PROGETTO LAUREE SCIENTIFICHE

"Write once, write everywhere " è il motto dei progettisti del linguaggio java. Infatti questo linguaggio è il risultato dell' esigenza di programmare per quello che si può chiamare mondo "on line". Il Web non è altro che un universo popolato da vari tipi di computer, sistemi operativi e CPU, proprio per questo motivo i programmi realizzati con il Java sono indipendenti dalla piattaforma e consentono di utilizzare lo stesso codice in Windows, Solaris, Linux, Macintosh e così via. Una volta scritto un programma vi possono essere vari fattori che ne determinano avarie, tra cui gli aggiornamenti dei sistemi operativi e dei processi o cambiamenti nelle risorse del nucleo del sistema, tutte risolvibili attraverso il compilatore. Infatti quest'ultimo produce codice a byte che non necessita di una specifica architettura ma che è progettato per essere interpretato in qualsiasi computer e traducibile "al volo" nel codice macchina nativo. Il bytecode è un set di istruzioni che può essere eseguito dal sistema run-time di Java la cosiddetta Java Virtual Machine (JVM).

Come abbiamo detto precedentemente Java deve il suo successo ad Internet, ma allo stesso modo Java ha avuto un profondo effetto su Internet. Nella rete esistono due categorie di oggetti: passivi ed attivi, dinamici. Ad esempio quando si legge la posta elettronica si vedono dati passivi, se invece si apre un'applet si ha un oggetto attivo. I programmi Java che funzionano sulle pagine Web si chiamano *applet*. Un'applet è un'applicazione che viene trasmessa ad un browser Web abilitato all'uso di Java. Quando l'utente scarica un'applet, il funzionamento è molto simile all'inserimento

di un'immagine in una pagina Web. L'applet diventa parte della pagina e il testo scorre intorno allo spazio utilizzato per l'applet. Ma l'immagine è *viva*: reagisce ai comandi dell'utente, modifica il suo aspetto e invia dati tra il computer che visualizza l'applet e il computer che la fornisce. Le prime applet sono state utilizzate anche per l'animazione (si pensi alle sfere che ruotano, ai personaggi danzanti dei cartoni animati, al testo mobile e così via). Spesso nella creazione di pagine Web viene adoperato Java. Questo è motivo di confusione, in quanto si pensa che Java sia una parte di *Hypertext Markup Language* (HTML). In realtà HTML è solo un mezzo visto che descrive la struttura di una pagina Web. Infatti anche se HTML permette all'utente di leggere un documento in modo dinamico non un linguaggio di programmazione. Il solo legame tra Java e HTML è la presenza del *tag* applet che permette di eseguire un applet Java.

Perch impiegare questo linguaggio come strumento per la didattica? Attualmente vi sono molti linguaggi e softwares specifici per la matematica (basti pensare a *Mathematica, Cabri-Geomètre, Derive* e così via). Tutti questi programmi sono molto validi e indubbiamente questa non è una gara per stabilire quale sia il migliore. Java usato come strumento di didattica della matematica ha molti aspetti positivi: Per utilizzare un qualsiasi software bisogna conoscere i comandi correlati e farli imparare anche ai propri alunni ma le applet di Java sono speciali applicazioni in quanto non hanno comandi specifici ma interagiscono con l'utente in modo pratico e veloce. Infatti, in genere, per rendere attiva un'applet bisogna solo cliccare con il mouse su un bottone, oppure usare la tastiera. Tutto ciò non richiede una preparazione di base, ma risulta quasi istintivo per qualsiasi persona che abbia un po' di

dimestichezza con il computer. Un altro punto a favore delle applet Java è la possibilità di usare la rete. Infatti esse possono essere scaricate e utilizzate anche da casa, senza dover installare alcun software aggiuntivo. Quindi, oltre all'attività legata ad un laboratorio didattico, un insegnante pu rendere interattivi anche i noiosissimi compiti a casa.

Come abbiamo detto prima, impiegare applet Java non richiede l'installazione di software aggiuntivi. Questo riduce i costi legati alle amministrazioni scolastiche. Perciò, se è già presente un laboratorio con un numero congruo di computer, non ci sono impedimenti all'uso delle applet Java. Inoltre le applet funzionano indipendentemente dalla rete. Infatti esse si presentano come pagine HTML quindi una volta scaricate, insieme ai relativi file con estensione ".class", esse rimangono attive poichè Java è un linguaggio di programmazione indipendente dalla piattaforma.

