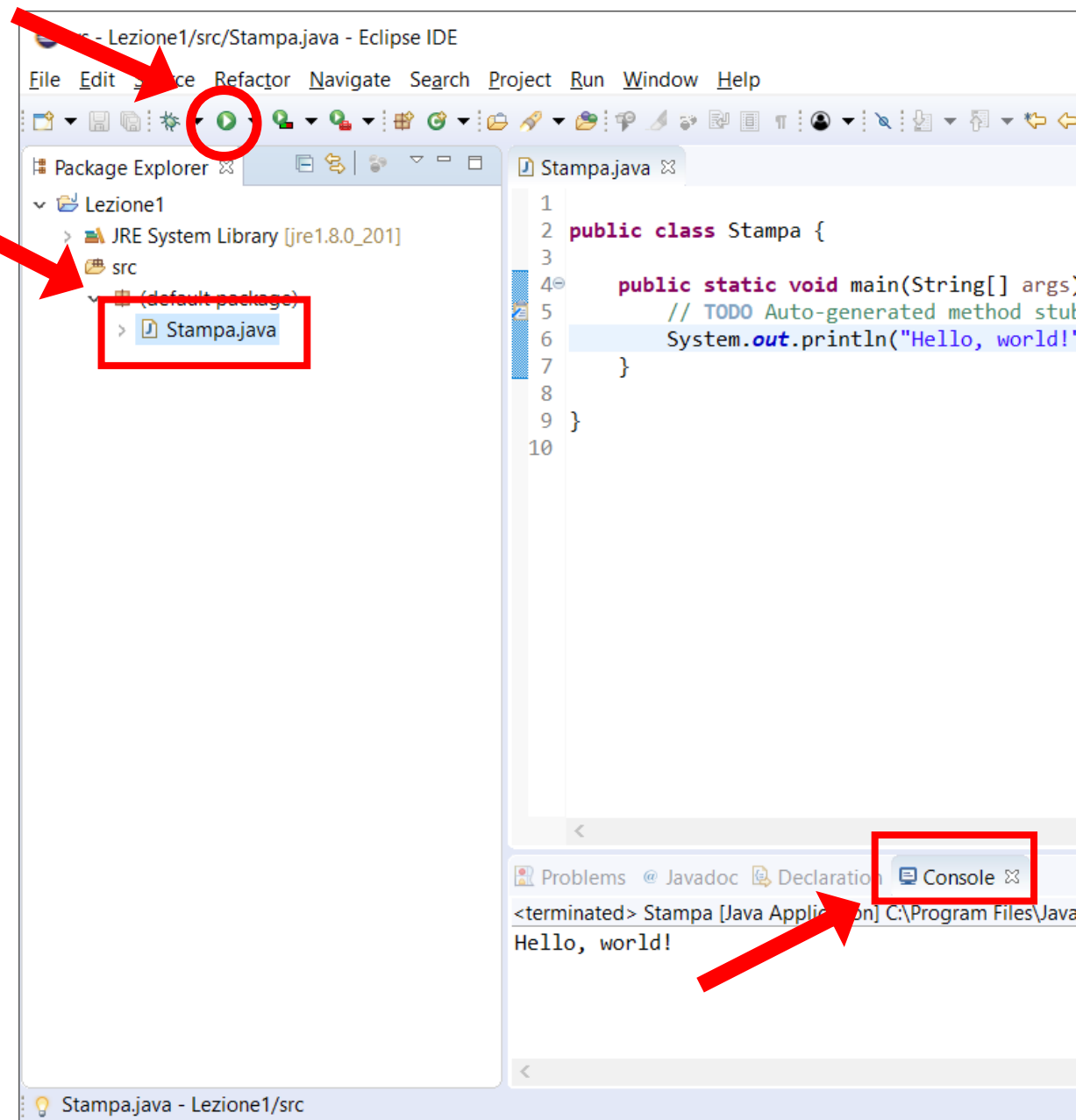
Compilare il codice

- Salvare il file “Stampa.java” usando l'apposito pulsante
- Se l'opzione di “build automatically” è attivata, il file verrà compilato automaticamente
- Altrimenti compilare selezionando sul menu Project → Build All
- Eventuali errori saranno mostrati nel tab “Problems”
- L'opzione “build automatically” può essere attivata e disattivata selezionando Project → Build Automatically



