Corso di base JAVA

Mauro Donadeo
mail: mauro.donadeo@gmail.com

INTRODUZIONE

HT Lab

Informazioni generali

Orari lezioni

- Martedì 27 marzo *ore 13:30 - 18:30*;
- Mercoledì 28 marzo *ore 14:00 - 19:00*;
- Venerdì 30 marzo *9:30 - 13:30*;
- Martedì 3 aprile *13:30 - 18:30*;
- Giovedì 5 aprile *13:30 - 18:30*;

Dove

Tutte le lezioni si svogeranno all'interno dell'aula CAR.

About me

Laureato in ingengeria informatica ad ottobre del 2011. Titolo della tesi: *Realizzazionde di un sistema di video conferenza 3D utilizzando il sistema di video conferenza MS Kinect*.

Posizione attuale

- Collaboro con il Prof. Gamberini all'interno di HTLab (htlab.psy.unipd.it);
- Gesture recognition su dispositivi touchless all'inteno del progetto europeo CEEDs (ceeds-project.eu);
- Misure di accuracy di più Kinect che funzionano contemporaneamente. Sempre con il fine di riconoscere gesture.

Cos'è un programma

Il computer

Tutti sappiamo che un computer è una macchina che:

- **memoriza dati** (numeri, parole, immagini suoni...);
- **interagisce con dispositivi** (schermo, tastiere, mouse, kinect...)
- **esegue programmi**.

Cos'è un programma

Il computer

Tutti sappiamo che un computer è una macchina che:

- **memoriza dati** (numeri, parole, immagini suoni...);
- **interagisce con dispositivi** (schermo, tastiere, mouse, kinect...)
- **esegue programmi**.

I programmi

I programmi sono **sequenze** di *istruzioni* che il **computer** *esegue*, e di *decisioni* che il **computer** prende per svolgere una certa attività.

Nonostante i programmi sono molto sofisticati e svolgano funzioni molto complesse, le istruzioni di cui sono composti sono **molto elementari** per esempio:

- estrarre un numero da una posizione di memoria;
- inviare un documento in stampa;
- accendere un punto rosso in una pos. determinata dello schermo;
- se un numero è negativo, allora si svolge una funzione più tosto che un altra.

Programmazione

Un programma descrive al computer in estremo dettaglio la sequenza necessaria di passi per svolgere un particolare compito:

L'attività di progettare e **realizzare un programma** *è detta* *programmazione*

Problemi

Quale dei seguenti due problemi può essere risolto da un computer:

- Dato un insieme di fotografie di paesaggi, qual'è il più rilassante?
- Avete un deposito di ventimila euro in un conto bancario che produce il 5% di interessi all'anno, capitalizzati annualmente, quanti anni occorrono affinché il saldo del conto arrivi al doppio della cifra iniziale?

Problemi

Quale dei seguenti due problemi può essere risolto da un computer:

- Dato un insieme di fotografie di paesaggi, qual'è il più rilassante?
- Avete un deposito di ventimila euro in un conto bancario che produce il 5% di interessi all'anno, capitalizzati annualmente, quanti anni occorrono affinché il saldo del conto arrivi al doppio della cifra iniziale?

Il primo problema non può essere risolto dal computer. **Perché**?

- **Un computer può risolvere soltanto problemi che potrebbero essere risolti anche manualmente**:
  - E' solo molto più veloce, non si annoia, e non fa errori (se programmato nella maniera giusta)

- **Un computer può risolvere soltanto problemi che potrebbero essere risolti anche manualmente**:
  - E' solo molto più veloce, non si annoia, e non fa errori (se programmato nella maniera giusta)

Cos'è un algoritmo

Si dice algoritmo la descrizione di un metodo di soluzione di un problema che:

- sia eseguibile;
- sia priva di ambiguità;
- arrivi ad una conclusione in un tempo finito.

- **Un computer può risolvere soltanto problemi che potrebbero essere risolti anche manualmente**:
  - E' solo molto più veloce, non si annoia, e non fa errori (se programmato nella maniera giusta)

Cos'è un algoritmo

Si dice algoritmo la descrizione di un metodo di soluzione di un problema che:

- sia eseguibile;
- sia priva di ambiguità;
- arrivi ad una conclusione in un tempo finito.

