

Capitolo 1

Introduzione

Cay S. Horstmann

Concetti di informatica e fondamenti di Java

Obiettivi del capitolo

- Capire il significato dell'attività di programmazione
- Imparare a riconoscere le componenti più importanti dell'architettura dei computer
- Comprendere la distinzione fra linguaggi macchina e linguaggi di programmazione di alto livello
- Prendere confidenza con l'ambiente di programmazione e con il compilatore
- Compilare ed eseguire il primo programma Java
- Riconoscere gli errori logici e di sintassi

Che cos'è la programmazione?

- Il computer deve essere programmato per svolgere compiti specifici
- Differenti compiti = differenti programmi
- Programma
 - esegue una sequenza di istruzioni elementari in sequenza
 - contiene le sequenze di istruzioni necessarie a eseguire tutti i suoi compiti.
- Programmi sofisticati richiede una squadra di programmatori altamente specializzati e di altre persone con elevate competenze.

L'anatomia di un computer

- Unità centrale di elaborazione
 - Chip (circuito integrato)
 - Transistor
- Memoria
 - Memoria primaria: Random-access memory (RAM)
 - Memoria secondaria: es. hard disk (disco rigido)
 - Dispositivi di memorizzazione rimovibili
es. dischi floppy, nastri, CD

L'anatomia di un computer

- Dispositivi periferici
- Esegue istruzioni molto semplici
- Esegue istruzioni molto rapidamente
- Dispositivo general purpose