Conoscere un linguaggio di programmazione è molto interessante e produttivo, ma non indispensabile. Infatti in Internet si pu trovare una specie di biblioteca virtuale. Basta servirsi di un qualsiasi motore di ricerca per trovare applet molto interessanti e funzionali. Il vantaggio di produrre le applet da soli è quello di poterle costruire secondo le proprie esigenze ed aspettative. Sicuramente un vestito confezionato su misura calza meglio di un qualsiasi altro vestito. Si possono scrivere sorgenti che simulano eventi, svolgono calcoli, disegnano grafici di funzioni e tutto quello che ci può venire in mente. Java è sicuramente un linguaggio per professionisti che, con un po' di buona volontà, è relativamente facile da imparare.

0.1 Primi passi in Java

Vediamo in pratica cosa si deve fare per utilizzare il Java.
Prima di tutto, come per ogni linguaggio di programmazione, bisogna scrivere il *sorgente*. Questo termine, in senso figurale, significa origine, e in effetti è proprio l'origine del programma. Nel codice sorgente si scrive tutto ciò che poi il compilatore deve tradurre in *codice oggetto*. Materialmente il codice sorgente viene scritto in un *editor*. L'editor varia a seconda del sistema operativo. In *Linux* la scelta migliore è *Emacs*. In *Windows* si può scegliere tra *WordPad, NotePade* e *Text-Pad*. Per le altre piattaforme *JEdit* è un'eccellente alternativa. Tutti questi editor sono già presenti su i nostri computer basta solo cercarli tra i programmi installati.
Una volta scritto il sorgente, esso va *compilato*. Per fare ciò è necessario un compilatore. Quest'ultimo è facilmente reperibile in rete tramite la stessa *Sun Microsystems*. Infatti, all'indirizzo **http://java.sun.com/j2se** si possono scaricare le versioni più complete di *Java 2 Standard Edition* per ogni piattaforma: *Solaris, Linux, Windows* ecc. Una volta installato il pacchetto *Java 2 Standard Edition* abbiamo a nostra disposizione, non solo un compilatore ma vari programmi. Tra essi ne troviamo uno per eseguire, uno per vedere le applet in anteprima e così via.
Riassumendo, i punti fondamentali sono:

- *Creare un sorgente*: cioè scrivere un file di testo nel linguaggio Java con un editor di testo. Salvarlo come documento di testo e usare l'estensione *.java*;

- *Compilare*: in altre parole, tramite il compilatore si traduce il sorgente

da file di testo, in file di bytecode. In questo modo il testo viene trasformato in istruzioni che la Java Virtual Machine pu capire;

- *Eseguire*: L'interprete Java prende il file di bytecode ed esegue le istruzioni traducendole in modo tale che il nostro computer può comprenderle;

Per compilare ed eseguire un programma si può scegliere di utilizzare la riga di comando o un altro programma, un ambiente di sviluppo integrato o un editor di testo. La nostra scelta è orientata verso la riga di comando. Per fare ciò bisogna aprire una finestra di Shell o di terminale ed immettere i seguenti comandi:

javac nome del file.java

java nome del file

e premere *Invio*.

Il programma *javac* è il compilatore e trasforma un file *.java* in uno *.class*. Il programma *java* è l'interprete, cioè quello che traduce i bytecode in istruzioni per il computer.
Quando il programma viene compilato pu accadere che vi siano degli errori e quindi il file *.class* non viene creato. Questo sistema è una ulteriore garanzia ad evitare i tanto temuti bachi.