Un computer può risolvere soltanto quei problemi per i quali sia noto un algoritmo

A cosa servono gli algoritmi

- L'identificazione di un algoritmo è il requisito indispensabile per risolvere un problema con il computer;
- la scrittura di un problema con il computer consiste,in genere, nella traduzione di un algoritmo in qualche linguaggio di programmazione;

A cosa servono gli algoritmi

- L'identificazione di un algoritmo è il requisito indispensabile per risolvere un problema con il computer;
- la scrittura di un problema con il computer consiste,in genere, nella traduzione di un algoritmo in qualche linguaggio di programmazione;

Prima di scrivere un programma è necessario individuare un algoritmo

Il linguaggio di programmazione JAVA

- 1954-1957 nasce il primo linguaggio di programmazione: *FORTRAN*
- *1959 COBOL dove la B sta per Business. Infatti divenne uno dei primi linguaggi di programmazione orientato per le applicazioni business;*
- *1972 Dennis Ritchie fonda il linguaggio di programmazione C. E' molto potente come linguaggio di programmazione.*
- *Bjarne Strousturp sviluppa il C++ differente dal suo predecessore C è uno dei primi linguaggi orientato a gli oggetti che rappresenta un grande passo in avanti.*
- *1995 Sun Microsystem rilascia la prima versione ufficiale di JAVA. Aggiunge al C++ il concetto Write Once, Run Anywhere.*

Java è un linguaggio orientato agli oggetti. Cosa vuol dire?

- In un linguaggio orientato agli oggetti puoi organizzare il tuo lavoro in Oggetti e Classi.

Esempio: immaginiamo di scrivere un programma che tiene traccia delle case di un nuovo condominio che si sta costruendo.

Ogni casa è differente essenzialmente dalle altre per piccoli accorgimenti ad es.: per il suo interno, colore delle pareti, stile della cucina, tipo di bagno. Con JAVA ogni casa è un **OGGETTO**.

Java è un linguaggio orientato agli oggetti. Cosa vuol dire?

- In un linguaggio orientato agli oggetti puoi organizzare il tuo lavoro in Oggetti e Classi.

Esempio: immaginiamo di scrivere un programma che tiene traccia delle case di un nuovo condominio che si sta costruendo.

Ogni casa è differente essenzialmente dalle altre per piccoli accorgimenti ad es.: per il suo interno, colore delle pareti, stile della cucina, tipo di bagno. Con JAVA ogni casa è un **OGGETTO**.

Sebbene le case differiscono leggermente una dall'altra la lista delle caratteristiche è sempre la stessa. Quindi all'interno del programma orientato agli oggetti ci sarà questa lista che contiene tutte le caratteristiche della casa. La lista è chiamata **CLASSE**

Quindi esiste una reale relazione tra classi ed oggetti. Il programmatore definisce una classe, e dalla definizione delle classi, il computer crea oggetti individuali.

Quindi esiste una reale relazione tra classi ed oggetti. Il programmatore definisce una classe, e dalla definizione delle classi, il computer crea oggetti individuali.

Supponiamo ora che il progetto cambi! Che la casa passa da un piano a due piani. E le stanze da letto al secondo piano per metà sono delle abitazioni sono quattro e per metà sono tre.

Quindi esiste una reale relazione tra classi ed oggetti. Il programmatore definisce una classe, e dalla definizione delle classi, il computer crea oggetti individuali.

Supponiamo ora che il progetto cambi! Che la casa passa da un piano a due piani. E le stanze da letto al secondo piano per metà sono delle abitazioni sono quattro e per metà sono tre.

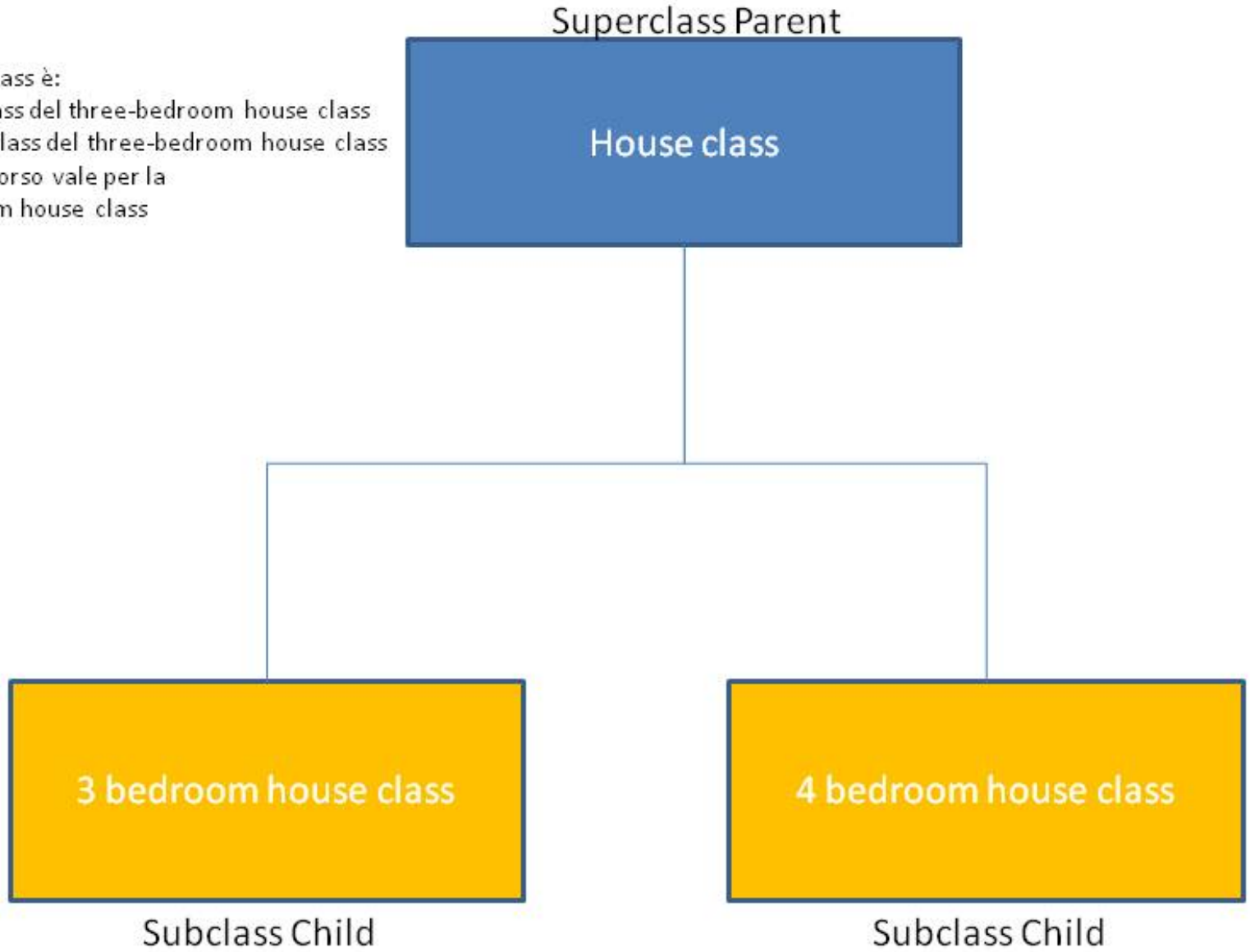
Bisogna buttare via tutto?