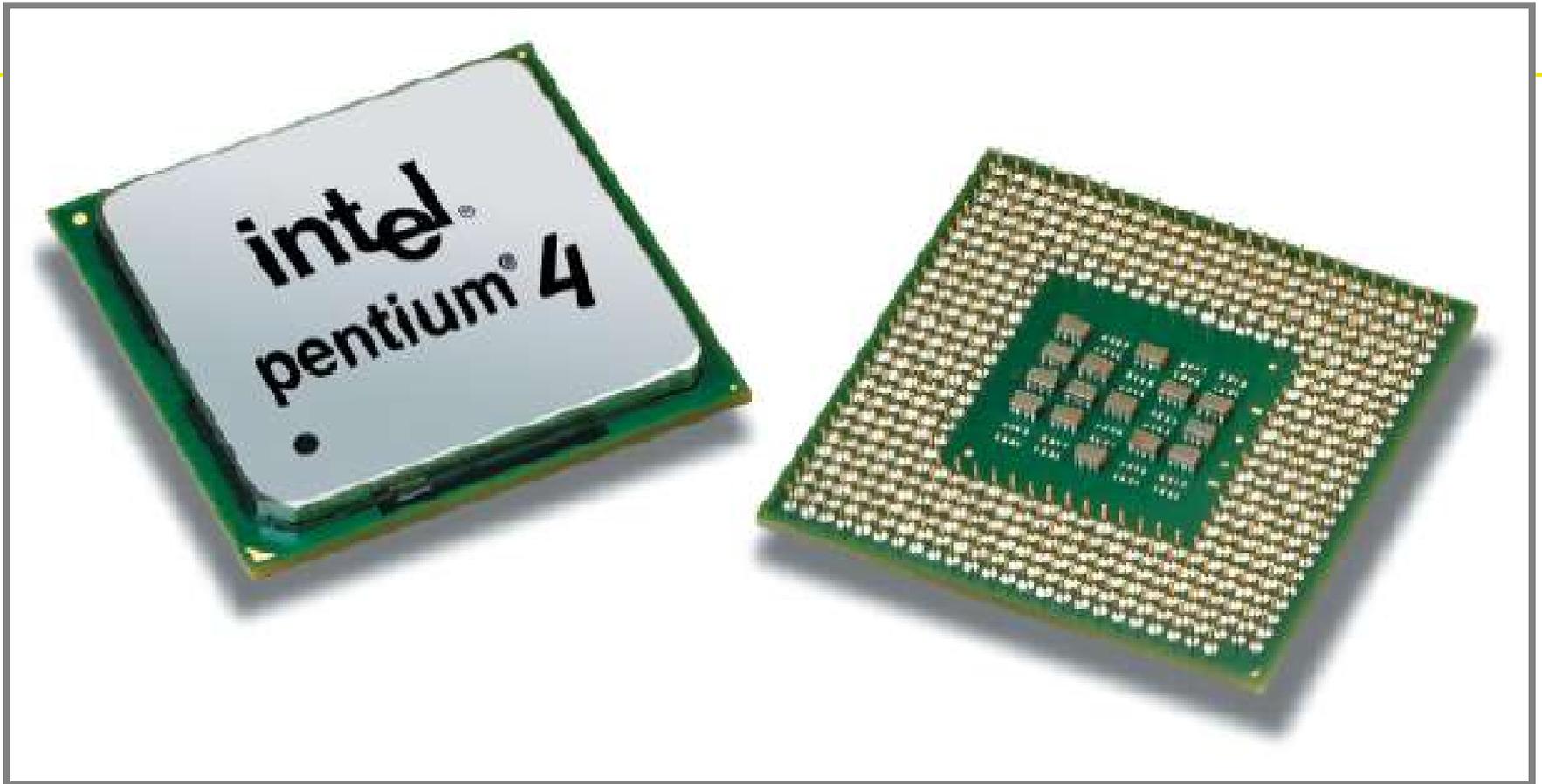


Figura 1: Unità di elaborazione centrale

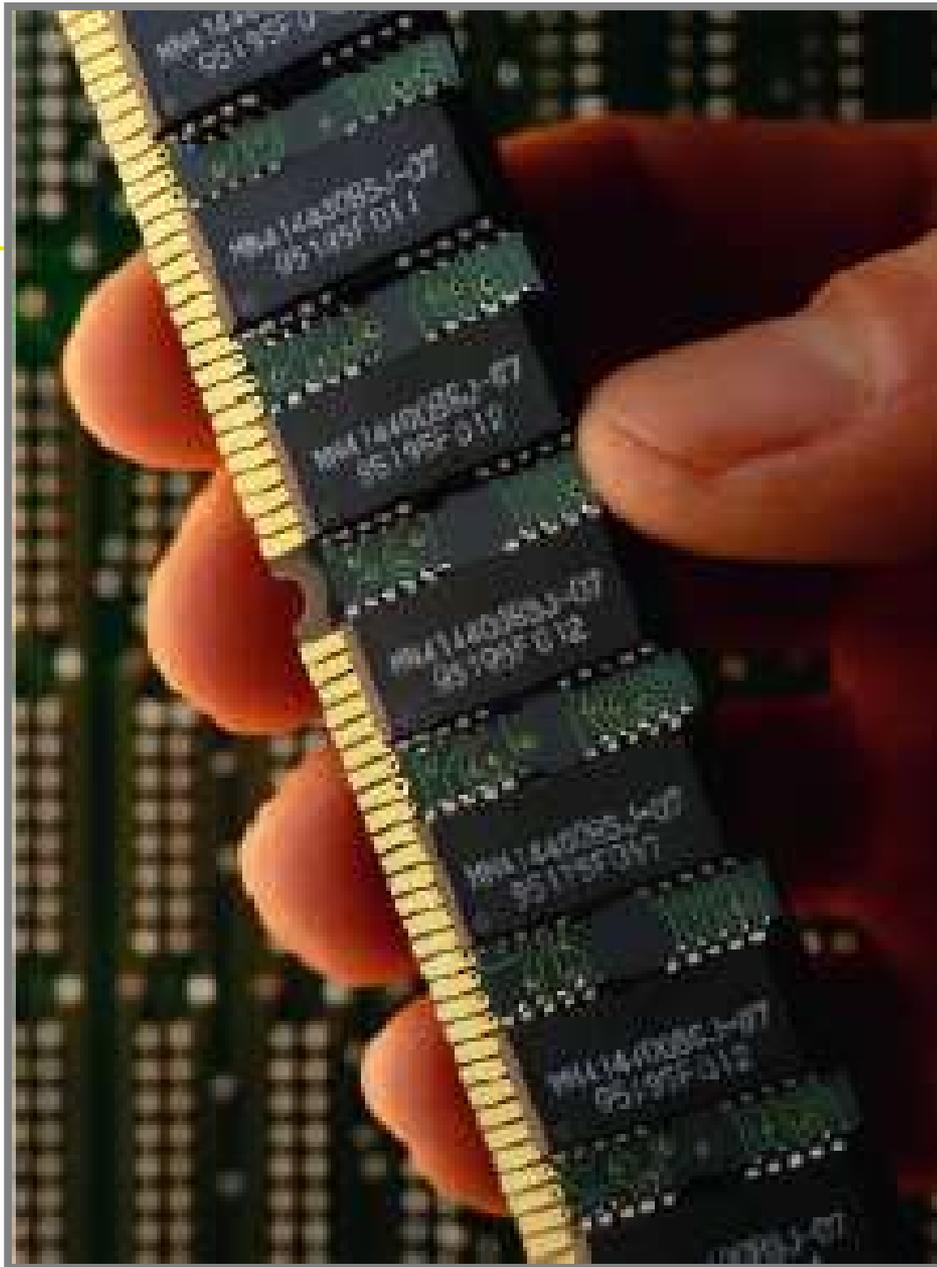


Figura 2
Un modulo con chip
di memoria



Figura 3 Un disco rigido
©Apogeo 2007

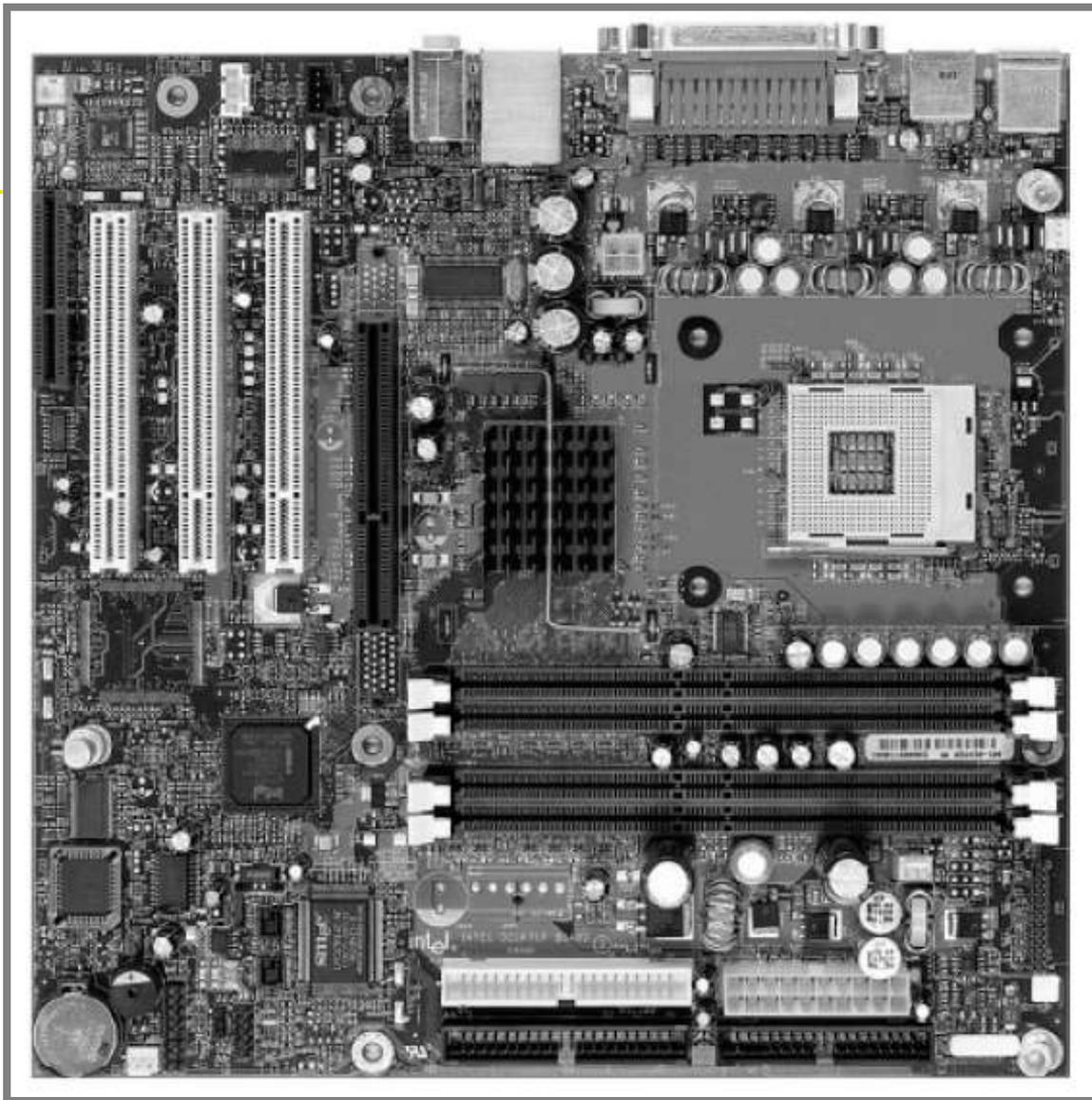


Figura 4 Una scheda principale (motherboard)

