

Corso Java

16/01/2018

redatto da:
Giuseppe ing. Grosso
16 gennaio 2018

Indice generale

| | |
|--|---|
| Lezione1..... | 3 |
| installazione ambiente di sviluppo..... | 3 |
| Installazione eclipse oxygen..... | 3 |
| installazione java cm (jdk 8)..... | 3 |
| Differenze tra jdk e jre..... | 4 |
| Esempio di programma in java con metodo main..... | 4 |
| Compilazione ed esecuzione in eclipse..... | 4 |
| Tipi di dato..... | 4 |
| Concetto di Classe o oggetto e istanza..... | 5 |
| Programmazione strutturata: interfaccia e classe astratta..... | 5 |

Lezione1

installazione ambiente di sviluppo.

Per poter sviluppare in java occorre effettuare alcune installazioni che di seguito vengono evidenziate:

installazione IDE di sviluppo: eclipse (oxygen) release a giugno 2017.

Installazione eclipse oxygen.

Collegarsi al sito:

<http://www.eclipse.org/downloads/eclipse-packages/>

e scaricare il pacchetto



Effettuato il download effettuare doppio click sul pacchetto scaricato ed installarlo.

installazione java cm (jdk 8).

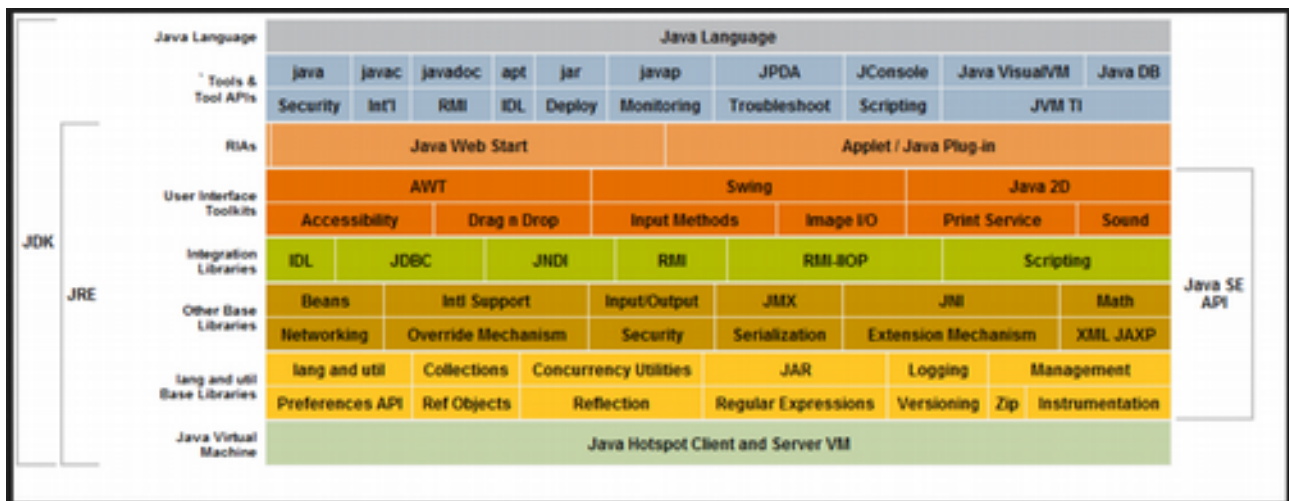
Se non presente la jdk, occorre scaricarla dal sito di oracle:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

| Product / File Description | File Size | Download |
|-------------------------------------|-----------|---|
| Linux ARM 32-bit Hard Float ABI | 77.92 MB | jdk-8u161-linux-arm32-vfp-hflt.tar.gz |
| Linux ARM 64-bit Hard Float ABI | 74.88 MB | jdk-8u161-linux-arm64-vfp-hflt.tar.gz |
| Linux x86 | 168.96 MB | jdk-8u161-linux-i586.rpm |
| Linux x86 | 183.76 MB | jdk-8u161-linux-i586.tar.gz |
| Linux x64 | 166.09 MB | jdk-8u161-linux-x64.rpm |
| Linux x64 | 180.97 MB | jdk-8u161-linux-x64.tar.gz |
| macOS | 247.12 MB | jdk-8u161-macosx-x64.dmg |
| Solaris SPARC 64-bit (SVR4 package) | 139.99 MB | jdk-8u161-solaris-sparcv9.tar.Z |
| Solaris SPARC 64-bit | 99.29 MB | jdk-8u161-solaris-sparcv9.tar.gz |
| Solaris x64 | 140.57 MB | jdk-8u161-solaris-x64.tar.Z |
| Solaris x64 | 97.02 MB | jdk-8u161-solaris-x64.tar.gz |
| Windows x86 | 198.54 MB | jdk-8u161-windows-i586.exe |
| Windows x64 | 206.51 MB | jdk-8u161-windows-x64.exe |

Selezionare la versione di java SE in base al proprio sistema operativo e procedere con l'installazione.