La potenza di Java

Non bisogna buttare via nulla. Quello che è stato progettato fino ad adesso diventerà la nostra *principale*. Sarà la nostra SUPERCLASS.

- Verrà creata una classe per le case con tre e quattro camere da letto. Che *erediteranno* le caratteristiche del progetto originale
- Si può inoltre dire che le classi appena create *estendono* la classe originale.
- In questo modo verrà creato un rapporto di parentela tra le classi. Le classi per le 3 e 4 stanze da letto sono figlie della classe originale.

La house class è:
La superclass del three-bedroom house class
La parent class del three-bedroom house class
Stesso discorso vale per la
4-beedroom house class



La four-bedroom house class.
Estende la house class. Eredita le caratteristiche di house class. È subclass della house class. È classe figlia di house class.

INSTALLAZIONE

Prerequisiti

Per installare la Java Development Kit (JDK) non sono richieste grosse prestazioni da parte della macchina. Unico prerequisito è avere i permessi anche di amministratore della macchina in quanto verranno modificate le variabili d'ambiente.

Verrà spiegato un metodo per installare la JDK per sistemi windows. Per chi utilizza Linux o Mac OsX posso dare dei consigli in privato. In linea di massima chi utilizza linux e mac comunque deve avere i permessi di amministratore della macchina.

1. Andate sul sito: `http://www.oracle.com/technetwork/java/javase/downloads/index.html` dove scaricherete la versione Standard Edition. Effettuate anche il download della documentazione di Java;
2. Double-click sulla icona della JDK;
3. Tra le varie features che ti permette di installare assicuratevi che sia selezionato
   - Development tools;
   - Public Java Runtime Enviroment.

La variabile d'ambiente PATH specifica un elenco di directory separate da punto e virgola ; in cui il sistema ricerca i file eseguibili (nell'ordine in cui sono indicate), oltre che nella directory corrente.

