

Impariamo a
programmare in

JavaTM
Lezione 3



Gp.it

Errori di programmazione

```
public class Hello  
{public static void main(String[] args)  
    {System.out.println("Hello, World!");}  
}
```

```
System.out.println("Hello, World!");
```

L'attività di programmazione, come ogni altra attività di progettazione, è soggetta ad errori di vario tipo, errori di sintassi o di compilazione.

```
System.aut.println("Hello, World!");
```

```
System.out.println("Hello, World!");
```

errori logici o di esecuzione

```
System.out.println("Hell, World!");
```

Errori di sintassi

```
public class Hello
{public static void main(String[] args)
    {System.out.println("Hello, World!");}
}
```

```
System.aut.println("Hello, World!");
```

In questo caso il compilatore riesce agevolmente ad individuare e segnalare l'errore di sintassi, perché identifica il nome di un oggetto (*simbolo*) che non è stato definito (**aut**) e che, quindi, non riesce a **trovare**.

POSIZIONE
(NUMERO DI RIGA)

```
C:\>javac Hello.java
Hello.java:3:cannot find symbol
Symbol : variable aut
Location : java.lang.System
{ System.aut.println("Hello, World!");
    ^
1 error
```

POSIZIONE
(NELLA RIGA)

DIAGNOSI

Errori di sintassi

```
public class Hello
{public static void main(String[] args)
    {System.out.println("Hello, World!");}
}
```

`System.out.println("Hello, World!);`

VIRGOLETTE
MANCANTI

Questo è invece un caso molto più complesso: viene giustamente segnalato il **primo errore**, una stringa non terminata, e viene evidenziato il punto dove *inizia* la stringa

```
C:\>javac Hello.java
Hello.java:3:unclosed string literal
    { System.out.println("Hello, World!);
                        ^
Hello.java:4: `)' ` expected
    {
    ^
2 errors
```

Errori di sintassi

```
public class Hello
{public static void main(String[] args)
    {System.out.println("Hello, World!");}
}
```

Viene però segnalato anche un **secondo errore**

- il compilatore *si aspetta di trovare una parentesi tonda chiusa*, in corrispondenza di quella aperta
- *la parentesi in realtà c'è*, ma il compilatore l'ha inserita all'interno della stringa, *cioè ha prolungato la stringa fino alla fine della riga*

```
C:\>javac Hello.java
Hello.java:3:unclosed string literal
    { System.out.println("Hello, World!");
                        ^
Hello.java:4: `)' ` expected ←
    {
    ^
2 errors
```

Errori Logici

```
public class Hello
{public static void main(String[] args)
    {System.out.println("Hello, World!");}
}
```

Sono molto più insidiosi degli errori di sintassi

- Il *programma* viene compilato correttamente, ma *non fa quello che dovrebbe fare*.

L'eliminazione degli errori logici richiede molta *pazienza*, eseguendo il programma e osservando con attenzione i risultati prodotti

- *È necessario collaudare i programmi, come qualsiasi altro prodotto che si compra.*

Si usano programmi specifici (*debugger*) per trovare gli errori logici (*bug*) in un programma

- *Noi non useremo un debugger*