



1 – Introduzione a Java

Maurizio Franco



Introduzione a Java

Cosa è Java?

Introduzione a Java

Java è una tecnologia.

Java è un linguaggio di programmazione.

Introduzione a Java

**Come nasce,
come si evolve...**

Introduzione a Java

1985 – Nasce a partire da ricerche alla Stanford University

1992 – Acquisito e prodotto dalla Sun Microsystems (linguaggio OAK)

1995 – Battezzato in "Java"

1997 – Rilasciato sotto licenza GPL

2010 – Oracle acquisisce Sun

Introduzione a Java

Perché Java

Introduzione a Java

Durante la sua nascita, l'obiettivo era creare un "super-linguaggio", un "linguaggio ideale" che superasse i limiti degli altri linguaggi esistenti. Un linguaggio che fosse in grado di fornire applicazioni distribuite in rete e piattaforma indipendenti.

Quanti apparecchi elettronici, oggi, utilizzano applicazioni java?

Introduzione a Java

Caratteristiche di Java

Introduzione a Java

Sintassi – semplice e simile al C e C++

Gratuito – scrivere applicazioni non bisogna pagare nessuna licenza

Robusto – gestione delle eccezioni, gestione automatica della memoria allocata

Architettura indipendente – “scrivi una volta gira sempre”

Orientato agli oggetti – ereditarietà, incapsulamento e polimorfismo

Sicuro

Introduzione a Java

Il primo ambiente di sviluppo

Introduzione a Java

JDK – Java Development Kit

Offre tutto il necessario supporto tecnologico
per la creazione di applicazioni Java.
Un compilatore, una Java Virtual Machine, ed
una serie di tools utili per lo sviluppatore.

Scarichiamola ed installiamola...

Introduzione a Java

Un semplice editor di testo.

Indipendentemente dal sistema operativo che stiamo utilizzando basterà un semplice editor di testo per scrivere la nostra prima applicazione Java.

Introduzione a Java

Struttura della JDK

Introduzione a Java

Una volta installata la JDK avrà formato una gerarchia di cartelle di questo tipo:

- bin: contiene tutti i file eseguibili della JDK, quindi tutti i tool di cui parlavamo in precedenza, utili allo sviluppatore.
- demo: contiene alcune mini applicazioni dimostrative.
- include e lib: contengono una serie di librerie scritte in C e in Java che sono utilizzate dal JDK.
- jre: sta per Java Runtime Environment (JRE). Si tratta della JVM, che una volta lanciata, interpreta il codice compilato e lo esegue, dando "vita" alla nostra applicazione.
- docs: questa cartella deve essere scaricata ed installata a parte e contiene la documentazione delle librerie standard di Java, più vari tutorial.

Inoltre, nella cartella principale, oltre a vari file di servizio (licenza, copyright etc...) ci sarà anche un file di nome "src.zip".

Una volta scompattato "src.zip", sarà possibile dare uno sguardo ai file sorgenti (i ".java") delle librerie fornite con la JDK.

Settaggio delle variabili di ambiente

Introduzione a Java

SETTAGGIO DELLE VARIABILI D'AMBIENTE:

JAVA_HOME

JRE_HOME

PATH

CLASSPATH

Introduzione a Java

I passi della prima applicazione

Introduzione a Java

- Apertura del semplice editor di testo.
 - Scrittura del codice
- Salvataggio in un file con estensione .java
 - Compilazione del file
- Esecuzione dell'applicazione

Introduzione a Java

Come si esegue la nostra prima applicazione?

Da riga di comando digitiamo:

```
java Prova
```

Questo comando lancerà la JVM dandogli come input la nostra applicazione, il file Prova.class

Introduzione a Java

La nostra prima applicazione

Introduzione a Java

Classico esempio di prima piccola applicazione...

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

Introduzione a Java

Come si compila un file .java?

Da riga di comando digitiamo:

```
javac Prova.java
```

Questo comando compilerà il nostro file java in codice, bytecode.

Verrà quindi prodotto un file con estensione .class
Prova.class

Introduzione a Java

Come si esegue la nostra prima applicazione?