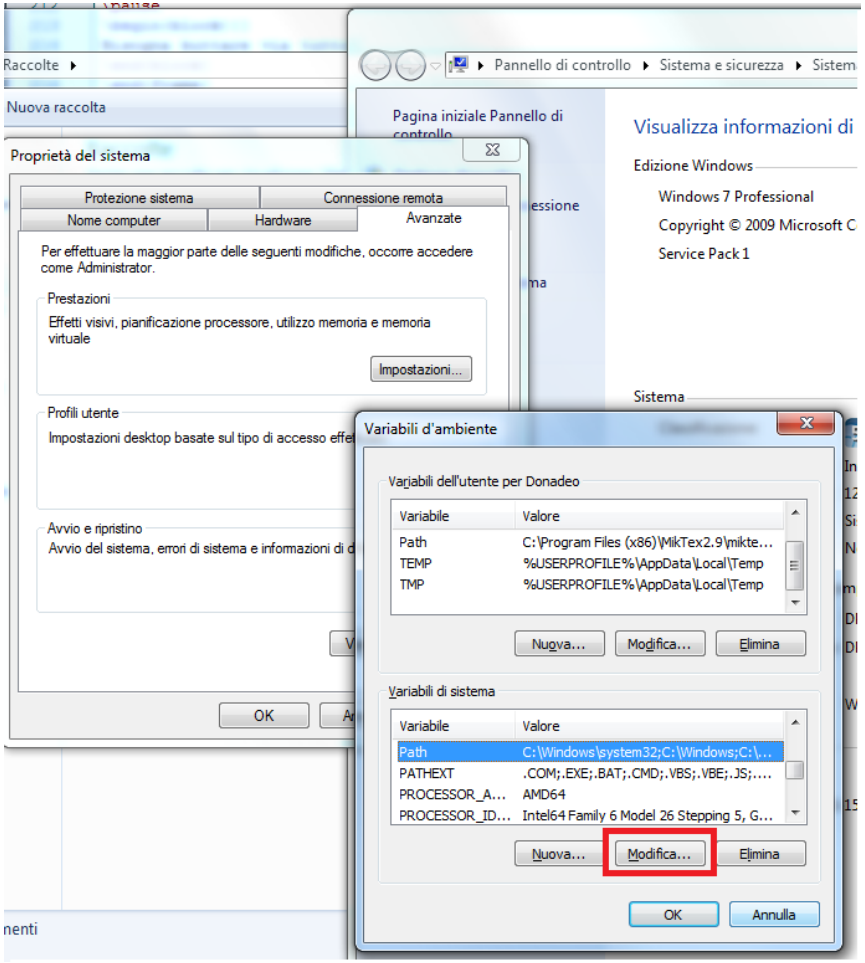


Figura: Tasto destro su computer->Proprietà->Impostazioni avanzate di sistema->Avanzate

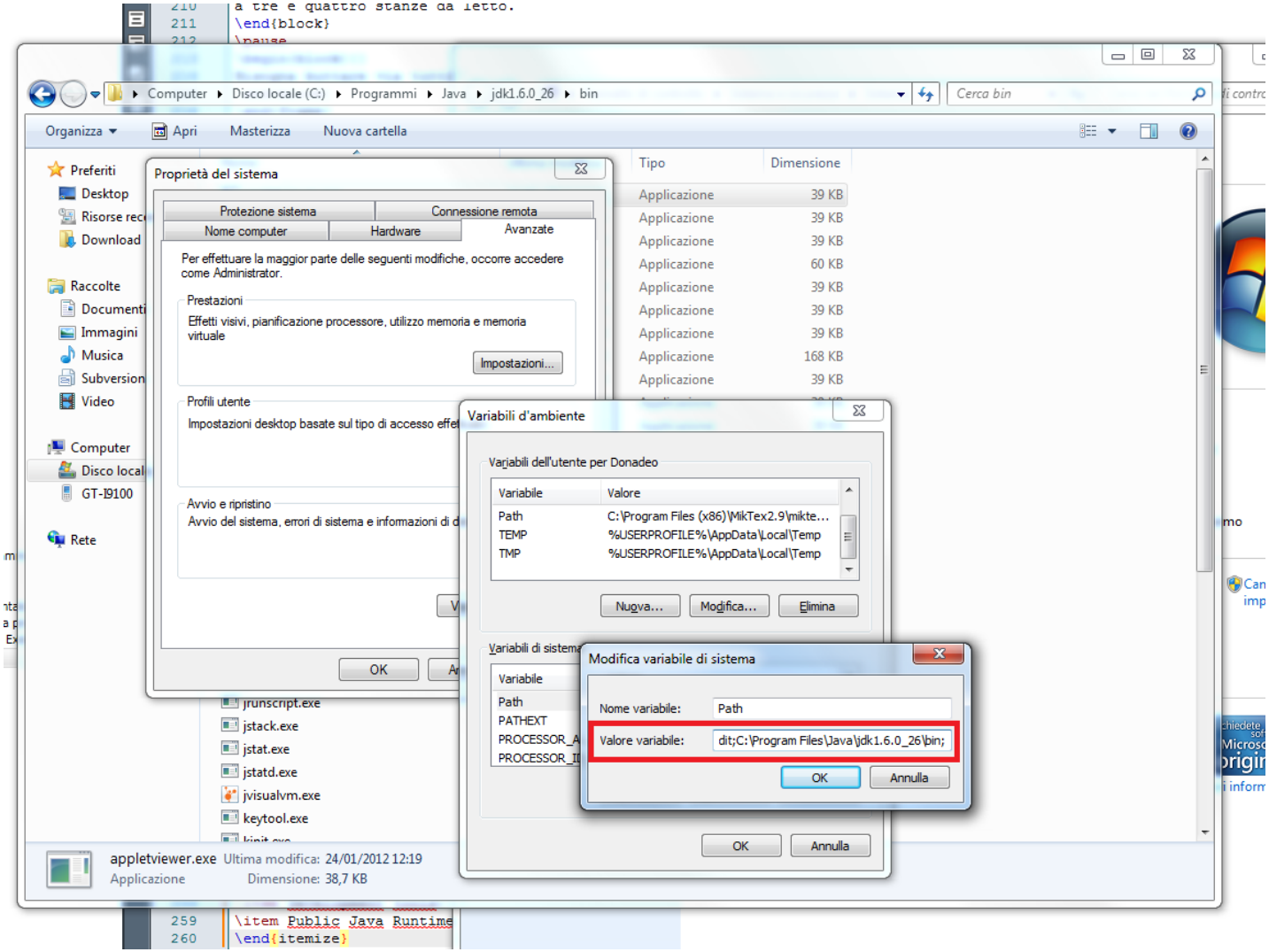


Figura: Aggiungete alla fine il percorso dove avete installato la vostra JDK. Solitamente è: C:/Programs Files/Java/jdk_ version/bin

