



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

<http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/galileosr2>

Selezione in base al sistema operativo

Search eclipse.org

DOWNLOAD

HOME / DOWNLOADS / PACKAGES / ECLIPSE IDE FOR JAVA DEVELOPERS

RELEASES

- Luna Packages
- Kepler Packages
- Juno Packages
- Indigo Packages
- Helios Packages
- Galileo Packages
- Ganymede Packages
- Europa Packages
- All Releases

Eclipse IDE for Java Developers

Package Description

The essential tools for any Java developer, including a Java IDE, a CVS client, XML Editor and Mylyn.

This package includes:

- ▶ Detailed features list

Download Links

- Windows 32-bit**
- Mac OS X (Cocoa) 32-bit**
- Mac OS X (Cocoa) 64-bit**
- Linux 32-bit**
- Linux 64-bit**

Downloaded 2 Times

▶ Checksums...

Bugzilla

Maintained by: Eclipse Mylyn Project

FAQ (Frequently Asked Question)

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

"A java runtime environment (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:

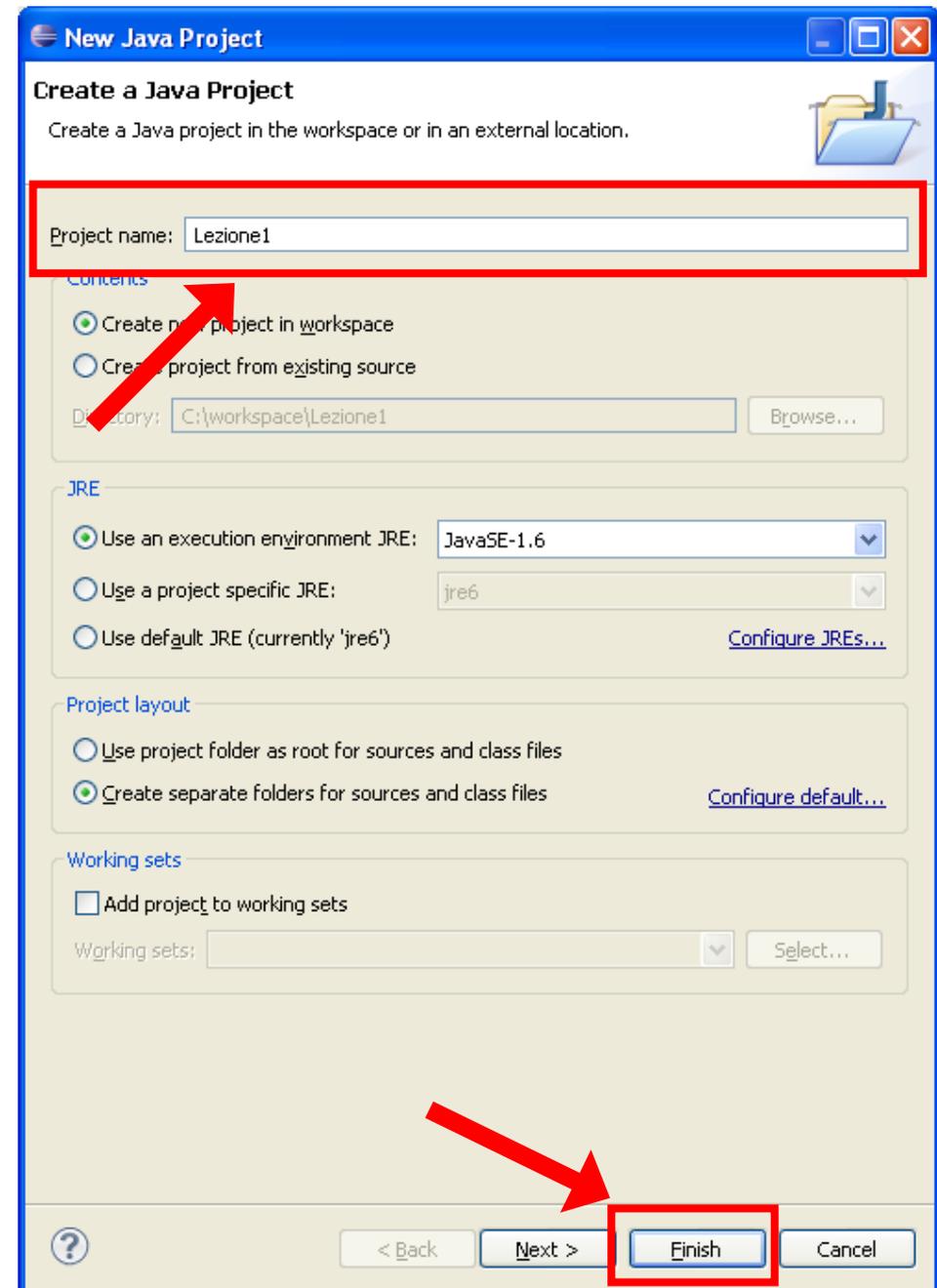
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Operazioni di base

