



Fondamenti di Programmazione

Lezione 1

Danilo Amendola

October 11, 2017

University of Trieste

- Homepage del corso: units.it
- email: danilo.amendola@uniroma1.it (usate [fondamenti] all'inizio dell'oggetto della mail + scegliete oggetto mail in modo appropriato)
- ricevimento: su appuntamento (sabato compreso)

Organizzazione del corso

Il corso inizia dal... Parte teorica sulle Reti Logiche col Prof. Fabris;
Parte teorica su fondamenti di programmazione col Prof. Omodeo; Parte
pratica, su Java e C, con me Danilo Amendola.

Faremo 20 lezioni da 2 ore:

- il ??? dalle ore alle ore
- il sabato dalle ore ??? alle ore ???

Per le date fare riferimento al sito web che verrà aggiornato periodicamente.

Obiettivi:

- Le basi della programmazione
- La programmazione strutturata in Java e C
- Strutturare il codice: metodi e classi
- Implementazione di algoritmi in Java
- Introduzione alla programmazione orientata ad oggetti

Cosa serve per iniziare

Cosa serve per iniziare a lavorare con Java:

- Installate un Editor di testo (notepad, notepad++, Sublime, etc);
- Installate la Java SDK (Software Developement Kit) [link](#);
- la command line di windows (cmd.exe) o la shell di Linux/OSX;
- Potete seguire una guida per l'installazione di Java: [guida all'installazione](#);

IDE (Integrated Development Environment)

Un IDE è un ambiente di sviluppo integrato che permette di automatizzare e semplificare la scrittura di codice, la compilazione, l'esecuzione ed il debugging.

Concetti chiave:

- **Compilazione:** è l'operazione che a partire dal codice sorgente genera il programma eseguibile da una macchina (o da VM);
- **Esecuzione:** è l'operazione di avvio del programma compilato;
- **Debugging:** è una modalità di esecuzione del programma che ci permette di analizzarne il comportamento delle variabili per identificare eventuali bug.

IDE (Integrated Development Environment)

Esistono molti IDE:

Concetti chiave, tra i più importanti per Java ci sono:

- Eclipse: eclipse.org, il più consigliato [[download](#)]
- Netbeans: netbeans.org
- IntelliJ IDEA: jetbrains.com
- BluJ: bluj.org
- DrJava drjava.org
- etc.

Esistono molti IDE:

Concetti chiave, tra i più importanti per Java ci sono:

- Eclipse: eclipse.org, il più consigliato [[download](#)]
- Netbeans: netbeans.org
- IntelliJ IDEA: jetbrains.com
- BluJ: bluj.org
- DrJava drjava.org
- etc.

Il primo programma di sempre

Il primo programma che si scrive è storicamente HelloWorld.java:

```
//Hello World in Java  
class HelloWorld {  
    static public void main( String args[] ) {  
        System.out.println( "Hello_World!" );  
    }  
}
```

Curiosità: su sito [helloworldcollection](#) trovate il programma HelloWorld scritto in oltre 556 (ad oggi) linguaggi diversi.

Il proprogramma: prime keyword

Termine	Spiegazione
<code>public</code>	Modificatore del metodo. I modificatori sono utilizzati in Java come nel linguaggio umano sono utilizzati gli aggettivi. Se si antepone un modificatore alla dichiarazione di un elemento Java (un metodo, una variabile, una classe, etc.), questo cambierà in qualche modo (a seconda del significato del modificatore) le sue proprietà. In questo caso trattasi di uno specificatore d'accesso che rende di fatto il metodo accessibile anche al di fuori della classe in cui è stato definito.
<code>static</code>	Altro modificatore del metodo. La definizione di <code>static</code> è abbastanza complessa. Per ora il lettore si accontenti di sapere che è essenziale per la definizione del metodo <code>main()</code> .
<code>void</code>	È il tipo di ritorno del metodo. Significa “vuoto” e quindi questo metodo non restituisce nessun tipo di valore. Il metodo <code>main()</code> non deve mai avere un tipo di ritorno diverso da <code>void</code> .
<code>main</code>	Trattasi del nome del metodo (detto anche “identificatore” del metodo).
<code>(String args[])</code>	Alla destra dell'identificatore di un metodo, si definisce sempre una coppia di parentesi tonde che racchiude opzionalmente una lista di parametri (detti anche argomenti del metodo). Il metodo <code>main()</code> , in ogni caso, vuole sempre come parametro un array di stringhe (agli array troveremo dedicata nel terzo modulo un'intera unità didattica). Si noti che <code>args</code> è l'identificatore (nome) dell'array, ed è l'unica parola che può variare nella definizione del metodo <code>main()</code> , anche se per convenzione si utilizza sempre <code>args</code> .