Un altro dei punti di forza di JAVA è la documentazione. Create una cartella docs all'interno della cartella principale di Java. Solitamente `C:/Programs Files/Java/jdk version/docs`. E fate un unzip della documentazione che avete scaricato qua. Cliccate su index.html e avrete:

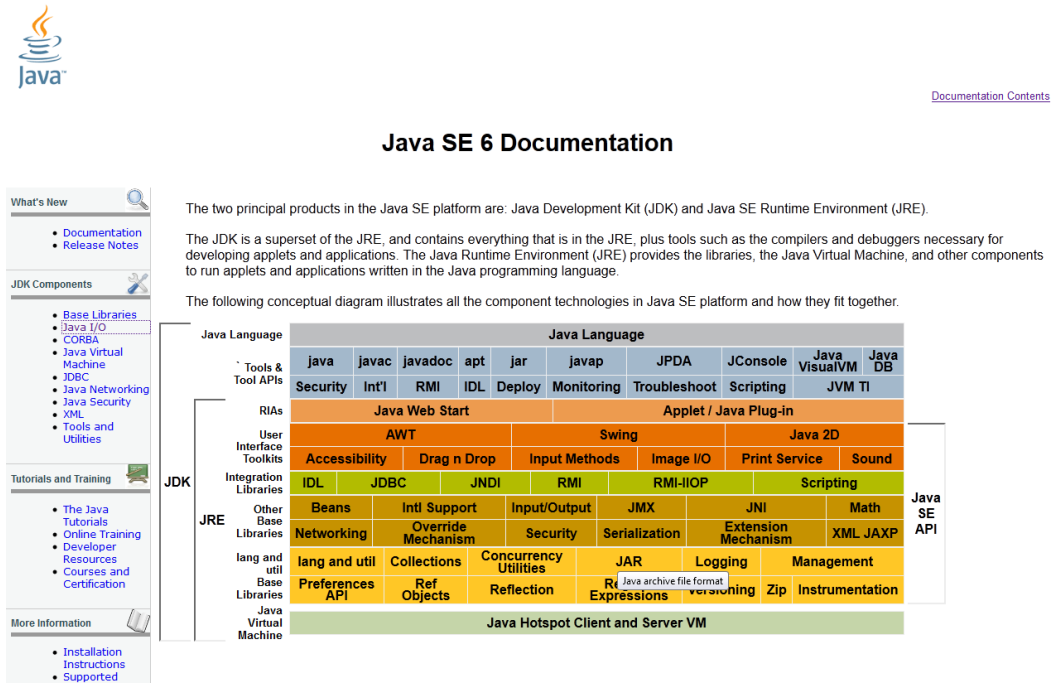


Figura: Documentazione Java. Troverete le API e la spiegazione di tutte classi già messe a disposizione da SUN.

IDE

Un *integrated development environment* IDE (altre si conosciuto come: *integrated design environment, integrated debugging environment o interactive development environment*) è un'applicazione che aiuta e facilita gli sviluppatori a scrivere del codice. Un IDE normalmente consiste in:

- Un editor di testo
- Un compilatore e/o un interprete
- Un tool che permetta di costruire in maniera automatica un eseguibile
- Un debugger.

Esistono vari IDE che possono aiutare alla progettazione del proprio software.

Figura: Geany - http://www.geany.org/

Figura: NetBeans - http://netbeans.org/

Figura: Eclipse - http://www.eclipse.org/

Le fasi della programmazione

La scrittura di un programma possiamo dire che si divide principalmente in 3 fasi:

- scrittura del programma (codice sorgente)
- compilazione del codice sorgente che porta quindi alla creazione del codice eseguibile
- esecuzione del programma.