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

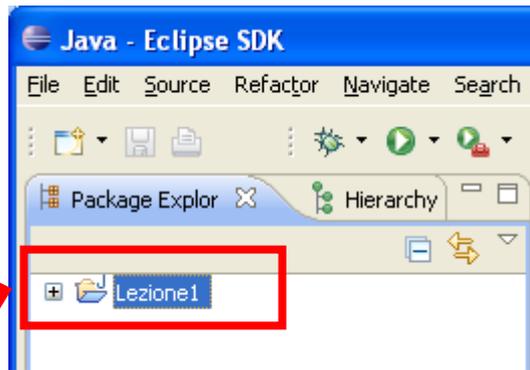
Creare un progetto

- Selezionare dal menù:
File → New → Project
- Scegliere “Java Project”
- Inserire il nome del progetto: “Lezione1”
- Cliccare “Finish”

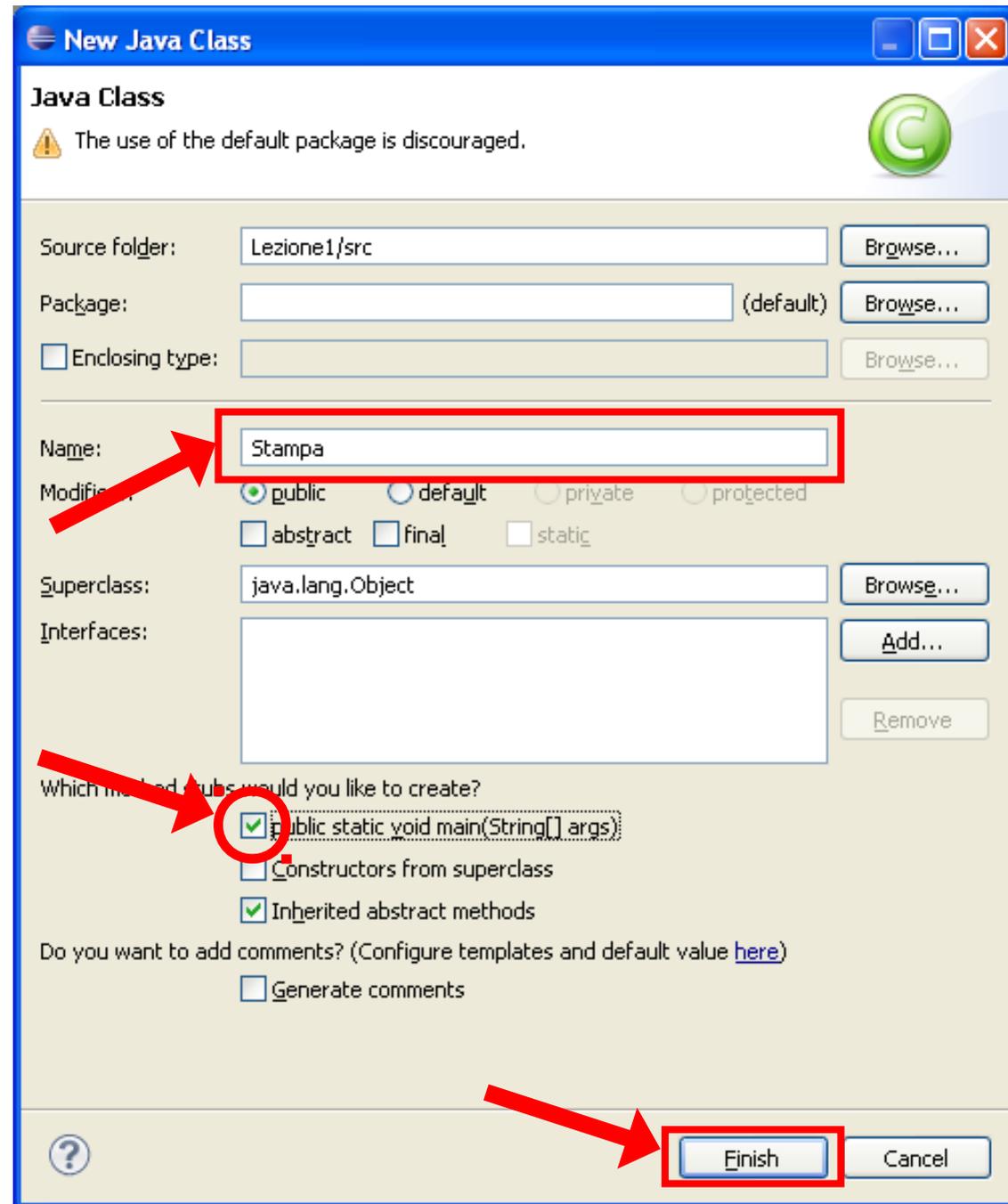


Creare una classe

- 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”