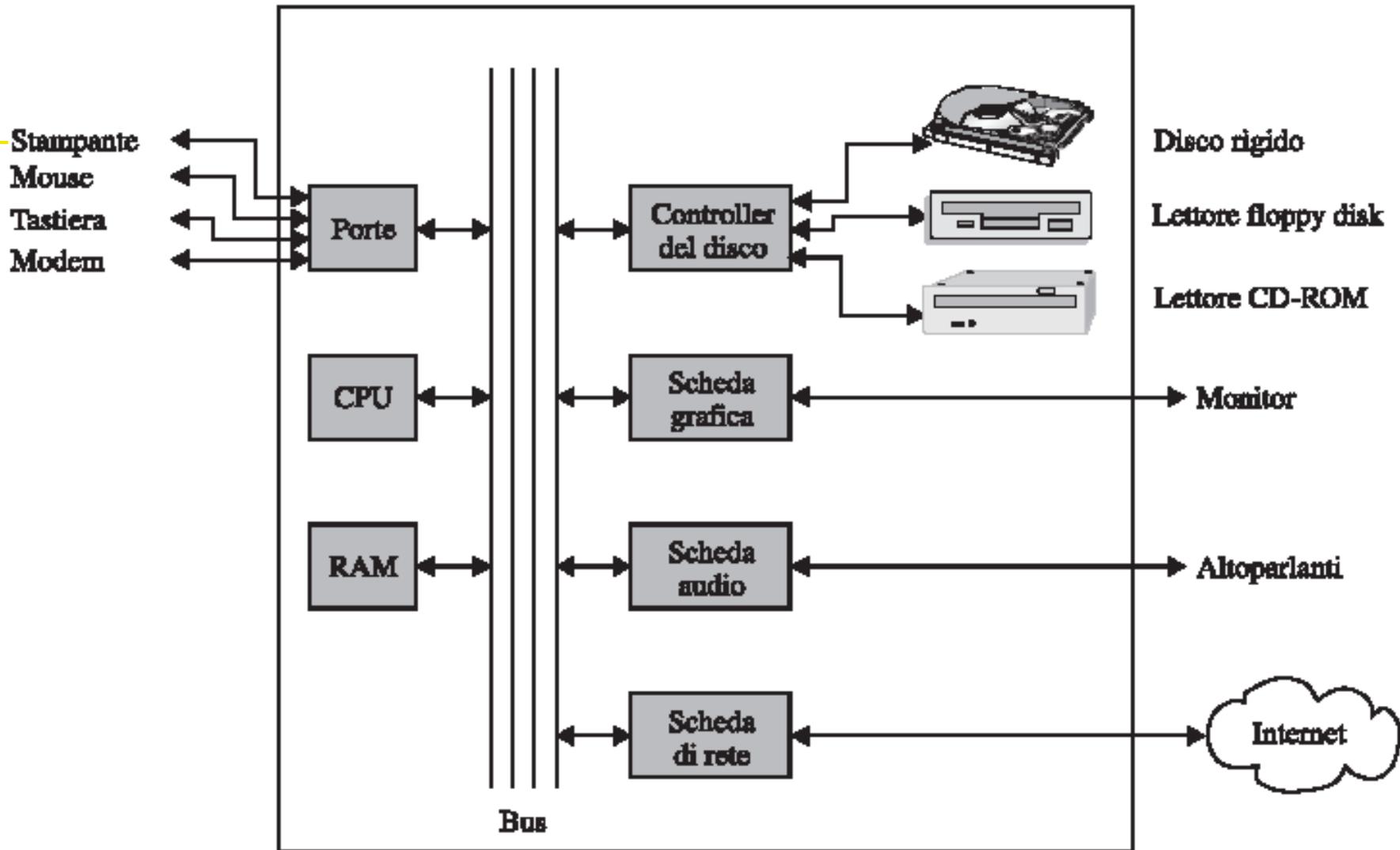


Figura 5 Diagramma schematico di un personal computer

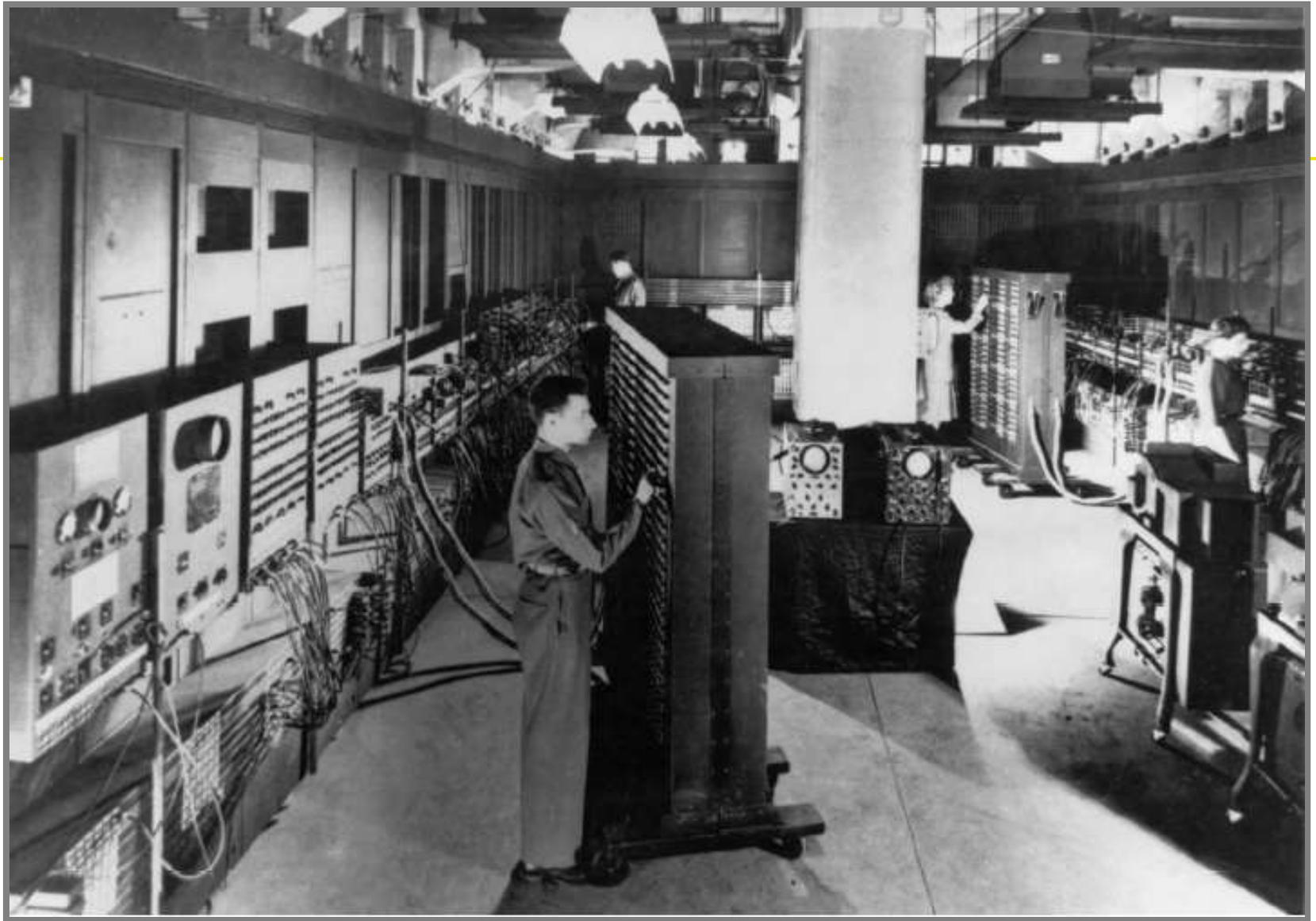


Figura 6 L'ENIAC

©Apogeo 2007

Codice macchina

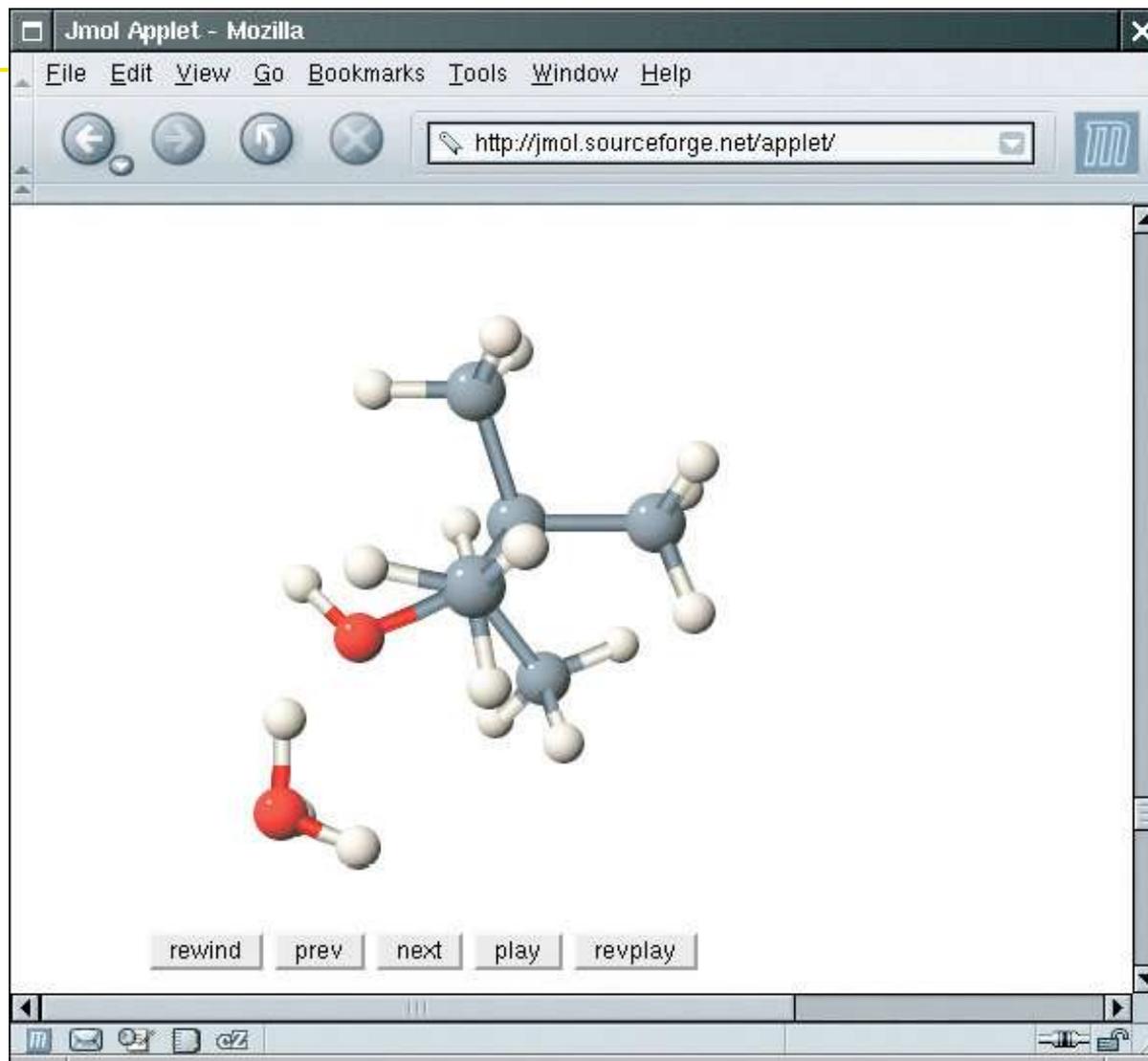
- Java Virtual Machine (JVM) – una tipica sequenza di istruzioni macchina:
 1. Carica il contenuto della posizione di memoria 40.
 2. Carica il valore 100.
 3. Se il primo valore è maggiore del secondo, prosegui con l'istruzione contenuta nella posizione di memoria 240.
- Le istruzioni macchina sono codificate sotto forma di numeri:

```
21 40 16 100 163 240
```
- Il compilatore traduce la descrizione di alto livello nelle istruzioni macchina