Individuare il compilatore Java

Esistono vari modi per compilare i programmi in Java: internamente all'IDE, utilizzando i comandi da tastiera con l'ausilio di una console (**è questo il nostro caso**) o tramite delle icone sullo schermo.

cmd.exe

Utilizzando il prompt dei comandi di Windows, grazie alle variabili d'ambiente, consente di compilare ed eseguire i programmi che verranno scritti in Java. Ecco alcuni comandi utili per utilizzare:

- Per aprire il prompt dei comandi di Windows è possibile digitare all'interno di cerca cmd.exe o si può trovare un collegamento dell'applicazione all'interno di Programmi/Accessori/Prompt dei comandi.
- cd nome cartella: permette di accedere alla cartella.
- dir restituisce un elenco delle cartelle presenti all'interno della cartella di cui si lancia il comando.
- cd .. ritorna alla cartella precedente.

cmd.exe

Utilizzando il prompt dei comandi di Windows, grazie alle variabili d'ambiente, consente di compilare ed eseguire i programmi che verranno scritti in Java. Ecco alcuni comandi utili per utilizzare:

- Per aprire il prompt dei comandi di Windows è possibile digitare all'interno di cerca cmd.exe o si può trovare un collegamento dell'applicazione all'interno di Programmi/Accessori/Prompt dei comandi.
- cd nome cartella: permette di accedere alla cartella.
- dir restituisce un elenco delle cartelle presenti all'interno della cartella di cui si lancia il comando.
- cd .. ritorna alla cartella precedente.

P.S. *Se premete il tasto tab la console completerà la parola permettendovi di risparmiare tempo*

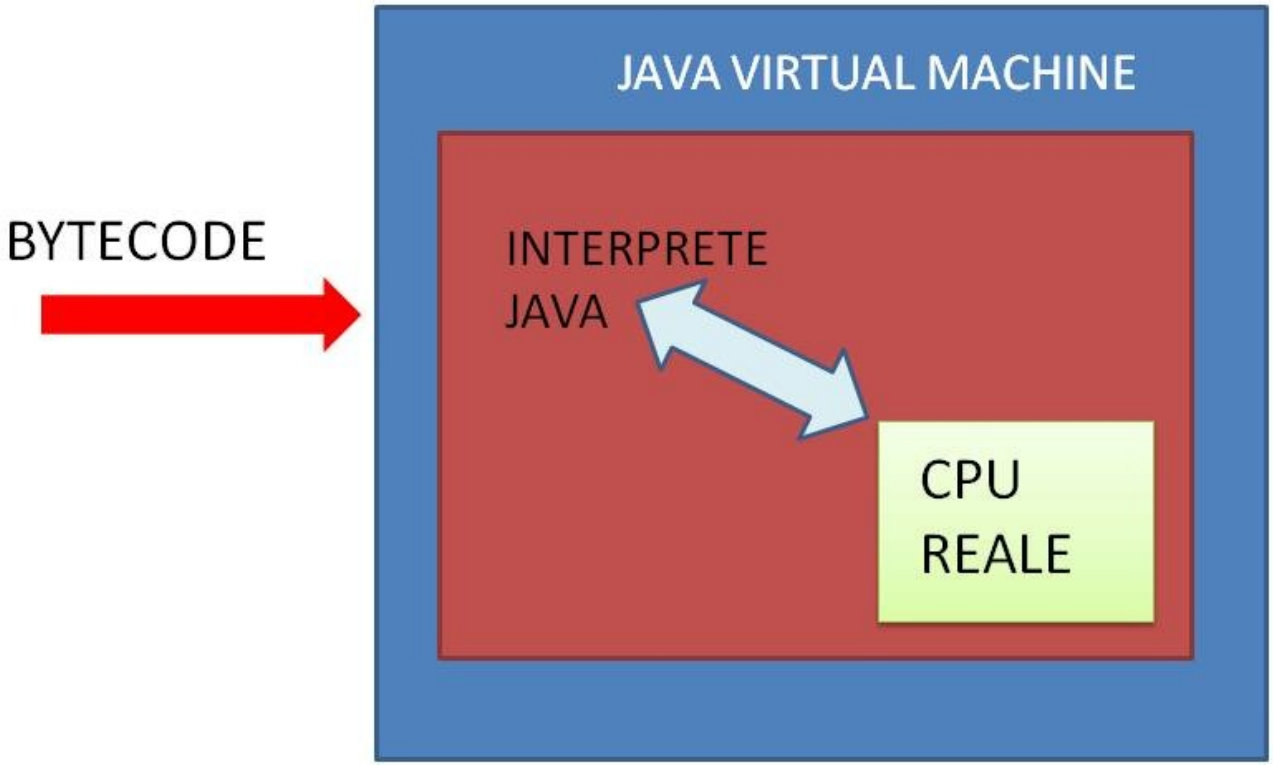
Cos'è la compilazione

La compilazione del *codice sorgente* di un programma consente di creare un particolare tipo di formato di codice eseguibile, detto **bytecode** che è il codice interpretato dalla Java Virtual Machine *JVM*.