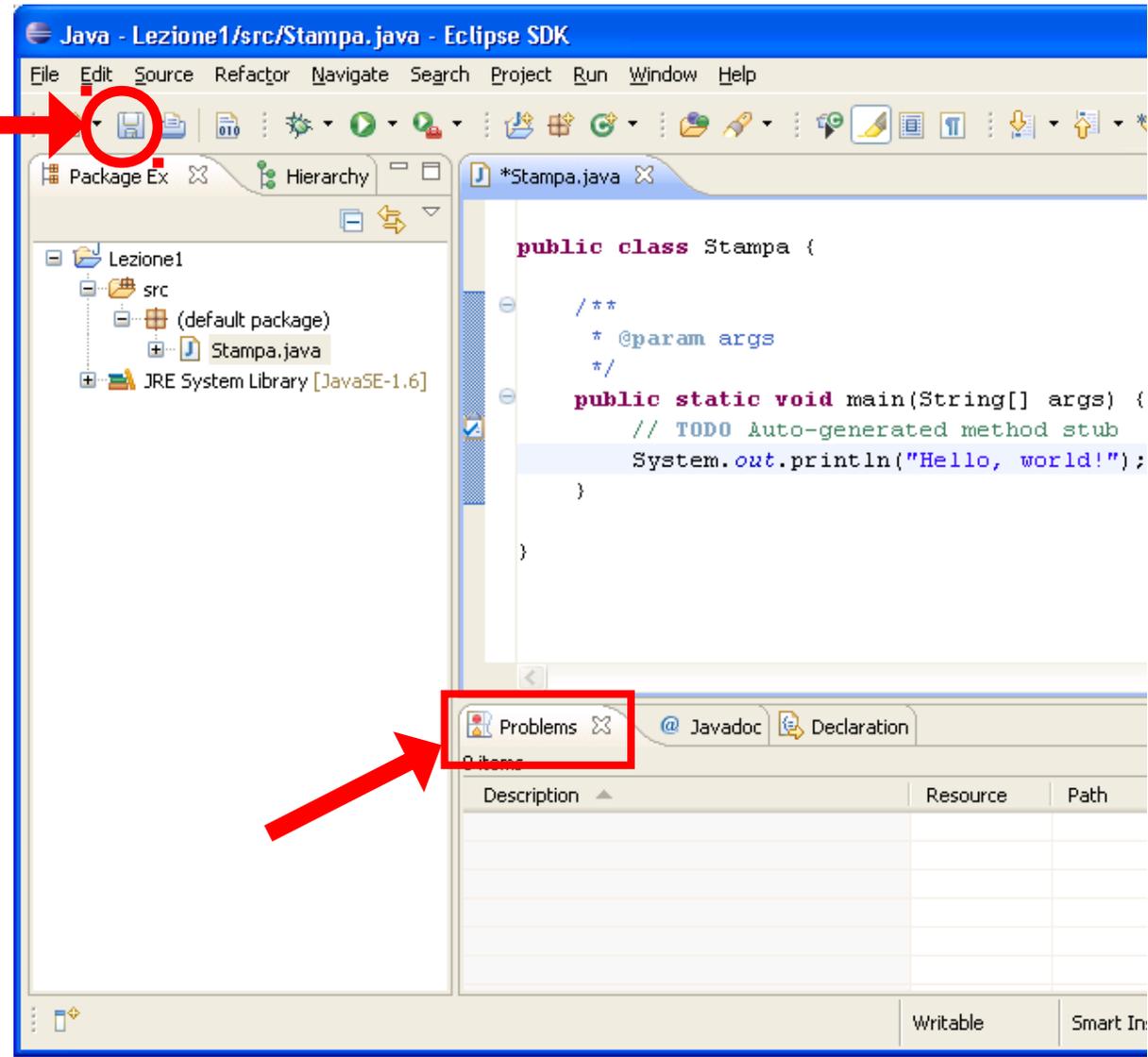
Scrivere il codice

- 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!");  
    }  
  
}
```

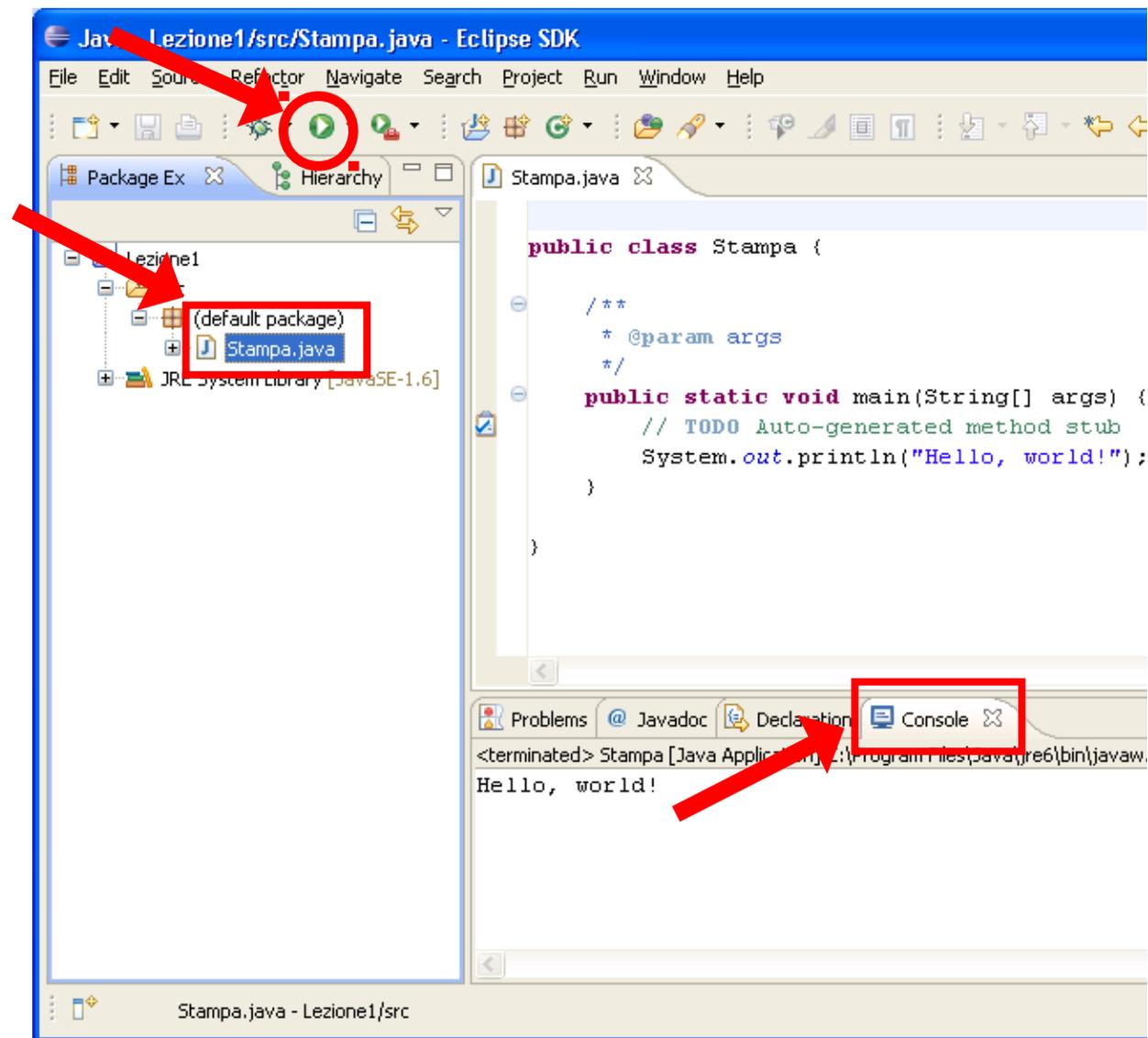
Compilare il codice

- 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)

The screenshot displays the Eclipse IDE interface. The main editor window shows the source code for the `Somma` class. The code defines a `main` method that initializes two variables, `a` and `b`, and calculates their sum, `c`. The output is printed to the console.

