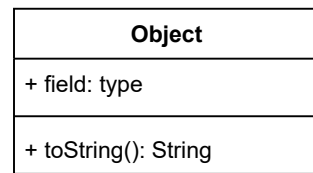
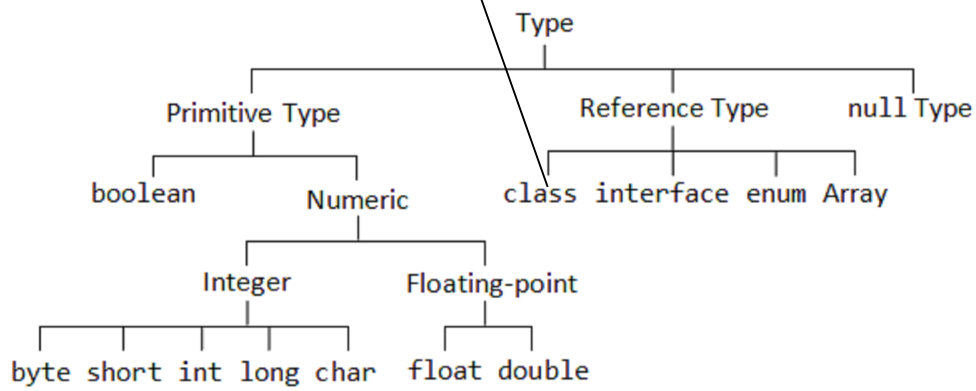
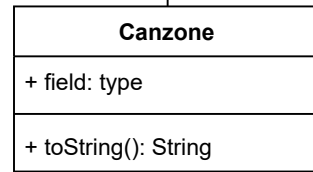
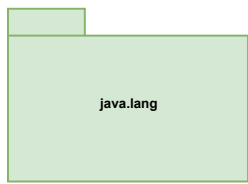


Class → Object



Extends





## Package java.lang

---

- Il package java.lang è il package più importante dell'API di Java, in quanto contiene moltissime classi e interfacce fondamentali per la programmazione Java, tanto che questo package viene importato in automatico in tutti i programmi.

Astrazioni di classe, oggetto, sistema, ...

- [Object](#)
- [System](#)
- Package
- Class
- ClassLoader
- ClassValue

[Classi wrapper](#) e gestione tipi

- Boolean
- Byte
- Character
- Double
- Float
- Integer
- Long
- Short
- Void
- Enum

Stringhe

- [String](#)
- [StringBuffer](#)
- [StringBuilder](#)

Matematica

- [Math](#)
- StrictMath
- Number

Altre funzionalità

- Compiler
- Process
- Runtime
- SecurityManager
- StackTraceElement
- Thread
- Throwable

## il metodo main()

*text file named HelloWorld.java*

*name*

*main() method*

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        // Prints "Hello, World" in the terminal window.
        System.out.print("Hello, World");
    }
}
```

*statements*

*body*

The diagram illustrates the structure of a Java program. It shows a code snippet for a class named HelloWorld with a main method. Annotations with arrows point to various parts of the code: 'text file named HelloWorld.java' points to the entire code block; 'name' points to the class name 'HelloWorld'; 'main() method' points to the 'main' method signature; 'statements' points to the code inside the main method's curly braces; and 'body' points to the entire main method block, including its signature and body.

```
Automobile a = new Automobile ();  
Motorcycle m = new Motorcycle ();  
String s = new String ("ABC");
```

**costruire gli oggetti**

## Esempio (costruttori con overloading)

```
Class Finestra {  
    String titolo;  
    String colore;  
    // Finestra senza titolo nè colore  
    Finestra () {  
        ...  
    }  
    // Finestra con titolo senza colore  
    Finestra (String t) {  
        ...  
        titolo = t;  
    }  
    // Finestra con titolo e colore  
    Finestra (String t, String c) {  
        ...  
        titolo = t; colore = c;  
    }  
}
```

Bicicletta
+ field: type
+ method(type): type



new