- `javac NomeFile.java`

effettua la compilazione del programma, e genera il file NomeFile.class

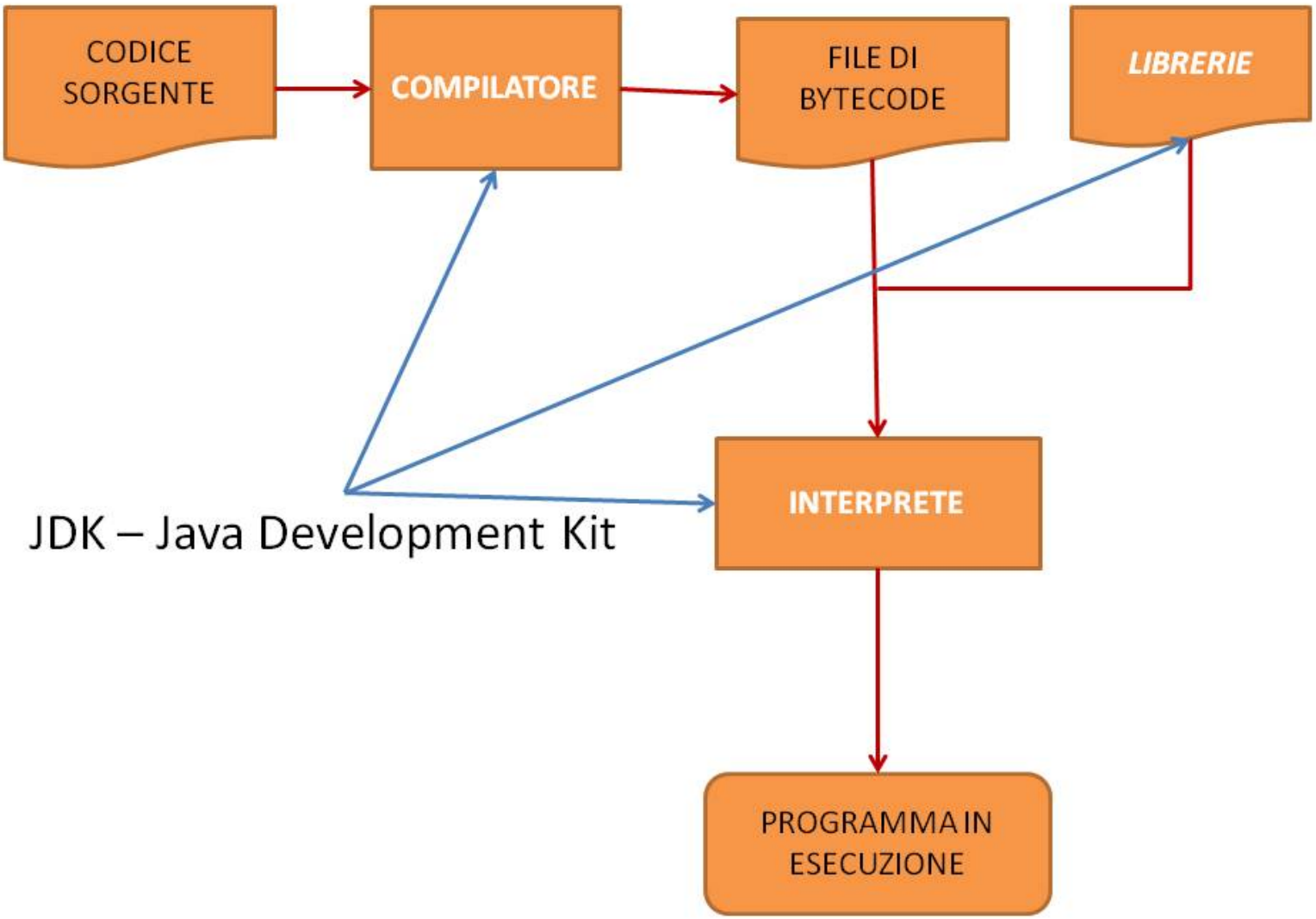
Il file che contiene il bytecode è una traduzione delle istruzioni in un linguaggio che la macchina interpreterà



Per eseguire un programma si usa l'*interprete* JAVA, un programma eseguibile sul computer dell'utente.

- Per poter interagire con il *prompt* è necessario interagire con il sistema operativo, un'operazione di basso livello che richiede conoscenze specifiche.
- Queste operazioni, sono state già realizzate dagli autori del linguaggio, che hanno scritto delle classi apposite (ad esempio: `System`).
- Il bytecode di queste classi si strova all'interno delle librerie standard, che sono raccolte di classi.

Il processo di programmazione JAVA



Parlando nel linguaggio Java

Il linguaggio di programmazione Java, ha tutti gli aspetti di una lingua normale ad esempio come l'Italiano. Java ha una sua grammatica, nomi comunemente utilizzati e cose di questo genere.

I creatori di Java hanno pensato di dividerlo in due parti. Come l'italiano ha la sua grammatica e nomi comunemente utilizzati il linguaggio di programmazione Java ha le sue specifiche (la sua grammatica) e Application Programming Interface (parole comunemente utilizzate).

Tradizionalmente, il primo programma che si scrive quando si impara un linguaggio di programmazione ha il compito di visualizzare a schermo il messaggio:

Hello, World!

Aprite l'editor di testo, Geany, installato sul vostro computer, e salviamo il nostro primo documento HelloWorld.java.