```
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 +  
    }  
}
```

The console window at the bottom shows the execution output:

```
<terminated> Somma [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (08/apr/2011 01.08.53)  
Il primo valore è : 10  
Il secondo valore è : 4  
Il risultato della somma tra 10 e 4 è 14
```

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 main editor window displays the following Java code:

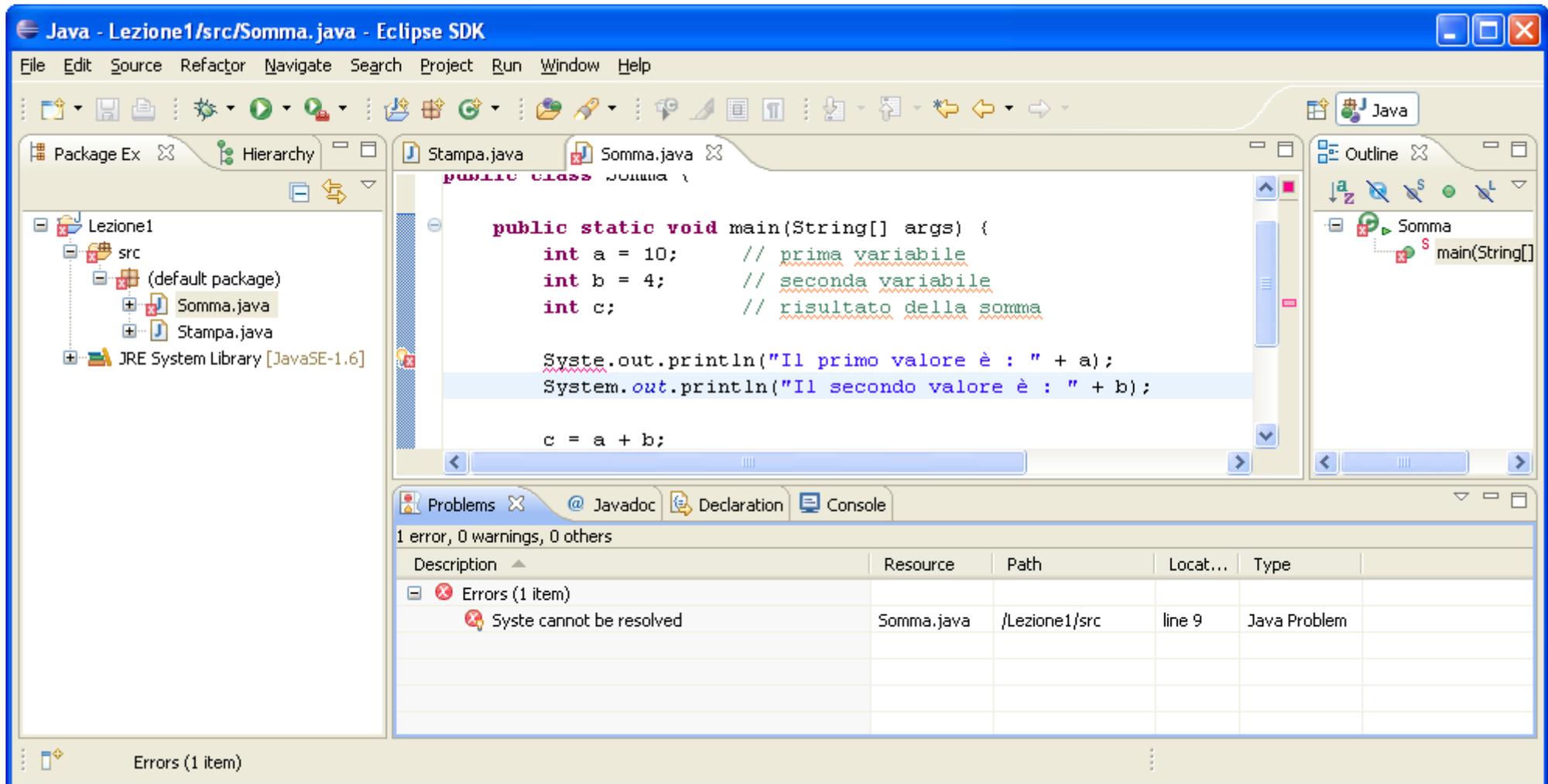
```
public class Somma {  
  
    public static void main(String[] args) {  
        int a = 10;    // prima variabile  
        int b = 4;    // seconda variabile  
        int c;        // risultato della somma  
  
        System.out.println("Il primo valore è : " + a);  
        System.out.println("Il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

The error message in the Problems view is:

Description	Resource	Path	Locat...	Type
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 interface. The main editor displays the following code in `Somma.java`:

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

The `Syste.out.println` line is highlighted in blue, indicating an error. The `Problems` view at the bottom shows the following error:

Description	Resource	Path	Locat...	Type
Errors (1 item)				
System cannot be resolved	Somma.java	/Lezione1/src	line 9	Java Problem

The status bar at the bottom left indicates "Errors (1 item)".