Il linguaggio di programmazione Java

- Semplice
- Sicuro
- Indipendente dalla piattaforma
- Ha una ricca libreria
- Pensato per internet

Figura 7
Un applet
che visualizza
molecole
in una pagina
Web



Prendere confidenza con il computer

- Log in
- Individuare il compilatore Java
- Capire file e folder
 - I programmi sono memorizzati in file
 - File: un insieme di informazioni memorizzate insieme; un file Java è un contenitore di istruzioni Java.
 - I file hanno un nome e le regole per i nomi validi sono diverse da un sistema all'altro.
 - I file sono conservati in cartelle o directory. Si tratta di contenitori di file, che si possono annidare, ovvero inserire uno dentro l'altro.

Prendere confidenza con il computer

- Scrivere un programma semplice (later)
- Salvare il proprio lavoro
 - Sviluppare una strategia di salvataggio del proprio lavoro, fare copie di backup.

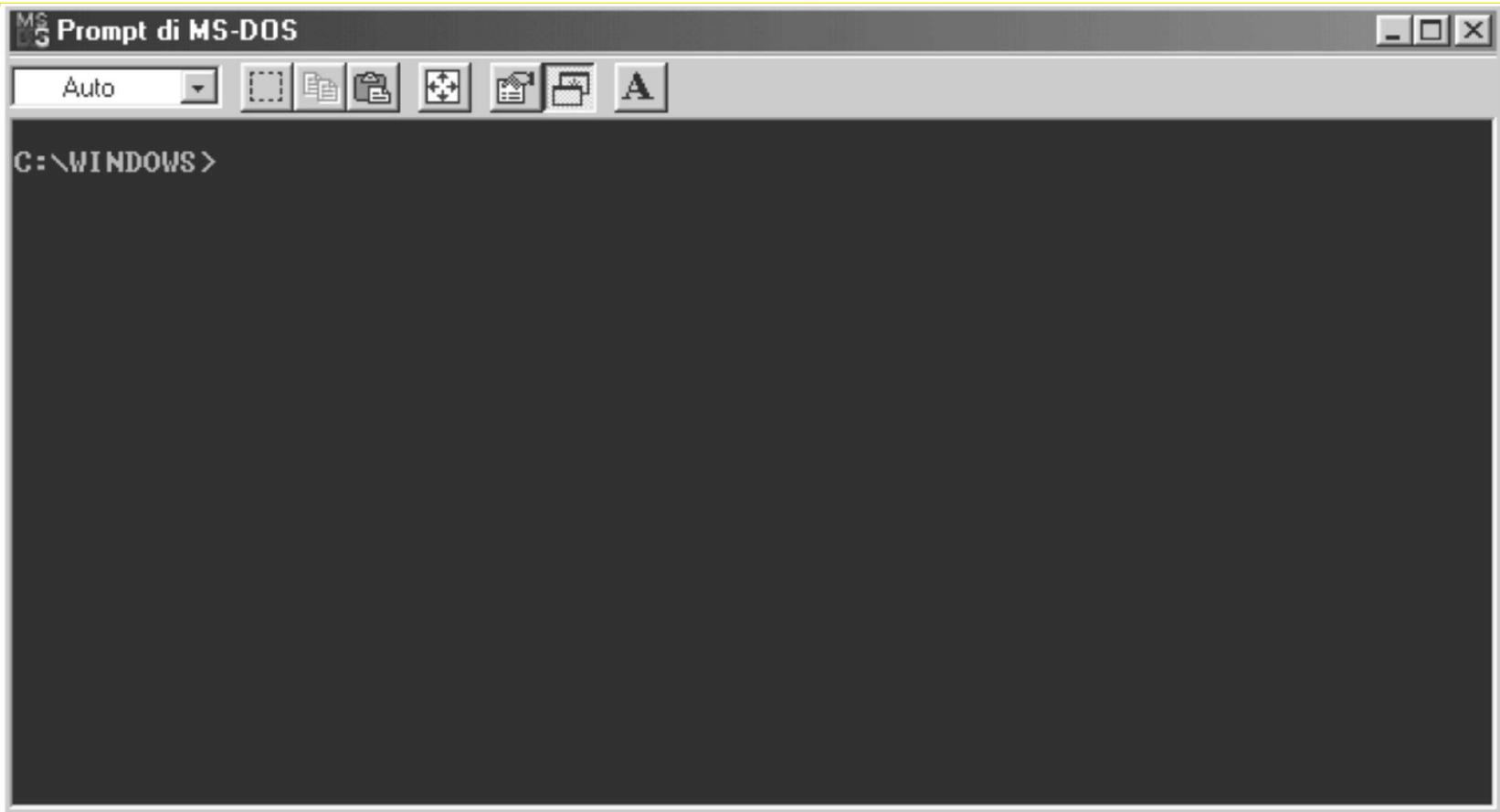


Figura 8 Una finestra di comando (shell)

