



# Utilizzare oggetti

## ▼ INDICE

[0 Introduzione](#)

[1 Sintassi java](#)

[1.1 Tipi di variabili](#)

[1.2 Cicli in Java](#)

[1.3 Metodi Java](#)

[1.4 Java OOP](#)

[Esercizi](#)

## 0 Introduzione

Java è un popolare linguaggio di programmazione, creato nel 1995.

È di proprietà di Oracle e più di 3 miliardi di dispositivi eseguono Java.

Java viene usato per:

- Applicazioni mobili(specialmente Android)
- Applicazioni desktop
- Applicazioni web
- Server web e server applicativi
- Giochi
- Connessione alla banca dati
- ecc...

## 1 Sintassi java

Prima di tutto dobbiamo conoscere la sintassi di Java per poter iniziare a programmare:

```
public class main{  
    public static void main(string[] args){  
        System.out.println("Hello world");  
    }  
}
```

Come possiamo vedere abbiamo una prima classe chiamata `Main` in questo caso.



### **NOTA:**

Il nome del file deve sempre corrispondere al nome della classe.

Successivamente avremo il metodo `main()` , metodo fondamentale per poter eseguire il proprio codice, e successivamente `System.out.println()` che è la stampa.

## 1.1 Tipi di variabili

Le variabili sono contenitori per la memorizzazione di valori di dati. In Java esistono diversi tipi di variabili, ad esempio:

- String
- Int
- Float
- Char
- Boolean

Per creare una variabile, è necessario specificare il tipo e assegnarle un valore, utilizzando l'operatore `=` che non significa “uguaglia” ma “diventa”:

```
tipo nomevariabile = value;
```

Andiamo a vedere degli esempi:

```
String = "Marco"; //String
int numero = 5; //Integer
float floatNum = 5.99f //Floating
char lettera = "D" //Char
boolean bool = true; //Boolean
```

## 1.2 Cicli in Java

---

In Java esistono 2 tipi di cicli:

- **While**: che va a ciclare finché una condizione specificata è **true**:

```
int i=0;
while (i<5) {
    System.out.println(i);
    i++;
}
```

- **For**: si usa quando si sa esattamente quante volte eseguire il ciclo:

```
for(int i=0;i<5;i++){
    System.out.println(i);
}
```

## 1.3 Metodi Java

---

Un **metodo** è un blocco di codice che viene eseguito solo quando viene chiamato.

È possibile passare dati, noti come **parametri**, in un metodo.

Possiamo dire che esistono 2 tipi di metodo, il quale si contraddistinguono tra:

- **Void**: ossia quei metodi che non restituiscono nessun valore;

```
public class Main {
    static void myMethod(String fname, int age) {
        System.out.println(fname + " ha " + age + " anni");
    }
}
```

```

public static void main(String[] args) {
    myMethod("Marco", 5);
    myMethod("Luca", 8);
    myMethod("Antonio", 31);
}
}

```

- **Int, char ecc...**: ossia quei metodi che restituiscono un valore.

```

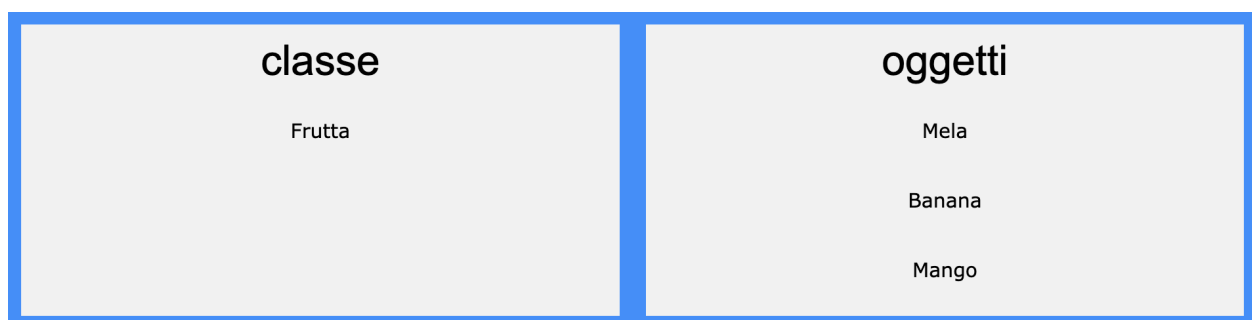
public class Main {
    static int MyMethod(int value) {
        int sum = 0;
        for(int i=0;i<=10;i++){ sum += i;}
        return sum;
    }
    public static void main(String[] args) {
        int value = 0;
        System.out.println(MyMethod(value));
    }
}

```

## 1.4 Java OOP

OOP sta per programmazione orientata ad oggetti. Le **classi e gli oggetti** sono i due aspetti principali della programmazione orientata ad oggetti.

Proviamo a fare un esempio per distinguere classe e oggetti:



Quindi una classe è un modello per oggetti e un oggetto è un'istanza di una classe.

Proviamo a vedere ora un esempio:

```
class Second{
    int x = 5;
}
public class Main {

    public static void main(String[] args) {
        Second object = new Second();
        System.out.println(5+ object.x);
    }
}
```

Sapendo questo possiamo introdurre i **costruttori** in java.

Un costruttore in Java è un **metodo speciale** utilizzato per inizializzare gli oggetti. Il costruttore viene chiamato quando viene creato un oggetto di una classe. Può essere utilizzato per impostare i valori iniziali per gli attributi dell'oggetto.



### **ATTENZIONE:**

Il costruttore deve **SEMPRE** avere lo stesso nome della classe dove viene creato.

Ad esempio il codice di prima con l'uso di un costruttore diventa:

```
class Second{
    int x;
    public Second(){
        x = 5;
    }
}
```

```
    }  
}  
public class Main {  
  
    public static void main(String[] args) {  
        Second object = new Second();  
        System.out.println(5+ object.x);  
    }  
}
```