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 following Java code:

```
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;  
    }  
}
```

The IDE highlights three errors in the Problems view:

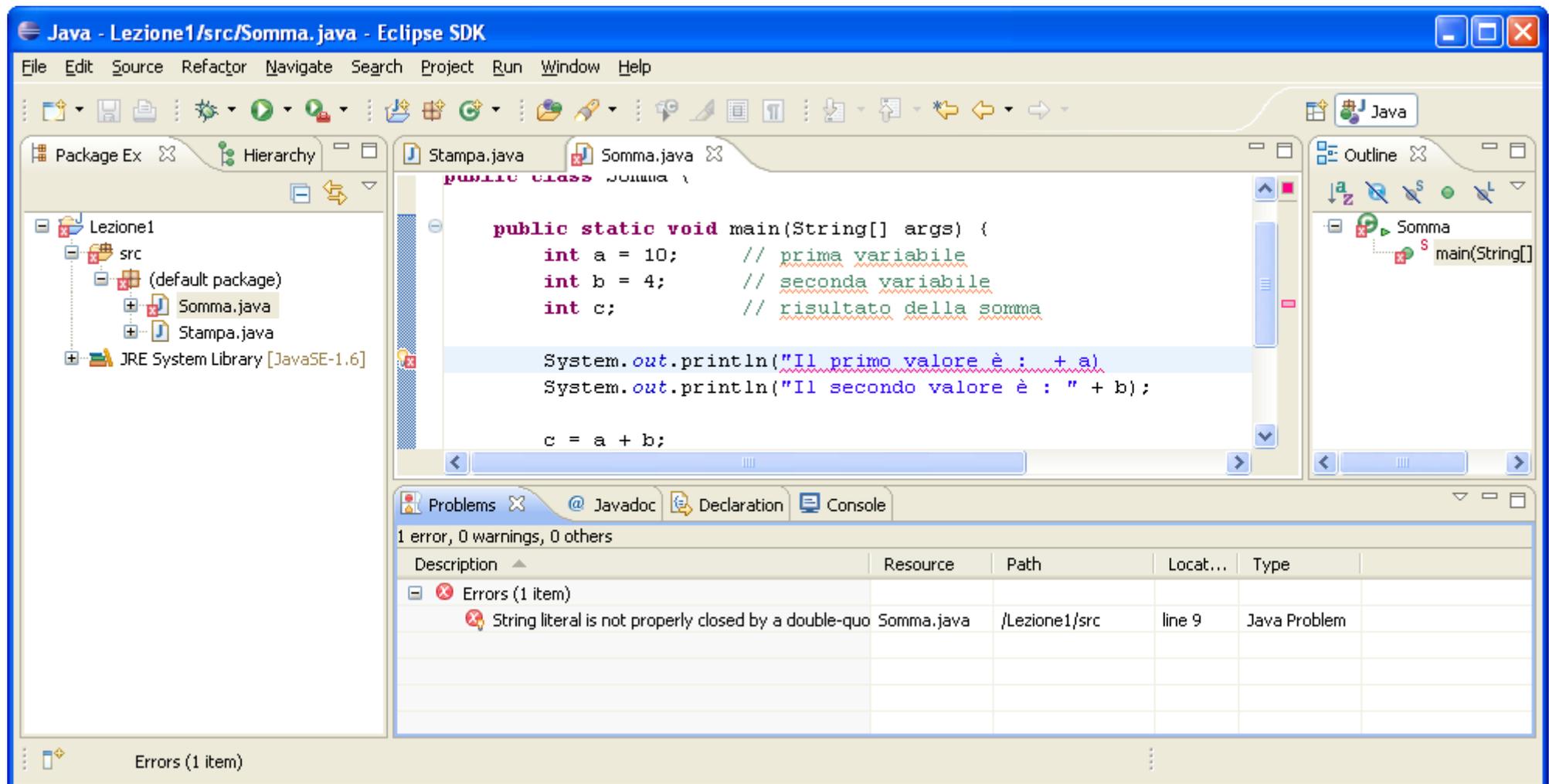
Description	Resource	Path	Locat...	Type
Errors (3 items)				
Syntax error, insert ";" to complete Statement	Somma.java	/Lezione1/src	line 9	Java Problem
system cannot be resolved	Somma.java	/Lezione1/src	line 9	Java Problem
system cannot be resolved	Somma.java	/Lezione1/src	line 10	Java Problem

The status bar at the bottom left indicates "Errors (1 item)".