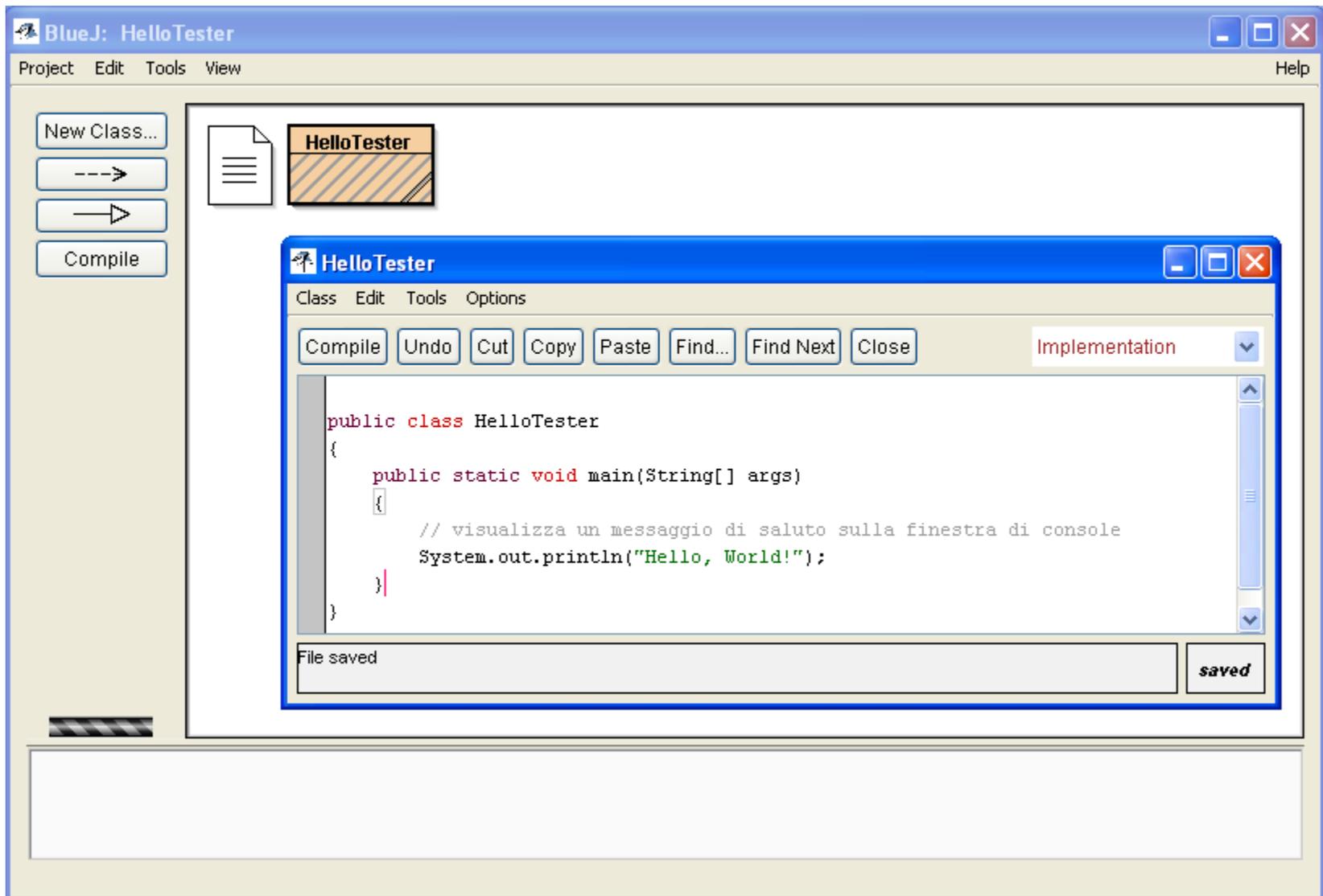
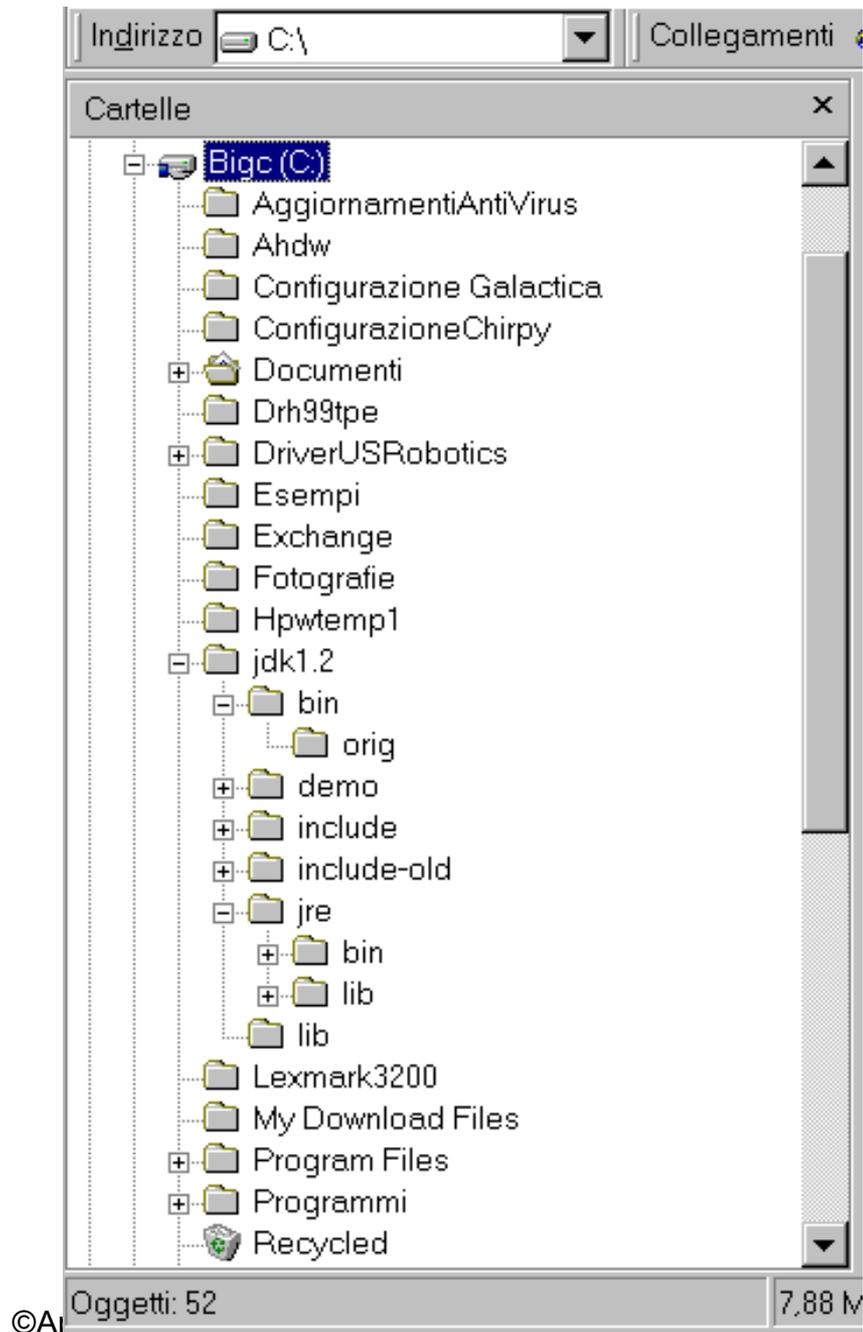


