

# **Programmazione Orientata agli Oggetti**

Appunti delle Lezioni di Programmazione Orientata agli Oggetti

*Anno Accademico: 2023/24*

*Giacomo Sturm*

*Dipartimento di Ingegneria Civile, Informatica e delle Tecnologie Aeronautiche  
Università degli Studi “Roma Tre”*

Sorgente del file LaTeX disponibile al seguente link:

<https://github.com/00Darxk/Programmazione-Orientata-agli-Oggetti>

## Indice

<b>1</b>	<b>Introduzione al Linguaggio Java</b>	<b>1</b>
1.1	Rapporto con il Linguaggio C . . . . .	1

## 1 Introduzione al Linguaggio Java

Java venne introdotto nel 1995, progettato per programmare dispositivi embedded, ma velocemente si utilizzò per scopi diversi da quelli descritti dai suoi creatori. Il team di Java sviluppò un sistema di embedded per i browser, le applet, delle piccole applicazioni che potevano essere eseguite all'interno del browser. Questo fu uno dei primi motivi per il successo di Java, un'altro aspetto importante per la diffusione di Java fu la sua caratteristica di semplificare aspetti di C e C++. Utilizzando un linguaggio più semplice diminuiva il costo associato ad un programmatore, per cui l'adozione di Java venne trainata dalle aziende alla ricerca di abbassare gli stipendi dei suoi dipendenti. In seguito venne utilizzato per creare la macchina virtuale su cui operano tutti i dispositivi Android.

Java è stato ideato con la capacità di mantenere la retro-compatibilità, nonostante le spinte di innovazione tecnologica che il linguaggio ha sostenuto negli anni. Nonostante la sua età Java è uno dei linguaggi di programmazione più usati dalle aziende. La diffusione più importante di Java si ebbe tra il '95 ed il '98.

Java fu il primo progetto industriale che introdusse un nuovo concetto di portabilità su scala industriale, un processore virtuale, poiché non viene progettato per un calcolatore fisico, ma la piattaforma Java Virtual Machine (JVM), per cui uno stesso programma Java può essere eseguito su tutte le piattaforme senza dover ricompilare nuovamente il codice per quella specifica piattaforma. Questa rappresenta una rivoluzione in ambito industriale, il codice oggetto Java "bytecode" prodotto dalla compilazione, come un file di estensione `.class`. Questo bytecode può quindi essere eseguito da qualsiasi piattaforma che dispone di un'implementazione della JVM. Invece programmi creati da un linguaggio più vecchio come C, producono un codice oggetto che presenta istruzioni macchina uniche alla piattaforma in cui è stato compilato. Astruendo il processore fisico si è reso possibile la creazione di programmi che non devono essere ricompilati per ogni piattaforma su cui deve essere eseguito.

Gli applet realizzati tramite Java, venivano eseguiti dalla JVM, e permise la creazione di primi siti web, rivoluzionando il paradigma di programmazione dell'epoca. La sintassi e la semantica del linguaggio sono descritte in un documento noto come Java Language Specification, per risolvere eventuali ambiguità. Inoltre fu uno dei primi linguaggi ad aver introdotto librerie incluse insieme al compilatore, sulla piattaforma Java.

### 1.1 Rapporto con il Linguaggio C

Il linguaggio C viene descritto come un linguaggio ad un medio livello di programmazione, di alto livello rispetto all'assembly, e permette di realizzare programmi ad alte prestazioni, utilizzato per scrivere sistemi operativi. Java invece venne introdotto in un ambiente diverso rispetto al linguaggio C, per cui si indica come un linguaggio di alto livello di astrazione, non è un linguaggio che cerca le migliori prestazioni, ma cerca di creare programmi semplici e portabili. Nel periodo in cui venne introdotto Java l'andamento delle prestazioni dei calcolatori permise di realizzare linguaggi che non avessero come scopo principale il risparmiare risorse del sistema. Il linguaggio C++ permette di avere prestazioni simili al linguaggio C, ma introducendo paradigmi realizzati in Java. Ma si preferisce per realizzare programmi ad alte prestazioni il C, rispetto al C++, poiché fornisce troppe scelte, complicando l'implementazione e l'uso del linguaggio.

In Java una dichiarazione di una variabile comprende la sua inizializzazione ad un valore nullo. Le dichiarazioni e assegnazioni in Java si ottengono mediante la stessa sintassi del linguaggio C. Si introduce un tipo **boolean** utilizzato per rappresentare i due possibili valori booleani **true** e **false**. Le descrizioni di tutti i tipi utilizzati in Java vengono definite nel JLS. Entrambi sono due linguaggi staticamente tipati, ovvero a tempo di compilazione deve essere noto il tipo di ogni dato utilizzato nel programma. Java introduce il tipo **String** utilizzato per rappresentare stringhe, non è un dato primitivo, e permette l'uso molto più semplice delle stringhe rispetto a C. Rappresenta un tipo particolare di classe, in appoggio al compilatore per favorire la loro gestione da parte del compilatore. Le stringhe vengono inizializzate a letterali stringhe, contenuti tra due doppi apici " ". Da Java 13 è possibile scrivere stringhe letterali multi-linea, utilizzando tripli doppi apici """.

La sintassi di controlli di flusso è esattamente la stessa del linguaggio C, ma le condizioni sono diverse a causa del tipo booleano.

Una grande differenza tra i due linguaggi consiste nella diversa diagnostica a tempo di compilazione e di esecuzione degli errori, poiché in Java i gli errori ed i messaggi di errore forniscono informazioni molto utili per la sua risoluzione. Gli errori a tempo di compilazione sono altrettanto efficaci in Java quanto in C, ma la differenza principale consiste negli errori a tempo di esecuzione tra i due linguaggi, poiché in C, in molte piattaforme, viene fornito un errore estremamente generico, in base alla versione di C, del compilatore, e da altre condizioni.