Eseguire il codice

- Selezionare il file “Stampa.java” dal Package explorer
- Selezionare dal menu Run → Run As... → Java Application
- In alternativa, usare il pulsante “run”
- L’output del programma sarà mostrato nel tab “Console”

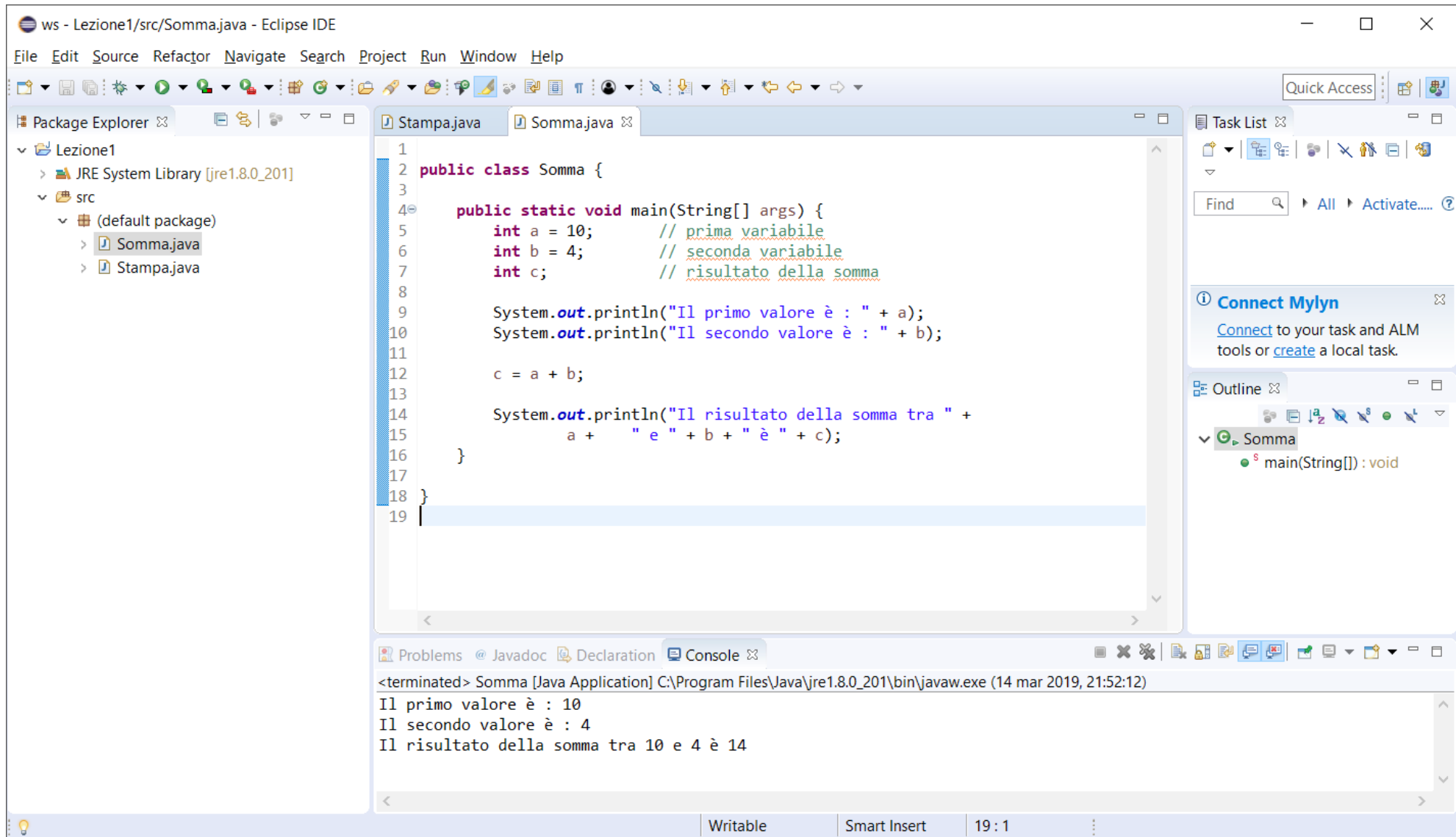


Esempio: Somma.java (1/2)