Differenze tra jdk e jre.



La **JDK** è una **estensione della JRE** che contiene questa, ma è inoltre un insieme di software che possiamo utilizzare per sviluppare le applicazioni basate su Java. La Java Development Kit (JDK) è necessaria per sviluppare nuove applicazioni Java.

La differenza principale con la JRE è che contiene **un compilatore Java** per trasformare i nostri codici java in bytecode, cioè in file di tipo .class. E' possibile richiamare il compilatore Java tramite il comando **javac** da linea di comando.

Esempio di programma in java con metodo main.

In questo esempio vediamo come effettuare una semplice classe java con metodo main, ovvero richiamabile ed eseguibile direttamente all'interno dell'ambiente di sviluppo nell'ide eclipse.

Compilazione ed esecuzione in eclipse.

La compilazione del programma in eclipse è molto semplice occorre semplicemente impostare la compilazione automatica e successivamente eseguire il programma java direttamente attraverso il metodo main.

Tipi di dato.

I tipi di dati in java possono essere primitivi (vedi tipi primitivi di seguito) oppure oggetti, sia oggetti nativi che custom (vedi classi).

Fonte wikipedia.

Tipi primitivi [\[modifica \]](#)

I tipi primitivi del Java sono i seguenti:

- *boolean*: ammette i soli valori *true* e *false*.
- Tipi numerici a virgola mobile: *float* e *double*. Essi rispondono alle indicazioni dello standard [IEEE 754](#).
- Tipi numerici interi:

| | | |
|-------|-----------------------------|---|
| byte | Da -2^7 a $2^7 - 1$ | Da -128 a 127 |
| short | Da -2^{15} a $2^{15} - 1$ | Da -32768 a 32767 |
| char | Da 0 a $2^{16} - 1$ | Da 0 a 65535, ovvero da <code>'\u0000'</code> a <code>'\uffff'</code> |
| int | Da -2^{31} a $2^{31} - 1$ | Da -2147483648 a 2147483647 |
| long | Da -2^{63} a $2^{63} - 1$ | Da -9223372036854775808 a 9223372036854775807 |

Concetto di Classe o oggetto e istanza.

Una classe può essere vista come un contenitore all'interno della quale troviamo due elementi principali: **attributi** e **metodi**.

Gli attributi della classe sono quelle variabili che identificano la classe e che vengono usati dai vari metodi per compiere qualche operazione.

I metodi sono delle funzioni che espletano un certo compito.

L'istanza della classe è una realizzazione fisica della classe, quindi ad esempio due istanze sono completamente indipendenti l'una dall'altra e quindi se creiamo due oggetti di tipo "Numeri" e richiamiamo il metodo di assegnazione passando numeri diversi otterremo due oggetti diversi. Dunque la somma e la sottrazione relative ai due oggetti saranno generalmente diversi.

Programmazione strutturata: interfaccia e classe astratta.

Fino ad adesso abbiamo visto alcuni esempi di implementazioni di classi semplice, proviamo ad analizzare scenari più complessi e a dare una struttura alla programmazione introducendo il concetto di interfaccia e di classe astratta.

Molto spesso le interfacce vengono confuse con le classi astratte dato che dal punto di vista logico sono molto simili, ma le interfacce possono essere intese come un'evoluzione delle classi astratte e permettono, di fatto, di simulare l'ereditarietà multipla.

Il vantaggio principale di una interfaccia, essendo comunque una classe anche se con proprietà particolari, è quello che oltre ad essere estesa può essere **implementata**. La differenza tra estendere e implementare è molto grande in quanto una classe può essere ereditata solo ed esclusivamente una volta, mentre una classe può implementare infinite interfacce permettendo così **l'ereditarietà multipla**.

Le **classi astratte in Java** sono utilizzate per poter dichiarare caratteristiche comuni fra classi di una determinata gerarchia. Pur definendo il nome di un tipo, la classe astratta non può essere istanziata, analogamente a quanto accade per le interfacce; ma a differenza di una interfaccia può avere field non statici, metodi non pubblici, un costruttore, insomma una classe a tutti gli effetti ma non istanziabile.