

# Java

Manuale per le ripetizioni di Informatica

Marco Lampis

5 dicembre 2022

# Indice

<b>0</b>	<b>Informazioni generali</b>	<b>1</b>
0.1	Come svolgo le lezioni . . . . .	1
<b>1</b>	<b>Introduzione</b>	<b>3</b>
<b>2</b>	<b>Variabili</b>	<b>5</b>
<b>3</b>	<b>If</b>	<b>7</b>
<b>4</b>	<b>Switch</b>	<b>9</b>
<b>5</b>	<b>For</b>	<b>11</b>
<b>6</b>	<b>While</b>	<b>13</b>
<b>7</b>	<b>do-while</b>	<b>15</b>
<b>8</b>	<b>Array</b>	<b>17</b>
<b>9</b>	<b>Stringhe</b>	<b>19</b>



# 0 Informazioni generali

Ciao! Sono **Marco**, sono uno studente magistrale in *Software Engineering* al Politecnico di Torino e mi sono laureato in Ingegneria Informatica all'università di Pisa. Nella vita sono un programmatore, uno smanettone e amante di videogiochi! Amo quello che studio, e per questo motivo fornisco ripetizioni di informatica con particolare attenzione a:

- programmazione (Java, C++, C, Python, C#, Javascript, PHP)
- algoritmi e strutture dati
- basi di Dati
- più o meno tutto quello che riguarda l'informatica!

Sia per studenti delle scuole superiori che per l'università.

## 0.1 Come svolgo le lezioni

La mia metodologia è innovativa e interattiva: utilizzo una piattaforma apposita per lavorare contemporaneamente sullo stesso file con gli studenti, come se fossimo accanto. In questo modo, le lezioni sono più coinvolgenti e divertenti, e gli studenti possono imparare in modo facile ed efficiente. Inoltre, le lezioni includono sia esercitazioni che approfondimenti teorici, aiuto compiti e revisione.

Per aiutare gli studenti a prepararsi al meglio e a esercitarsi anche a casa, ho creato un sito web dedicato con risorse utili come teoria, esercizi e slide. Su questo sito, gli studenti possono trovare tutto il materiale di cui hanno bisogno, e il materiale è sempre disponibile gratuitamente per i miei studenti. Inoltre, uso un iPad per prendere appunti durante le lezioni e poi condividerli con gli studenti, in modo che possano rivedere tutto ciò che è stato trattato.

Se volete migliorare le vostre conoscenze in informatica e avere un supporto personalizzato e professionale, non esitate a contattarmi. Sarò felice di fornirvi maggiori informazioni e di fissare una lezione con voi.



# 1 Introduzione

Java è un linguaggio di programmazione orientato agli oggetti sviluppato da Sun Microsystems (ora Oracle Corporation) negli anni '90. È uno dei linguaggi di programmazione più popolari e diffusi al mondo e viene utilizzato per sviluppare un'ampia gamma di applicazioni, come giochi, app per dispositivi mobili, programmi per il Web e così via.

Una delle caratteristiche principali di Java è la sua portabilità. Ciò significa che un programma scritto in Java può essere eseguito su qualsiasi piattaforma che supporta Java senza modifiche, a patto che la piattaforma abbia una macchina virtuale Java (JVM) installata.

Java utilizza una sintassi simile a quella di altri linguaggi di programmazione C-style, come C++ e C#. Ecco un esempio di un programma semplice in Java che stampa "Hello, World!" sullo schermo:

```
1 public class HelloWorld {  
2     public static void main(String[] args) {  
3         System.out.println("Hello, World!");  
4     }  
5 }
```

In questo esempio, abbiamo creato una classe chiamata HelloWorld che contiene un metodo main. Il metodo main è il punto di ingresso per qualsiasi programma Java e viene chiamato quando il programma viene eseguito. Il codice all'interno del metodo main viene eseguito in sequenza, quindi la prima linea che viene eseguita è `System.out.println("Hello, World!");`, che stampa "Hello, World!" sullo schermo.

Java è un linguaggio fortemente tipizzato, il che significa che ogni variabile e ogni espressione ha un tipo specifico che determina quali operazioni possono essere eseguite su di esso. Ad esempio, in Java non è possibile assegnare un valore di tipo int (intero) a una variabile di tipo String (stringa).