- Programma che effettua la somma di due interi

```
public class Somma {  
    public static void main(String[] args) {  
        int a = 10;           // prima variabile  
        int b = 4;            // seconda variabile  
        int c;                 // risultato della somma  
  
        System.out.println("Il primo valore è : " + a);  
        System.out.println("Il secondo valore è : " + b);  
  
        c = a + b;  
  
        System.out.println("Il risultato della somma tra " +  
                           a + " e " + b + " è " + c);  
    }  
}
```

Esempio: Somma.java (2/2)



Debugging

- L'attività di debugging consiste nel rilevare e correggere gli errori di programmazione (bug)
- Esistono due tipi di errori:
 - Errori sintattici, che impediscono la compilazione del codice
 - Errori semantici, che si manifestano a tempo di esecuzione
- Il compilatore indica gli errori sintattici nel tab “Problems” e sottolineando le istruzioni errate nel codice.
- Gli errori semantici sono causati da errori nella logica del programma e causano un output non corretto. Il compilatore non rileva tali errori.

Errori di compilazione (1/6)

“.” mancante

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left shows a project named 'Lezione1' with a source folder 'src' containing 'Somma.java' and 'Stampa.java'. The main editor displays the code for 'Somma.java'. The code is as follows:

```
1
2 public class Somma {
3
4     public static void main(String[] args) {
5         int a = 10;      // prima variabile
6         int b = 4;       // seconda variabile
7         int c;           // risultato della somma
8
9         System.out.println("Il primo valore è : " + a);
10        System.out.println("Il secondo valore è : " + b);
11
12        c = a + b;
13
14        System.out.println("Il risultato della somma tra " +
15                           a + " e " + b + " è " + c);
16    }
17 }
18
19
```

A red error icon is visible on line 9. The bottom of the IDE shows the 'Problems' view with the following error:

Description	Resource	Path	Location	Type
1 error, 0 warnings, 0 others				
✖ Errors (1 item)				
✖ Syntax error, insert ";" to complete Statement	Somma.java	/Lezione1/src	line 9	Java Problem

Errori di compilazione (2/6)

Nome della classe "System" errato

The screenshot shows the Eclipse IDE with a project named 'Lezione1'. The Package Explorer on the left shows the project structure: 'JRE System Library [jre1.8.0_201]', 'src', '(default package)', 'Somma.java', and 'Stampa.java'. The central editor window displays the code for 'Somma.java':

```
1 public class Somma {
2
3
4     public static void main(String[] args) {
5         int a = 10;    // prima variabile
6         int b = 4;     // seconda variabile
7         int c;         // risultato della somma
8
9         Syste.out.println("Il primo valore è : " + a);
10        System.out.println("Il secondo valore è : " + b);
11
12        c = a + b;
13
14        System.out.println("Il risultato della somma tra " +
15                           a + " e " + b + " è " + c);
16    }
17 }
18 }
19 }
```

The error is highlighted on line 9: 'Syste.out.println'. The IDE's Problems view at the bottom shows the following error:

Description	Resource	Path	Location	Type
1 error, 0 warnings, 0 others				
Errors (1 item)				
System cannot be resolved	Somma.java	/Lezione1/src	line 9	Java Problem

The status bar at the bottom indicates 'Writable', 'Smart Insert', and the time '9:55'.

Errori di compilazione (3/6)

Più errori: “;” mancante e nome della classe System errato.
Le maiuscole/minuscole devono essere usate correttamente!