Fonte: Manuale di Java 8 di CDS Cesari

Il proprogramma: prime keyword

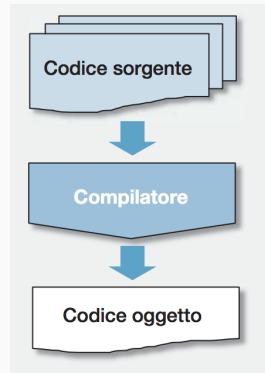
Dopo aver installato la Virtual Machine di Java, la Java Runtime Machine possiamo iniziare. [Consiglio: spulcia le cartelle di installazione della VM (.../Java/*)]

Dal codice all'esecuzione:

Compilare col comando: `javac`;

Eeguire col comando: `java`;

Curiosità: per esercitarvi senza avere la VM sul computer esistono degli ambienti online come: repl.it



Fonte:

Commenti ed Indentazione

I commenti ci permettono di inserire del testo che verrà ignorato dal compilatore. Serve ad aumentare la leggibilità del codice.

Alcune modalità di commento:

- Commentare un'intera riga → `//`
- Commentare una porzione di codice → `/ * ... */`

L'**indentazione** è una buona pratica per avere del codice leggibile.

```
class HelloWorld {
    static public void main(String
a) {
        System.out.println("Hi!");
    }
}
```

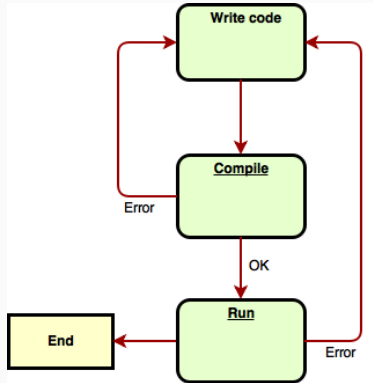
```
class HelloWorld {
    static public void main(String a) {
        System.out.println("Hi!");
    }
}
```

Compilazione ed Esecuzione

Un programma viene prima scritto, poi compilato ed infine eseguito (ammesso che tutto vada bene...):

I comandi necessari:

- per compilare: `javac`
- per eseguire: `java`



Come si rappresenta l'informazione???

-

I tipi primitivi in Java i

Il **tipo** di un dato rappresenta l'insieme delle caratteristiche che il valore assunto da una variabile dovrà soddisfare.

I tipi primitivi in Java:
Qualche esempio:

- `boolean b = true;`
- `char c = 'k';`
- `byte x = -10;`
- `short t = -10000;`
- `int j = 7;`
- `long l = 10000;`
- `float f = 0.003;`
- `double d = -0.000004432;`

I tipi primitivi in Java ii

Il **tipo** di un dato rappresenta l'insieme delle caratteristiche che il valore assunto da una variabile dovrà soddisfare.

I tipi primitivi in Java:

Tipo	Q.ta Mem.	Range	Descrizione
boolean	→ 1 bit	[true false]	Vero/falso (true o false)
char	→ 16 bit	['\u0000', '\uffff'] o [0, 65535]	Caratteri charset Unicode
byte	→ 8 bit	[-128,127]	
short	→ 16 bit	[-32.768, 32.767]	Numeri interi
int	→ 32 bit	$[-2^{31}, 2^{31} - 1]$	Numeri interi
long	→ 64 bit	$[-2^{63}, 2^{63} - 1]$	Numeri interi
float	→ 32 bit	spec. IEEE 754 8 bit exp.: [-126, 127]	Num. virgola mobile (s.p.)
double	→ 64 bit	spec. IEEE 754	“ doppia precisione

I primi esempi

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

```
public class Esercizio1 {
    /*
     * Scrivere un programma che dati base ed altezza di un
     * triangolo, ne calcoli l'area.
     */
    public static void main(String[] args) {
        double base = 3.5;
        double altezza = 4.3;
        double area = base * altezza / 2;
        System.out.println("Area: " + area);
    }
}
```