Figura 9 Un ambiente di sviluppo integrato

Figura 10
Cartelle annidate

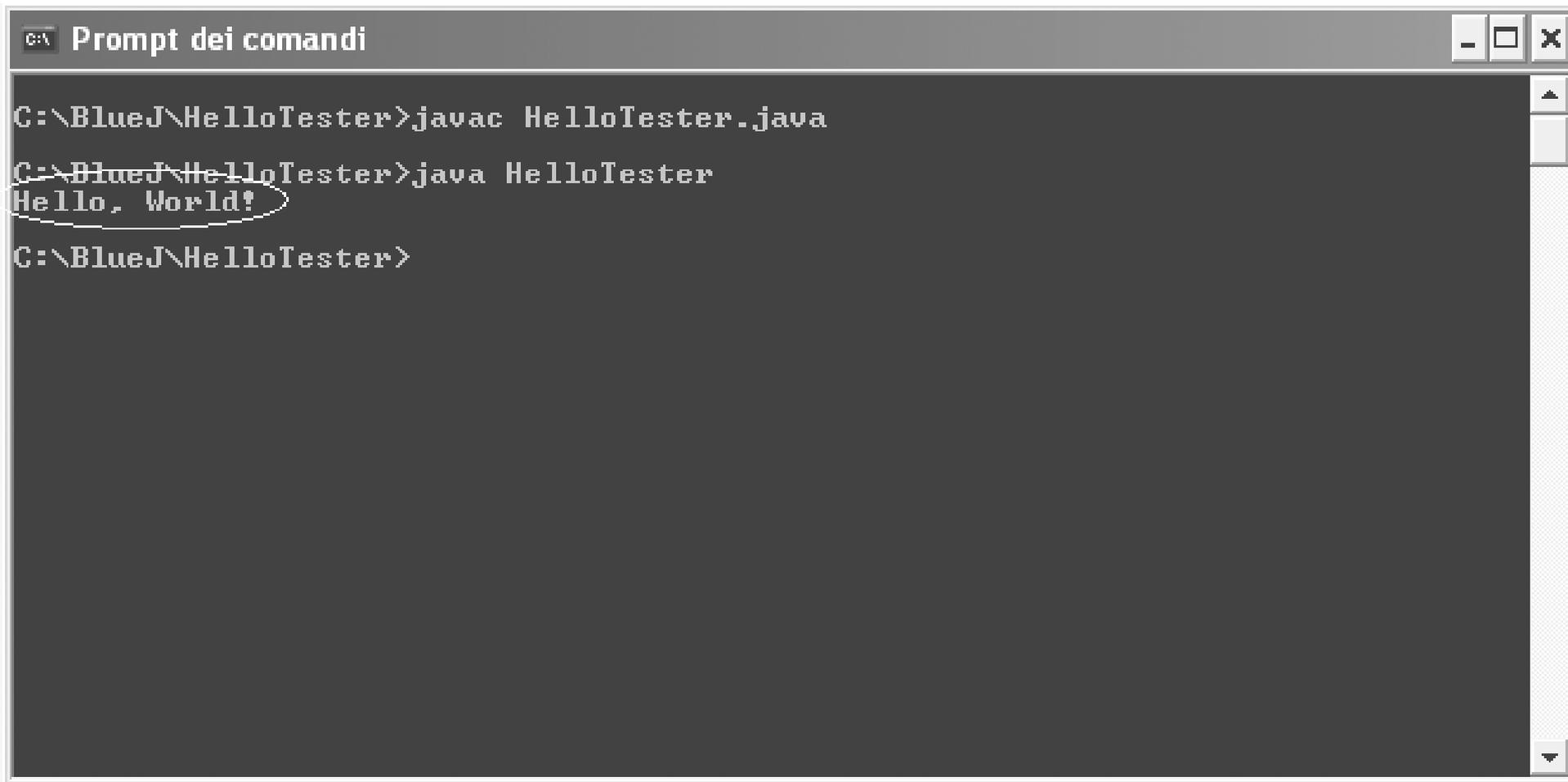


File HelloTester.java

```
1: public class HelloTester
2: {
3:     public static void main(String[] args)
4:     {
5:         // visualizza un messaggio di saluto sulla finestra di console
6:
7:         System.out.println("Hello, World!");
8:     }
9: }
```

Visualizza

```
Hello, World!
```



```
C:\> Prompt dei comandi

C:\BlueJ\HelloTester>javac HelloTester.java

C:\BlueJ\HelloTester>java HelloTester
Hello, World?

C:\BlueJ\HelloTester>
```