0.2 Un semplice programma Java

Al fine di comprendere meglio tutto ciò che si è detto in precedenza faremo un esempio classico, che è presente ovunque si parli di Java. Reperibile anche sul tutorial presente nel sito della *Sun Microsystems*: **http://java.sun.com/docs/books/tutorial/index.html**. Esso non è un applet ma una semplice applicazione che fa apparire un messaggio nella finestra di console. Il suo sorgente è:

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

Esaminiamo il codice sorgente. La parola *public* un *modificatore di accesso*. Qualsiasi cosa dichiarata come *public* è visibile ai membri della classe *(class)*. In altri termini un modificatore di accesso controlla quali altre parti del programma, al di fuori di quelle parti incluse in *class*, possono utilizzare il codice. Oltre *public* ci sono altri modificatori di accesso come *private, protected*. La parola *class* indica la classe, cioè un contenitore che definisce il comportamento di un'applicazione, fondamentale per la programmazione orientata agli oggetti. Dopo *class* c'è il suo nome. Per i nomi delle classi si possono utilizzare sia lettere che numeri. L'importante è che le iniziali siano lettere maiuscole. Non ci sono restrizioni sulla lunghezza. Se un nome è costituito da più parole per ogni iniziale si deve adoperare una lettere

maiuscola. Quando salviamo il codice sorgente dobbiamo utilizzare lo stesso nome della classe pubblica, con l'aggiunta dell'estensione *.java* e non *.txt*. In questo caso si salverà il file sorgente come *HelloWorldApp.java*, specificando che si tratta di un file di testo. Il file va salvato nella stessa cartella in cui si trova il pacchetto della *Java 2 Standard Edition*. In generale nel momento dell'istallazione si crea una cartella dal nome "Java".

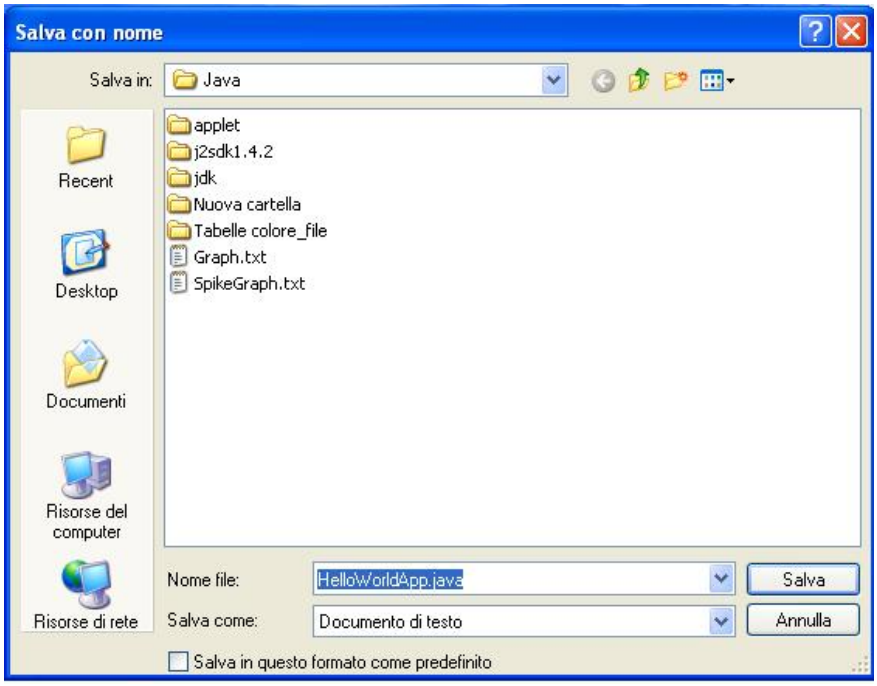

Figure 1: Salvare un file .java

Bisogna fare molta attenzione alle lettere maiuscole e minuscole per non causare errori nella esecuzione del codice sorgente. Se non ci sono stati errori di digitazione ed il nome corretto, dopo aver cambiato *directory* e indicato il percorso da seguire, usando il compilatore nella finestra di Shell potremo

vedere:

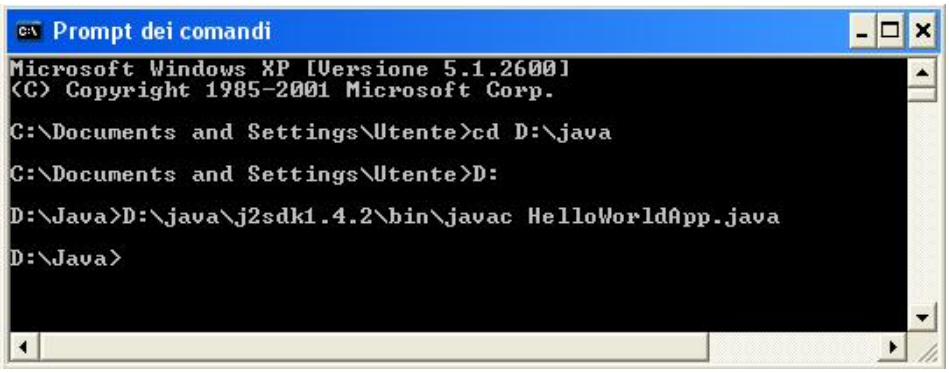

Figure 2: Compilazione file *HelloWorldApp.java*

La scritta D:\java> indica che il programma stato compilato correttamente e quindi è stato creato il file *HelloWorldApp.class*. A questo punto bisogna eseguire il file e quindi immettere, sempre dopo aver dichiarato il percorso, il comando *java HelloWorldApp*, così premendo *Invio* potremmo vedere la stringa "Hello World!".

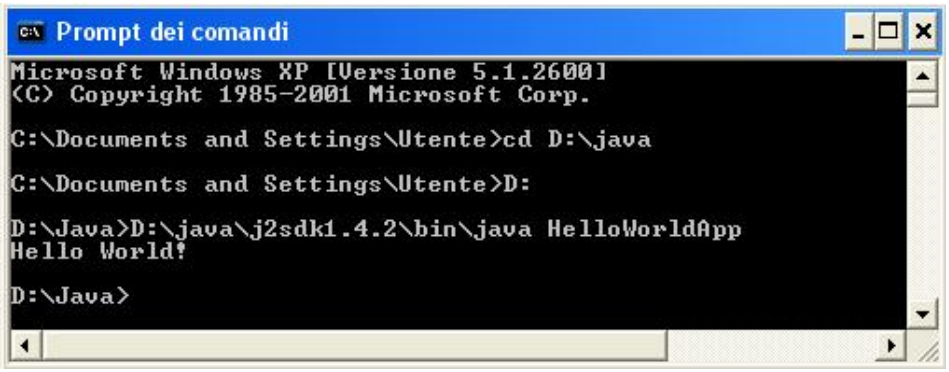

Figure 3: Stringa finale

Nel codice è anche presente il metodo *main*; quest'ultimo fa si che il codice venga eseguito. Ci sono anche altri metodi e inoltre pu accadere che in esso non ci siano istruzioni, in questo caso si usano le parentesi vuote. Le parentesi graffe in Java vengono utilizzate per delineare le parti del programma, ciò che viene chiamato "blocco". Gli spazi vuoti presenti nel codice vengono ignorati, quindi si pu scegliere di adoperarli per separare meglio i blocchi l'uno dall'altro. Le parole *static void* possono essere considerate una parte del programma, indispensabile per la sua compilazione. L'importante è che il programma abbia un metodo *main*, che deve presentare la seguente struttura:

```
        public class NomeClasse {
    public static void main(String[] args) {
        istruzioni programma;
    }
}
```

Come in quasi tutti i programmi le istruzioni Java sono considerate come delle frasi del linguaggio naturale. Ogni istruzione deve essere seguita dal punto e virgola. Java, come C++, adopera le doppie virgolette per delimitare le stringhe.

0.3 Le applet

Finora abbiamo mostrato come compilare, editare ed eseguire un programma Java, e come fondamentalmente opera Java con le classi. Nell'esempio *HelloWorldApp* abbiamo messo in evidenza il fatto che esso non è un'applet.

Infatti un'applet è un particolare programma Java che un browser può scaricare da Internet ed eseguire. Prima di Java per scrivere una pagina Web si utilizzava HTML *(Hyper Text Markup Language)*. Esso però non è un linguaggio, ma solo uno strumento per strutturare il *layout* di una pagina Web. Per fare ciò si usano i *tag*. Ad esempio si può indicare il titolo di una pagina adoperando il *tag* <Title> seguito dal titolo che si vuole dare alla pagina. Per indicare la fine del titolo si utilizza il *tag* </Title> (la barra indica, in generale, la fine dell'elemento).