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 following Java code:

```
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;  
    }  
}
```

The error message in the Problems view is:

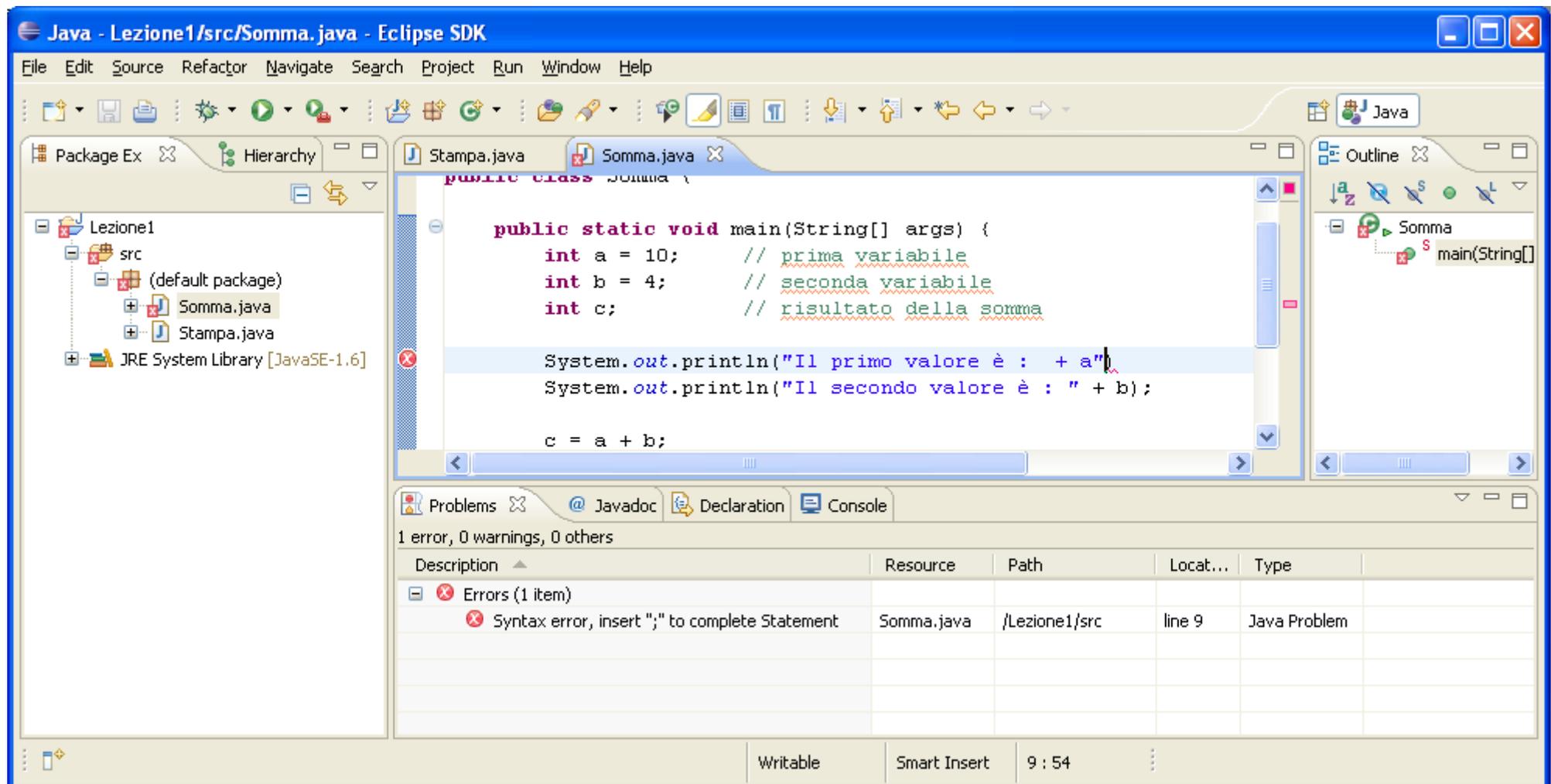
Description	Resource	Path	Locat...	Type
Errors (1 item)				
String literal is not properly closed by a double-quo	Somma.java	/Lezione1/src	line 9	Java Problem

The error message is truncated in the image. The status bar at the bottom indicates "Errors (1 item)".