Java è anche un linguaggio orientato agli oggetti, il che significa che è possibile creare oggetti a partire da classi e utilizzare questi oggetti per rappresentare entità reali o concettuali. Ad esempio, in un gioco potremmo creare una classe Player per rappresentare un giocatore e utilizzare istanze di questa classe per creare oggetti che rappresentano i singoli giocatori del gioco.



## 2 Variabili

In Java, una variabile è un contenitore per memorizzare un valore. Ogni variabile ha un nome e un tipo, che determina il tipo di valore che può contenere. Ad esempio, una variabile di tipo `int` può contenere un numero intero, mentre una variabile di tipo `String` può contenere una stringa.

Per dichiarare una variabile in Java, si utilizza la sintassi seguente:

```
1 tipo nomeVariabile;
```

Ad esempio, per dichiarare una variabile di nome `numero` di tipo `int`, potremmo scrivere:

```
1 int numero;
```

Per assegnare un valore a una variabile, si utilizza l'operatore di assegnazione `=`, ad esempio:

```
1 numero = 10;
```

Ecco alcuni tipi di variabili comunemente usati in Java:

- `int`: contiene un numero intero, ad esempio 1, -10, 100.
- `double`: contiene un numero con la virgola, ad esempio 3.14, -0.5, 0.0.
- `char`: contiene un singolo carattere, ad esempio 'A', 'B', '1'.
- `String`: contiene una stringa, ad esempio "Ciao", "Mondo", "123".
- `boolean`: contiene un valore booleano, cioè `true` o `false`.

Ecco un esempio che mostra come dichiarare e utilizzare alcune variabili di diversi tipi:

```
1 int numeroIntero = 10;
2 double numeroVirgola = 3.14;
3 char carattere = 'A';
4 String stringa = "Ciao Mondo";
5 boolean veroOFalso = true;
6
7 System.out.println(numeroIntero); // stamperà 10
8 System.out.println(numeroVirgola); // stamperà 3.14
9 System.out.println(carattere); // stamperà A
10 System.out.println(stringa); // stamperà Ciao Mondo
11 System.out.println(veroOFalso); // stamperà true
```





## 3 If

In Java, l'istruzione if viene utilizzata per eseguire un blocco di codice solo se una condizione specificata è vera. Ad esempio, potremmo utilizzare un'istruzione if per stampare un messaggio solo se un numero è maggiore di 10.

La sintassi per utilizzare un'istruzione if in Java è la seguente:

```
1 if (condizione) {  
2     // codice da eseguire se la condizione é vera  
3 }
```

Ad esempio, per stampare un messaggio solo se un numero è maggiore di 10, potremmo scrivere:

```
1 int numero = 15;  
2  
3 if (numero > 10) {  
4     System.out.println("Il numero é maggiore di 10");  
5 }
```

Questo codice stamperà il seguente messaggio:

```
1 Il numero é maggiore di 10
```



## 4 Switch

In Java, l'istruzione switch viene utilizzata per eseguire un blocco di codice specifico in base al valore di una variabile. Ad esempio, potremmo utilizzare un'istruzione switch per stampare un saluto in base al giorno della settimana.

La sintassi per utilizzare un'istruzione switch in Java è la seguente:

```
1  switch (variabile) {
2      case valore1:
3          // codice da eseguire se la variabile é uguale a valore1
4          break;
5      case valore2:
6          // codice da eseguire se la variabile é uguale a valore2
7          break;
8      ...
9      default:
10         // codice da eseguire se la variabile non é uguale a nessun
           valore specificato
11 }
```

Ad esempio, per stampare un saluto in base al giorno della settimana, potremmo scrivere:

```
1  int giorno = 3; // supponiamo che oggi sia mercoledì
2
3  switch (giorno) {
4      case 1:
5          System.out.println("Lunedì");
6          break;
7      case 2:
8          System.out.println("Martedì");
9          break;
10     case 3:
11         System.out.println("Mercoledì");
12         break;
13     case 4:
14         System.out.println("Giovedì");
15         break;
16     case 5:
17         System.out.println("Venerdì");
18         break;
19     case 6:
20         System.out.println("Sabato");
```

```
21         break;
22     case 7:
23         System.out.println("Domenica");
24         break;
25     default:
26         System.out.println("Valore non valido");
27 }
```

Questo codice stamperà il seguente saluto:

```
1 Mercoledì
```

## 5 For

In Java, il ciclo for viene utilizzato per eseguire un blocco di codice ripetutamente per un numero specifico di volte. Ad esempio, potremmo utilizzare un ciclo for per stampare tutti gli elementi di una matrice.