Per impiegare un'applet in una pagina Web bisogna indicare al browser quali applet scaricare e la posizione che devono occupare nella pagina Web. Quindi abbiamo bisogno di un *tag* che indichi al browser quali file *.class* prendere, dove metterli e quali dimensioni devono assumere. Il browser reperisce i file da internet o da una directory nel computer dell'utente, dopodichè esegue automaticamente gli applet tramite un *Java Runtime Environment* esterno o la Java *Virtual Machine* incorporata. Quando inseriamo un'applet nella pagina HTML, abbiamo anche la possibilità di inserire tutti gli altri elementi HTML in uso in una pagina Web come i font multipli, elenchi puntati, immagini grafiche, collegamenti e così via. Bisogna ricordare che Java e HTML sono completamente diversi e che Java è uno strumento per dare vita alle pagine HTML.

All'inizio le applet dovevano utilizzare il browser Sun HotJava, perciò non ebbero grande successo. Infatti pochi erano disposti a cambiare browser. Il loro successo è stato decretato quando *Netscape* ha inserito una Java Virtual Machine sul suo browser Navigator. In seguito anche *Microsoft Internet Explorer* seguì l'esempio di Netscape. Ma questo non ha risolto il problema

in quanto entrambi i browser non tenevano il passo con le versioni più aggiornate di Java. Così la Sun ha ovviato al problema creando "Java Plug-In", uno strumento che permette ad entrambi i browser di eseguire applet Java utilizzando un ambiente *run-time* esterno.
Come abbiamo detto prima il precedente esempio non era un'applet, ma può diventarlo apportando semplici modifiche al sorgente in questo modo:

```
        import java.applet.*;
        import java.awt.*;
        public class HelloWorld  extends Applet {
   public void paint(Graphics g)
   {
     g.drawString("Hello world!", 50, 25);
   }
}
```

facile convertire un'applicazione in un'applet che pu essere incorporata in una pagina Web. Essenzialmente quasi tutto il codice rimane invariato. In questo caso bisogna creare una sottoclasse della classe Applet e renderla pubblica ed eliminare il metodo *main*. In questo sorgente troviamo nuove scritture: *import java.applet.*; e import java.awt.*;* ed inoltre l'uso di *Graphics*. La parola *import* è una scorciatoia per fare riferimento alle classi di package. Quest'ultimo non è altro che una raccolta di classi. La libreria Java standard è formata da numerosi package tra cui *java.lang, java.util,* e così via. I package vengono utilizzati per garantire l'unicità dei nomi delle classi. Se non usassimo *import* dovremmo aggiungere il nome completo del package prima di ogni nome di classe. Le istruzioni *import* vengono collocate all'inizio del

codice sorgente.

Graphics rappresenta il contesto grafico in quanto la stringa viene considerata come un elemento grafico.

Anche in questo caso il sorgente va salvato come file di testo, ma con estensione *.java* con lo stesso nome della classe pubblica *HelloWorld.java*. Dopo averlo compilato non c'è bisogno di eseguirlo poichè questo programma è un'applet. Abbiamo però la possibilità di vedere come sarà l'applet in anteprima, con il visualizzatore di applet *appletviewer*. Per fare ciò dobbiamo inserire l'applet in una pagina HTML. Ecco cosa si vedrà con appletviewer:

Figure 4: HelloWorld.java

Per inserire un'applet in una pagina HTML abbiamo bisogno di *tag* specifici. Il *tag* per indicare al browser quali applet scaricare e la posizione che devono occupare nella pagina Web è:

```
<APPLET CODE="HelloWorld.class" WIDTH=200 HEIGHT=100>
```

In questo modo si dice al browser di scaricare il file *HelloWorld.class* e di inserirlo nella pagina Web, e si indicano le dimensioni dell'applet (larghezza

200 e altezza 100). Se si desidera si può anche dare un titolo all'applet:

```
<TITLE>A simple applet</TITLE>
```

Ecco come scrivere il sorgente per questa pagina:

```
<HTML>
<HEAD>
<TITLE>A simple applet</TITLE>
</HEAD>
<APPLET CODE="HelloWorld.class" WIDTH=200 HEIGHT=100>
</APPLET>
</HTML>
```

Il sorgente va sempre scritto in un editor e poi salvato con estensione *.html*, ma non va ne compilato ne eseguito, proprio perchè HTML non è un linguaggio.