The screenshot shows the Eclipse IDE interface. The main editor displays the code for `Somma.java`. The code is as follows:

```
1 public class Somma {
2
3
4 public static void main(String[] args) {
5     int a = 10;    // prima variabile
6     int b = 4;    // seconda variabile
7     int c;        // risultato della somma
8
9     system.out.println("Il primo valore è : " + a);
10    system.out.println("Il secondo valore è : " + b);
11
12    c = a + b;
13
14    System.out.println("Il risultato della somma tra " +
15        a + " e " + b + " è " + c);
16 }
17 }
18 }
19 }
```

The IDE shows three errors in the `Problems` view at the bottom:

- Syntax error, insert ";" to complete Statement (line 9)
- system cannot be resolved (line 9)
- system cannot be resolved (line 10)

The `Outline` view on the right shows the class `Somma` with the `main(String[]): void` method.

Description	Resource	Path	Location	Type
Syntax error, insert ";" to complete Statement	Somma.java	/Lezione1/src	line 9	Java Problem
system cannot be resolved	Somma.java	/Lezione1/src	line 9	Java Problem
system cannot be resolved	Somma.java	/Lezione1/src	line 10	Java Problem

Errori di compilazione (4/6)

Spesso non tutti gli errori vengono rilevati contemporaneamente.

In questo caso, la mancanza del " nasconde la mancanza del “;”

The screenshot shows the Eclipse IDE interface. The main editor displays the code for `Somma.java`. The code is as follows:

```
1 public class Somma {
2
3
4     public static void main(String[] args) {
5         int a = 10;    // prima variabile
6         int b = 4;     // seconda variabile
7         int c;         // risultato della somma
8
9         System.out.println("Il primo valore è : + a)
10        System.out.println("Il secondo valore è : " + b);
11
12        c = a + b;
13
14        System.out.println("Il risultato della somma tra " +
15            a + " e " + b + " è " + c);
16    }
17 }
18 }
19 }
```

The error message in the Problems view is:

1 error, 0 warnings, 0 others

Description	Resource	Path	Location	Type
String literal is not properly closed by a double-qu	Somma.java	/Lezione1/src	line 9	Java Problem

The error message is truncated in the screenshot. The status bar at the bottom shows "Writable", "Smart Insert", and "9 : 54".

Errori di compilazione (5/6)

È buona norma ricompilare il codice dopo la correzione di ciascun errore, per rilevare eventuali errori nascosti.

Dopo la correzione del " mancante, ricompilando si rileva la mancanza del ":"

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left shows a project named 'Lezione1' with a source folder 'src' containing 'Somma.java' and 'Stampa.java'. The main editor displays the code for 'Somma.java'. The code is as follows:

```
1 public class Somma {
2
3
4     public static void main(String[] args) {
5         int a = 10;    // prima variabile
6         int b = 4;     // seconda variabile
7         int c;         // risultato della somma
8
9         System.out.println("Il primo valore è : + a");
10        System.out.println("Il secondo valore è : " + b);
11
12        c = a + b;
13
14        System.out.println("Il risultato della somma tra " +
15            a + " e " + b + " è " + c);
16    }
17 }
18
19
```