La sintassi per utilizzare un ciclo for in Java è la seguente:

```
1 for (inizializzazione; condizione; incremento) {  
2     // codice da eseguire per ogni iterazione del ciclo  
3 }
```

Ad esempio, per stampare ogni elemento di una matrice di numeri interi, potremmo scrivere:

```
1 int[] numeri = {1, 2, 3};  
2  
3 for (int i = 0; i < numeri.length; i++) {  
4     System.out.println(numeri[i]);  
5 }
```

Questo codice stamperà i seguenti numeri:

```
1 1  
2 2  
3 3
```



## 6 While

In Java, il ciclo while viene utilizzato per eseguire un blocco di codice ripetutamente finché una condizione specificata è vera. Ad esempio, potremmo utilizzare un ciclo while per stampare i numeri da 1 a 10.

La sintassi per utilizzare un ciclo while in Java è la seguente:

```
1 while (condizione) {  
2     // codice da eseguire finché la condizione é vera  
3 }
```

Ad esempio, per stampare i numeri da 1 a 10, potremmo scrivere:

```
1 int numero = 1;  
2  
3 while (numero <= 10) {  
4     System.out.println(numero);  
5     numero++;  
6 }
```

Questo codice stamperà i seguenti numeri:

```
1 1  
2 2  
3 3  
4 4  
5 5  
6 6  
7 7  
8 8  
9 9  
10 10
```





## 7 do-while

In Java, il ciclo do-while è simile al ciclo while, ma la condizione viene controllata alla fine del ciclo invece che all'inizio. Ciò significa che il blocco di codice verrà eseguito almeno una volta, indipendentemente dal fatto che la condizione sia vera o falsa.

La sintassi per utilizzare un ciclo do-while in Java è la seguente:

```
1 do {  
2     // codice da eseguire almeno una volta  
3 } while (condizione);
```

Ad esempio, per stampare i numeri da 1 a 10, potremmo scrivere:

```
1 int numero = 1;  
2  
3 do {  
4     System.out.println(numero);  
5     numero++;  
6 } while (numero <= 10);
```

Questo codice stamperà i seguenti numeri:

```
1 1  
2 2  
3 3  
4 4  
5 5  
6 6  
7 7  
8 8  
9 9  
10 10
```



## 8 Array

In Java, un array è una struttura dati che permette di archiviare una sequenza di valori di uno stesso tipo. Ogni elemento dell'array ha un indice, che permette di accedervi.

Ecco un esempio di come dichiarare un array di interi in Java:

```
1 int[] numeri = new int[5];
```

In questo esempio, abbiamo dichiarato un array di interi chiamato `numeri`, che può contenere fino a 5 elementi. Tutti gli elementi vengono inizializzati a 0 per impostazione predefinita.

Per accedere a un elemento dell'array, utilizziamo l'indice dell'elemento tra parentesi quadre dopo il nome dell'array. Ad esempio, per accedere al primo elemento dell'array `numeri`, useremmo `numeri[0]`.

```
1 numeri[0] = 1;  
2 numeri[1] = 2;  
3 numeri[2] = 3;  
4 numeri[3] = 4;  
5 numeri[4] = 5;
```

In questo esempio, abbiamo assegnato il valore 1 al primo elemento dell'array, il valore 2 al secondo elemento, e così via.

È anche possibile inizializzare un array con valori specifici al momento della dichiarazione, come mostrato di seguito:

```
1 int[] numeri = {1, 2, 3, 4, 5};
```

In questo esempio, abbiamo dichiarato un array di interi chiamato `numeri` e lo abbiamo inizializzato con i valori 1, 2, 3, 4 e 5.

È importante notare che, una volta che un array viene creato, la sua dimensione non può essere modificata. Quindi, se si desidera aggiungere o rimuovere elementi dall'array, è necessario creare un nuovo array con la dimensione appropriata e copiare gli elementi dell'array originale nel nuovo array.



## 9 Stringhe

In Java, una stringa è una sequenza di caratteri. Le stringhe vengono utilizzate per rappresentare testo, ad esempio una parola o una frase.

Per creare una stringa in Java, si utilizza la sintassi seguente:

```
1 String stringa = "Questa é una stringa";
```

Le stringhe possono essere concatenate utilizzando l'operatore +, ad esempio:

```
1 String saluto = "Ciao";  
2 String nome = "Mondo";  
3 String saluto_completo = saluto + " " + nome;  
4  
5 System.out.println(saluto_completo); // stamperà "Ciao Mondo"
```