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 main editor window displays the following Java code:

```
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;  
    }  
}
```

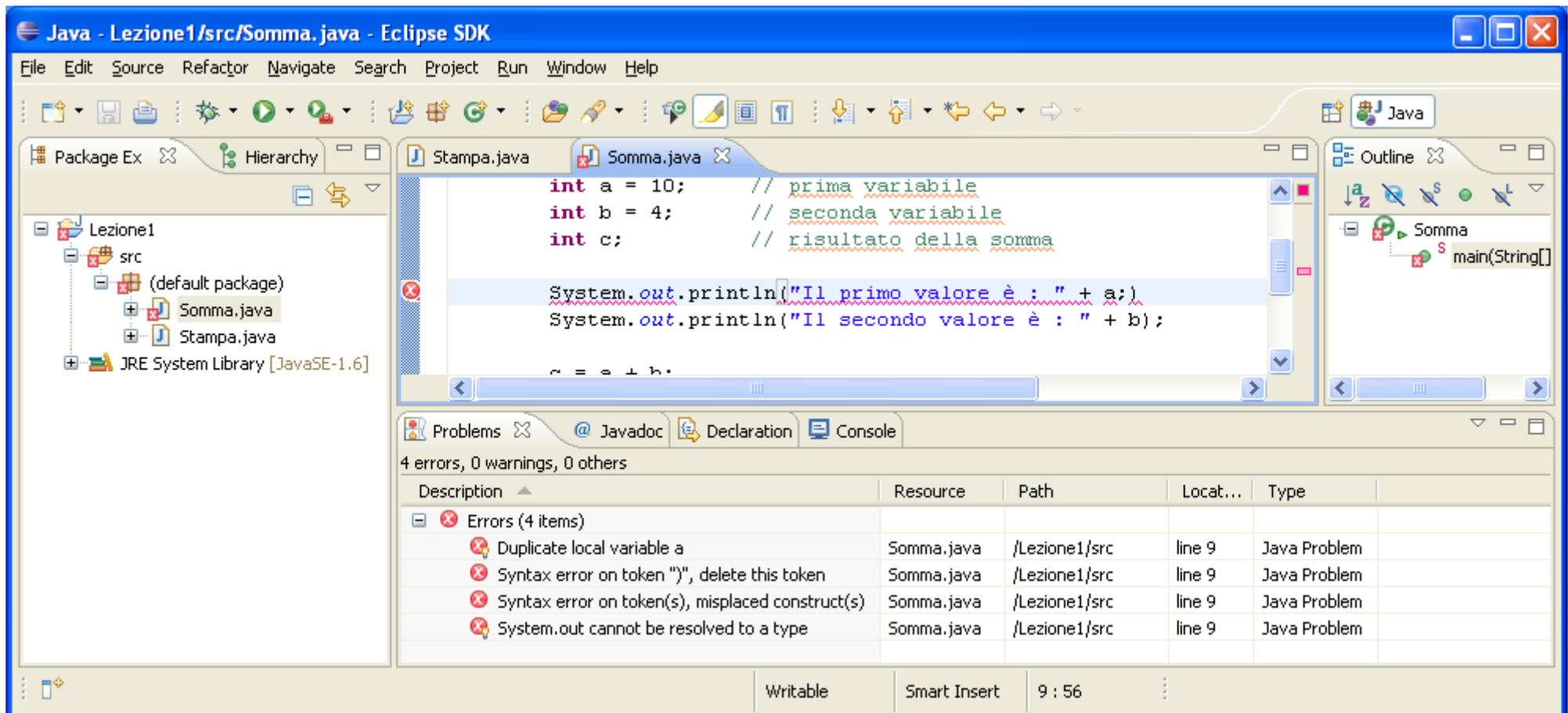
The error message in the Problems view is:

Description	Resource	Path	Locat...	Type
Errors (1 item)				
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.



```
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;
```

4 errors, 0 warnings, 0 others

Description	Resource	Path	Locat...	Type
Errors (4 items)				
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

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

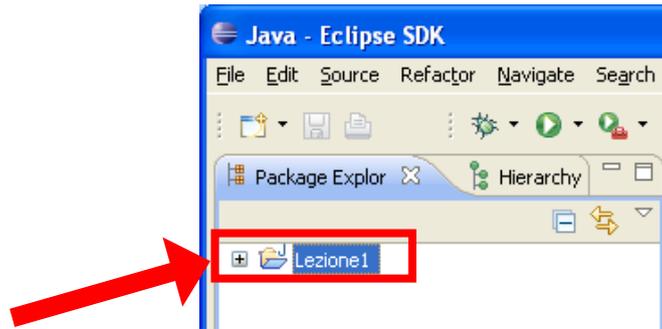
```
public class AreaTriangoloVariante2 {  
  
    public static void main(String[] args) {  
        int altezza = 10;  
        int base = 4;  
  
        System.out.println("La base è : " + base);  
        System.out.println("L'altezza è : " + altezza);  
  
        System.out.println("L'area è: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