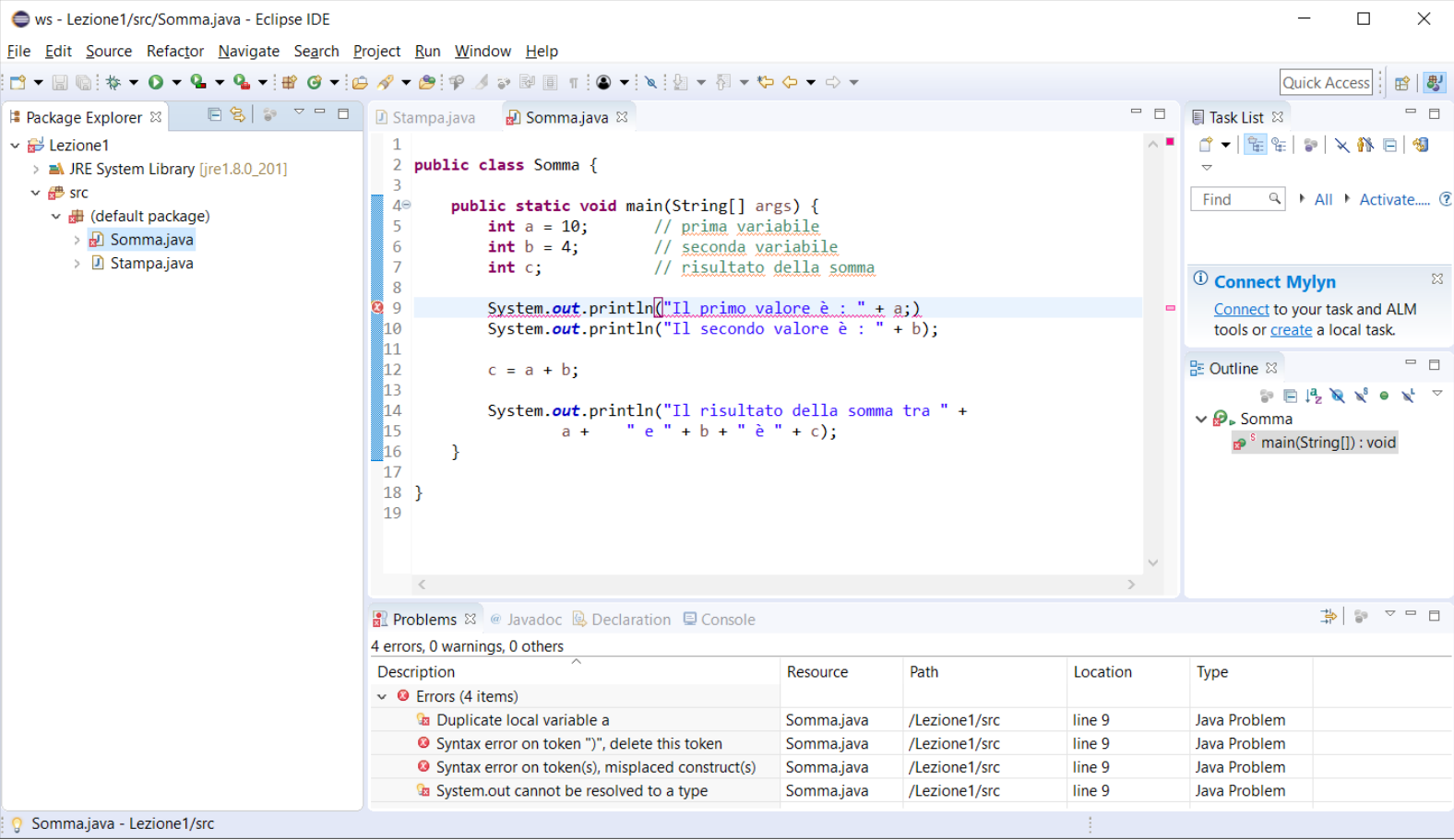
Line 9 has a red error icon next to it. The error message is: 'Syntax error, insert ";" to complete Statement'. The error is located at line 9, column 40. The status bar at the bottom shows '1 error, 0 warnings, 0 others'.

Description	Resource	Path	Location	Type
Syntax error, insert ";" to complete Statement	Somma.java	/Lezione1/src	line 9	Java Problem

Errori di compilazione (6/6)

A volte i messaggi del compilatore possono essere ambigui o fuorvianti. Non bisogna sempre seguire alla lettera le indicazioni del compilatore, ma usarlo come ausilio per capire l'azione da intraprendere.

In questo caso, la parentesi ed il ";" sono invertiti, ma il compilatore non riesce a capirlo e dà una serie di messaggi fuorvianti.