HelloWorld.java

```
class HelloWorld{
    public static void main(Strings[] args){
        //Visualizza il messaggio sullo ↩
            schermo
        System.out.println("Hello, World");
    }
}
```

Attenzione

- maiuscole e minuscole sono considerate distinte
- il file deve chiamarsi HelloWorld.java

- A questo punto compiliamo il nostro primo programma:
  - Prompt dei comandi

- A questo punto compiliamo il nostro primo programma:
  - Prompt dei comandi
  - cd Con il percorso per arrivare alla cartella contente il file `HelloWorld.java`

- A questo punto compiliamo il nostro primo programma:
  - Prompt dei comandi
  - cd Con il percorso per arrivare alla cartella contente il file `HelloWorld.java`
  - javac HelloWorld.java

- A questo punto compiliamo il nostro primo programma:
  - Prompt dei comandi
  - cd Con il percorso per arrivare alla cartella contente il file `HelloWorld.java`
  - javac HelloWorld.java
  - se non ci sono errori il compilatore genera il file HelloWorld.java

- A questo punto compiliamo il nostro primo programma:
  - Prompt dei comandi
  - cd Con il percorso per arrivare alla cartella contente il file `HelloWorld.java`
  - javac HelloWorld.java
  - se non ci sono errori il compilatore genera il file HelloWorld.java
- Ora **eseguiamo** il nostro primo programma:
  - java HelloWorld

- A questo punto compiliamo il nostro primo programma:
  - Prompt dei comandi
  - cd Con il percorso per arrivare alla cartella contente il file `HelloWorld.java`
  - javac HelloWorld.java
  - se non ci sono errori il compilatore genera il file HelloWorld.java
- Ora **eseguiamo** il nostro primo programma:
  - java HelloWorld

Ottenuto qualcosa sullo schermo?

Analisi del progrmma Java

La prima riga

La prima riga:

```
public class HelloWorld
```

definisce una nuova **classe**.

- Come abbiamo definito prima le classi rappresentano un concetto fonamentale in Java.
- Per il momento, consideriamo gli oggetti come **elementi da manipolare in un programma Java**.

- **parola chiave** public indica che la classe `HelloWorld` può essere utilizzata da tutti
- Una *parola chiave* è una parola riservata e che non può essere usata per altri scopi.
- Ciascun file sorgente può contenere una sola classe pubblica il cui nome deve coincidere con il nome del file.

- Le nostre classi per ora conterranno solo dei metodi
- Un metodo serve a definire una sequenza di istruzioni che servono per descrivere come svolgere un determinato compito
- Un metodo deve inserito all'interno di una classe, quindi le classi rappresentano il meccanismo principale di organizzazione dei programmi.

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

main

con la riga di sopra viene definito il metodo main.

- Un'applicazione Java **deve** avere un metodo main.
- `static` significa il che il metodo main **esamina e non modifica gli oggetti della classe** `HelloWorld` **a cui appartiene**

Commenti

Nel programma sono presenti anche dei **commenti** che vengono ignorati dal compilatore, ma che rendono il programma molto più comprensibile.

- Un commento inizia con la doppia barra // e termina a fine della riga;
- All'interno del commento può essere scritta qualsiasi cosa;
- Se un commento si estende su più righe allora è più comodo farlo iniziare con /* e finire con */

```
//questo è un commento
//lungo, inutile .....
//... e anche scomodo
```

```
/* questo è un commento
lungo ma comunque inutile
*/
```

Gli enunciati del **corpo** di un metodo vengono eseguiti uno alla volta nella sequenza con cui sono scritti

- Ogni enunciato termina con il carattere ; (salvo piccole eccezioni)
- Il metodo main del nostro esempio ha un solo enunciato che termina con ;

La stringa Hello, World può essere stampata in diverse parti: su un file, sull'**output_standard** o in una finestra. In Java, l'output standard è rappresentato da un oggetto di nome out.

- come ogni metodo, anche gli oggetti devono essere inseriti all'interno delle classi:
  - out è inserito all'interno della classe `System` della libreria standard.

```
System.out.println("Hello, World");
```

Quando si ha un oggetto, bisogna specificare cosa si vuole fare con questo:

- in questo caso vogliamo **usare un metodo** dell'oggetto out, il metodo println che stampa una riga di testo.
- la coppia di parentesi tonde racchiude le informazioni necessarie per l'esecuzione del metodo (**parametri**)

Il punto

A volte il carattere **punto** significa *usa un oggetto di una classe*, altre volte *usa un metodo di un oggetto*

Sintassi

```
oggetto.nomeMetodo(parametri);
```

- Viene invocato il metodo `nomeMetodo` dell'`oggetto`, fornendo dei `parametri` se sono richiesti.
- Se non sono richiesti parametri le parentesi vanno messe ugualmente.
- se ci sono più parametri questi saranno separati da una **virgola**.

println

- Il metodo println può stampare anche numeri, senza indicare gli apici.
- C'è anche il metodo print, che funziona come println ma **non va a capo al termine della stampa**.