Da riga di comando digitiamo:

```
java Prova
```

Questo comando lancerà la JVM dandogli come input la nostra applicazione, il file Prova.class

Introduzione a Java

Analisi dell'applicazione "Prova.java"

Introduzione a Java

```
public class Prova {}
```

Dichiarazione della classe Prova, pubblica.
Tutto il codice contenuto fra le due parentesi graffe è il
corpo della classe Prova.

Il nome della classe pubblica Prova, deve
obbligatoriamente corrispondere al nome del file.

Introduzione a Java

```
public static void main(String[] args) {}
```

Dichiarazione del metodo main.

Il metodo è fondamentalmente un'azione di una classe.

Il metodo main, in particolare, indica l'entry point(il punto iniziale) di un'applicazione.

Le istruzioni public, static e void sono dei modificatori.

Le istruzioni contenute fra le parentesi tonde(String[] args) indicano un parametro di input.

Tutto ciò che è contenuto fra le parentesi graffe compone il corpo del metodo.

Introduzione a Java

```
System.out.println>Hello World!);
```

Comando che stampa a video la stringa "Hello World!".

Problemi in fase di compilazione ed esecuzione

Introduzione a Java

Una delle caratteristiche positive di java è la robustezza.

In particolare, la compilazione, è una fase che mette subito alla prova il codice scritto dallo sviluppatore.

E' bene iniziare già da subito a leggere e capire i messaggi di errore che vengono generati all'esecuzione del comando `javac`.

Messaggi di errore in fase di compilazione

Introduzione a Java

Javac: Command not found

Il comando javac si trova nella directory bin dell'installazione di java.

Il non riconoscimento del comando indica che, non è stato indicato nel PATH dell'ambiente, il percorso per il corretto puntamento alla suddetta cartella bin.

Introduzione a Java

```
Prova.java:6: cannot resolve symbol  
symbol : method printl (java.lang.String)  
location: class java.io.PrintStream  
System.out.printl("Hello World!");
```

Il compilatore ci indica che non trova il metodo `printl` all'interno della classe `PrintStream`.

In realtà abbiamo semplicemente sbagliato(in questo caso) a scrivere il nome del metodo da chiamare: `printl` invece di `println`

Introduzione a Java

Prova.java:1: 'class' or 'interface' expected
Class Prova

Abbiamo scritto il modificatore class con la c
maiuscola.

Messaggi di errore in fase di interpretazione

Introduzione a Java

Exception in thread "main" java.lang.NoSuchMethodError: main

Abbiamo omesso il metodo main dalla classe che stiamo invocando per la nostra applicazione.
O magari abbiamo semplicemente omesso qualche modificatore "necessario" nella sua firma.

Introduzione a Java

```
Exception in thread "main" java.lang.NoClassDefFoundError: Provaaaa (wrong
                                name: Prova)
    at java.lang.ClassLoader.defineClass0(Native Method)
    at java.lang.ClassLoader.defineClass(ClassLoader.java:486)
    at java.security.SecureClassLoader.defineClass(SecureClassLoa
                                [.....])
```

Non abbiamo dato alla classe il nome corretto.
Vale a dire lo stesso nome che abbiamo dato al file.

Introduzione a Java

Ambienti di sviluppo complessi

Introduzione a Java

Una volta capita la logica della scrittura di una classe java e di una piccola applicazione java, lo sviluppatore passerà ad un ambiente di sviluppo più complesso.

Definiti IDE, gli Integrated Development Environment, sono ambienti di sviluppo che integrano al loro interno una serie di funzionalità come la compilazione, l'esecuzione dell'applicazione, il debug, il test, e via dicendo....

Introduzione a Java

Fra i più comuni IDE in circolazione citiamo:
Eclipse, NetBeans, JDeveloper della Oracle,
JBuilder della Borland, IBM Rational
Application Developer, ecc..

Conclusioni

Introduzione a Java

In questo modulo abbiamo introdotto java come tecnologia e linguaggio. Abbiamo accennato alla sua evoluzione, ed abbiamo visto come installare un primo e semplice ambiente di sviluppo, nonché scritto, compilato ed eseguito la nostra prima applicazione.