The screenshot shows the Eclipse IDE interface with the file `Somma.java` open. The code contains several syntax errors. The `Problems` view at the bottom displays four errors:

Description	Resource	Path	Location	Type
Duplicate local variable a	Somma.java	/Lezione1/src	line 9	Java Problem
Syntax error on token ")", delete this token	Somma.java	/Lezione1/src	line 9	Java Problem
Syntax error on token(s), misplaced construct(s)	Somma.java	/Lezione1/src	line 9	Java Problem
System.out cannot be resolved to a type	Somma.java	/Lezione1/src	line 9	Java Problem

The code in `Somma.java` is as follows:

```
1 public class Somma {
2
3
4     public static void main(String[] args) {
5         int a = 10;    // prima variabile
6         int b = 4;    // seconda variabile
7         int c;        // risultato della somma
8
9         System.out.println("Il primo valore è : " + a);
10        System.out.println("Il secondo valore è : " + b);
11
12        c = a + b;
13
14        System.out.println("Il risultato della somma tra " +
15            a + " e " + b + " è " + c);
16    }
17 }
18
19
```

Esercizio 1

- Effettuare il debug del seguente programma

```
public class Esercizio1 {  
  
    public static void main(String[] args) {  
  
        int a = 5;  
        int b = 2;  
  
        System.out.println(a;  
        System.out.println(b);  
  
    }  
  
}
```

Esercizio 2

- Effettuare il debug del seguente programma

```
public class Esercizio2 {  
  
    public static void main(String[] args) {  
  
        int a;  
        int b = 2;  
  
        System.out.println(a);  
        System.out.println(b)  
  
    }  
  
}
```

Esercizio 3

- Scrivere un programma che:
 - Inizializza due variabili di tipo intero che rappresentano la base e l'altezza di un triangolo.
 - Calcola l'area del triangolo come numero intero
 - Stampa a video il risultato

Soluzione

```
public class AreaTriangolo {  
  
    public static void main(String[] args) {  
        int altezza = 10;  
        int base = 4;  
        int area, tmp;  
  
        System.out.println("La base è : " + base);  
        System.out.println("L'altezza è : " + altezza);  
  
        tmp = base * altezza;  
        area = tmp/2;  
  
        System.out.println("L'area è: " + area);  
  
    }  
  
}
```

Variante 1

Variante col calcolo in un solo passo

```
public class AreaTriangoloVariante1 {  
  
    public static void main(String[] args) {  
        int altezza = 10;  
        int base = 4;  
        int area;  
  
        System.out.println("La base è : " + base);  
        System.out.println("L'altezza è : " + altezza);  
  
        area = (base * altezza)/2;  
  
        System.out.println("L'area è: " + area);  
  
    }  
  
}
```

Variante 2

Variante col calcolo nel println

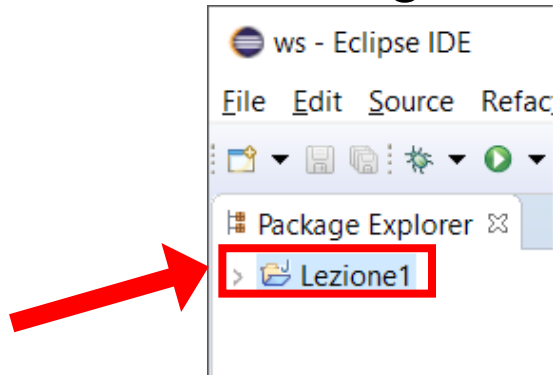
```
public class AreaTriangoloVariante2 {  
  
    public static void main(String[] args) {  
        int altezza = 10;  
        int base = 4;  
  
        System.out.println("La base è : " + base);  
        System.out.println("L'altezza è : " + altezza);  
  
        System.out.println("L'area è: " + (base * altezza)/2);  
    }  
}
```

Package Java

- I package sono collezioni di classi java raggruppate in un unico “pacchetto”.
- Esistono package già compilati che mettono a disposizione al programmatore funzionalità avanzate.
- Spesso sono memorizzati sotto forma di file “.jar”
- Alcuni esempi:
 - java.io — operazioni su file
 - java.math — operazioni aritmetiche avanzate e trigonometria
 - java.nio — il nuovo framework I/O per Java
 - java.net — operazioni di rete, socket, ricerca DNS, ...
 - java.security — generazione di chiavi crittografiche, sicurezza e crittografia

Aggiungere un package

- Assicurarsi di aver selezionato il progetto dal “Package Explorer”



- Selezionare dal menù: Project → Properties
- Selezionare: “Java Build Path”
- Selezionare il tab: “Libraries”
- Cliccare: “Add External JARs”
- Selezionare il file “.jar”

