

# Il mio primo programma java

## Java

Java è il linguaggio di programmazione che utilizzeremo durante questo corso. Come già detto è un linguaggio di tipo a codice intermedio ed è modellato per la programmazione orientata agli oggetti (OOP Object Oriented Programming).

Esistono più di un modo operativo per scrivere programmi sorgenti Java, tradurli ed eseguirli. Inizialmente utilizziamo un ambiente che predispone in modo unitario tutto quanto server per le diverse fasi e viene pertanto chiamato ambiente di sviluppo (in inglese IDE: Integrated Development Environment).

Questo ambiente si chiama NetBeans ed è realizzato dalla società informatica SUN che è anche la società che ha inventato Java. Il programma è scaricabile gratuitamente all'indirizzo web <http://www.netbeans.org/> .

## Scriviamo il primo programma

Ogni scritto dal più formale al più divulgativo che presenti come realizzare il primo programma in una qualunque ambiente propone di realizzare un saluto "Ciao mondo!" e così sarà anche nel nostro caso...

Esplicito in un modo un po' più formale l'obiettivo: intendiamo progettare e realizzare un programma che sia in grado di visualizzare la frase "Ciao mondo!". I passaggi sono :

- scrivere secondo le regole di java il programma, correggendo eventuali errori fino ad ottenere un programma che può essere eseguito
- richiedere l'esecuzione del programma ottenendo l'effetto desiderato

Per arrivarci dobbiamo conoscere alcune caratteristiche di Java.

## La dichiarazione di classe

Un primo punto è la regola su come descrivere una classe:

```
class <nomeclasse> {  
    <descrizione della classe>  
}
```

in cui:

- class è un **parola chiave (keyword)** che va sempre scritta esattamente così, serve per indicare che ciò che segue è la dichiarazione di una classe, cioè la descrizione della classe che si sta realizzando
- <nomeclasse> rappresenta genericamente una parola che scegliamo noi arbitrariamente per indicare (identificare) la classe che stiamo dichiarando. Viene chiamato identificatore della classe

Quindi potrà essere:

```
class Ciao {  
    <descrizione della classe>  
}
```

Va precisato che il linguaggio java prevede la netta distinzione tra caratteri minuscoli e maiuscoli.

Quindi se si scrive Class, o CLASS o altre combinazioni minuscolo/maiuscolo la parola non viene riconosciuta come parola chiave.

Per quanto riguarda invece l'identificatore di classe seguiremo la convenzione che viene tipicamente adottata: l'identificatore di classe ha iniziale maiuscola e resto minuscolo se costituito da una unica parola. Può essere una parola composta da più termini ottenuta scrivendo di seguito, tutto attaccato, le parole componenti, ciascuna con l'iniziale maiuscola. Esempio potremmo chiamare la nostra classe CiaoMondo.

Dopo la parola chiave 'class' e l'identificatore 'Ciao' segue una parentesi graffa aperta '{' a cui corrisponde più avanti la graffa chiusa '}'. Anche questi sono elementi obbligatori che fanno parte della corretta sintassi di scrittura di una classe.

Tra le due graffe la *<descrizione della classe>* rappresenta genericamente quanto effettivamente descrive ciò che la classe contiene (attributi) e/o deve poter fare (metodi).

## **Metodi di una classe**

I metodi di una classe descrivono ciò che desideriamo che la classe sia in grado di fare, rappresentano dunque le capacità di azione della classe.

In generale siamo liberi di progettare quanti metodi riteniamo siano utili ai nostri fini ma nel nostro caso stiamo progettando una classe che ha un compito specifico: rispondere al comando di avvio programma.

Quindi posticipo la descrizione di come comportarsi in modo generico sui metodi e partiamo dal particolare metodo di questa classe.

## **Il metodo main**

Riformulo la proprietà che ci serve.

Ci serve una classe che è in grado di rispondere allo stimolo di avvio del programma (ad esempio il doppio click di un'icona, o un comando scritto in una finestra a comandi).

Per fare in modo che una classe abbia questa caratteristica deve avere un metodo che sia così scritto:

```
public static void main (String[] args) {  
    <contenuto del metodo>  
}
```

La parte *<contenuto del metodo>* rappresenta genericamente ciò che lo specifico nostro programma dovrà realizzare, le altre parti invece sono 'fisse' cioè vanno scritte esattamente così. Ci chiariremo strada facendo l'esatto significato di ognuno dei termini utilizzati. Per ora vorrei soffermarmi sul perché di questo vincolo su questo metodo.

Quando facciamo eseguire un programma (ad esempio con doppio click su un'icona) c'è una sorta di "passaggio di consegne" in cui il computer prende in carico il programma e incomincia ad eseguirlo. Ma da che punto? Ci vuole un punto "inizia da qui" che sia immediatamente riconoscibile. Bene, nei programmi java "l'inizia da qui" è costituito dal metodo con le regole descritte in precedenza. Per essere completi va anche aggiunto che tale metodo deve essere all'interno di una classe che ha come identificatore lo stesso

nome del programma.

Riassumendo quanto visto fin qui, la classe è così definita:

```
class Ciao {  
    public static void main (String[] args) {  
        <contenuto del metodo>  
    }  
}
```

dobbiamo ancora definire cosa mettere nel contenuto del metodo ma intanto salviamo il file con il nome Ciao.java in modo che il programma che otterremo abbia lo stesso nome della classe.

Come descrivere il "contenuto del metodo". In sostanza vogliamo che appaia la scritta 'Ciao mondo!', quindi ci serve un 'qualcosa che sa' come scrivere una frase.

Per questo abbiamo a disposizione in java alcune classi ed oggetti "predefiniti" che sono cioè pronti all'uso. Si tratta di ricordarsi che esistono, cosa fanno e come vanno utilizzati... :-)

Nel nostro caso ci viene in aiuto "System.out.println" che è un metodo che abbiamo a disposizione per poter far visualizzare una frase in una finestra di comandi. La frase va specificata come parametro passato al metodo in questo modo: System.out.println("Ciao mondo!"), e dunque ora abbiamo un aspetto completo per la nostra classe:

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

La riga che contiene la 'chiamata' del metodo println termina con un ";". Questa regola è generale, ogni istruzione java termina con ";".

Preciso che siccome i passaggi sono già stati tanti ne ho qui saltato uno piuttosto importante che vedremo nel prossimo paragrafo e di cui faremo uso comunque da lì in poi.

cc

---

cc Quest'opera è stata rilasciata con licenza Creative Commons Attribuzione - Non commerciale - Condividi allo stesso modo 3.0 Italia. Per leggere una copia della licenza visita il sito web <http://creativecommons.org/licenses/by-nc-sa/3.0/it/> o spedisci una lettera a Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.  
Giovanni Ragno – ITIS Belluzzi Bologna 2012