Figura 11

Esecuzione del programma HelloTester in una finestra di comando

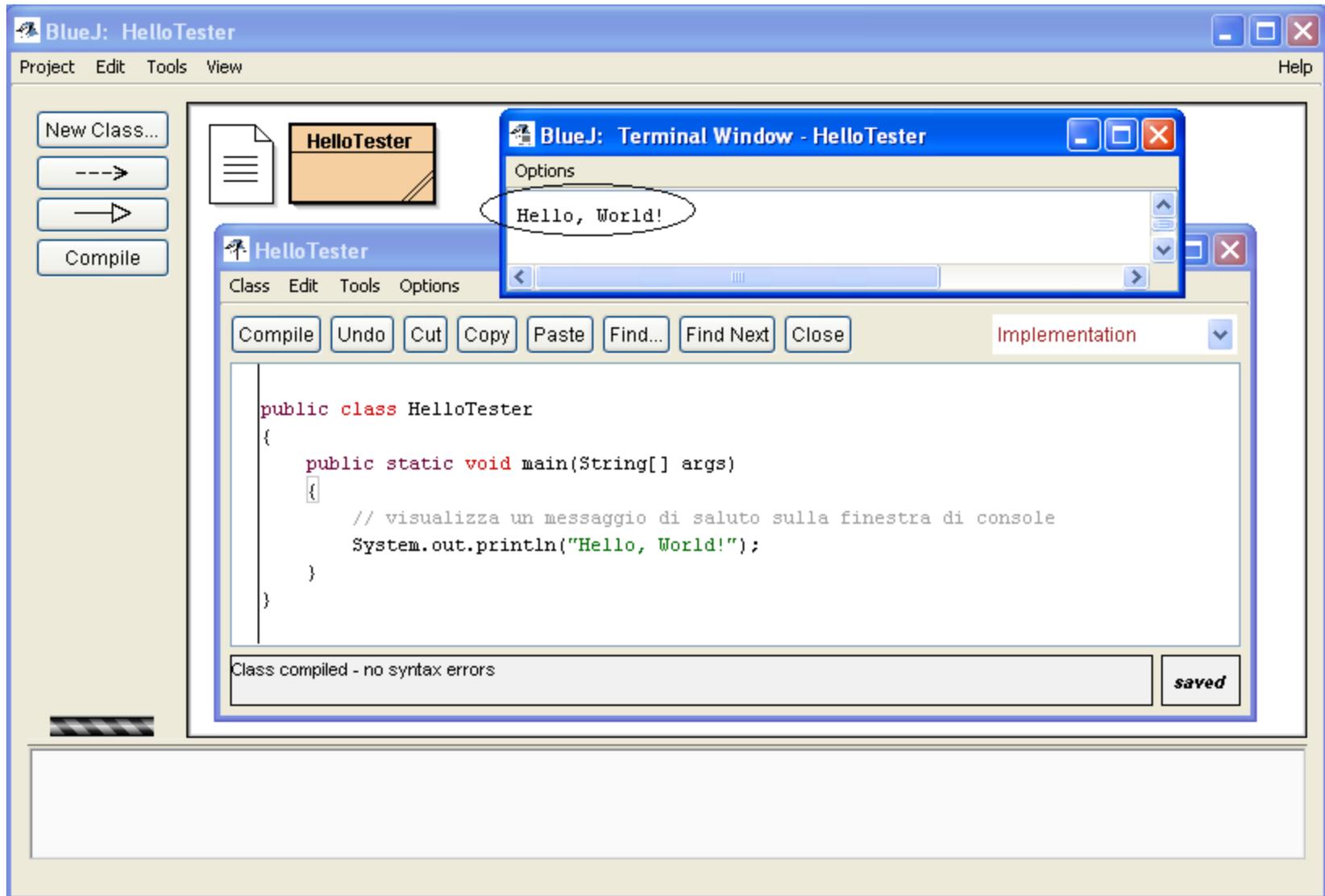
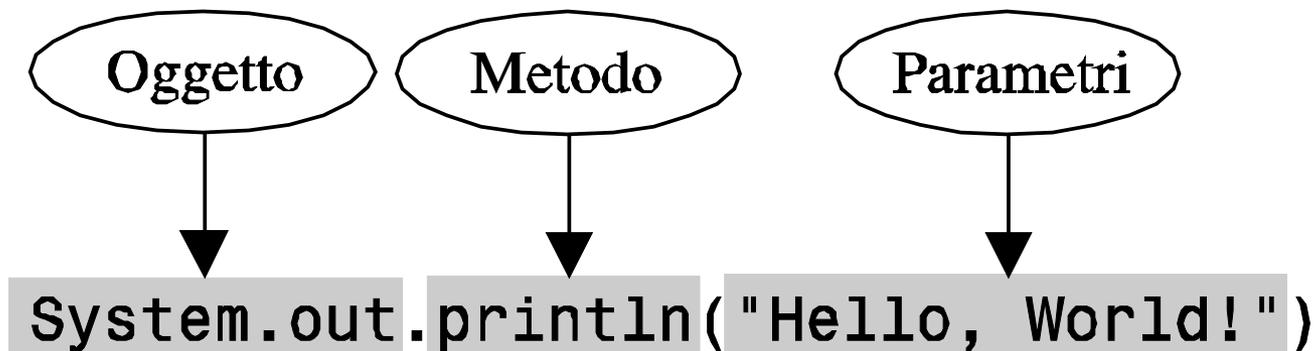


Figura 12

Esecuzione del programma HelloTester in un ambiente di sviluppo integrato

Un semplice programma

- `public class NomeClasse`
- `public static void main(String[] args)`
- `// comment`
- **Method call**



System Class
System.out Oggetto
println Metodo

Figura 13
Invocazione di un metodo

Sintassi di Java

1.1: Invocazione di metodo

object.methodName(parameters)

Esempio:

```
System.out.println("Hello, Dave!");
```

Obiettivo:

Invocare il metodo di un oggetto e fornire eventuali parametri aggiuntivi.

Errori

- Errore di sintassi

```
System.ouch.print(". . .");  
System.out.print("Hello);
```

- Identificato dal compilatore
- Errore di esecuzione o errore logico

```
System.out.print("Hell");
```

- Identificato tramite collaudo

Il processo di compilazione

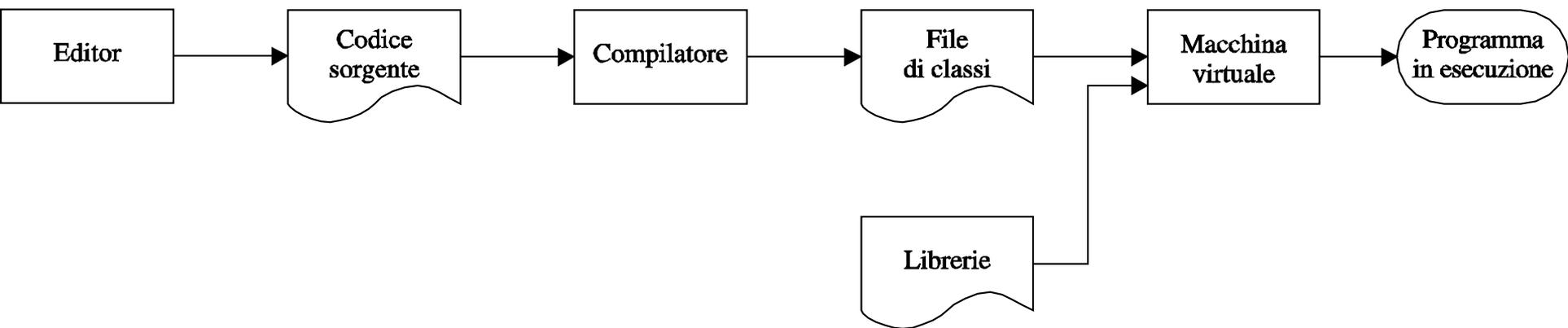


Figura 14

Dal codice sorgente al programma in esecuzione

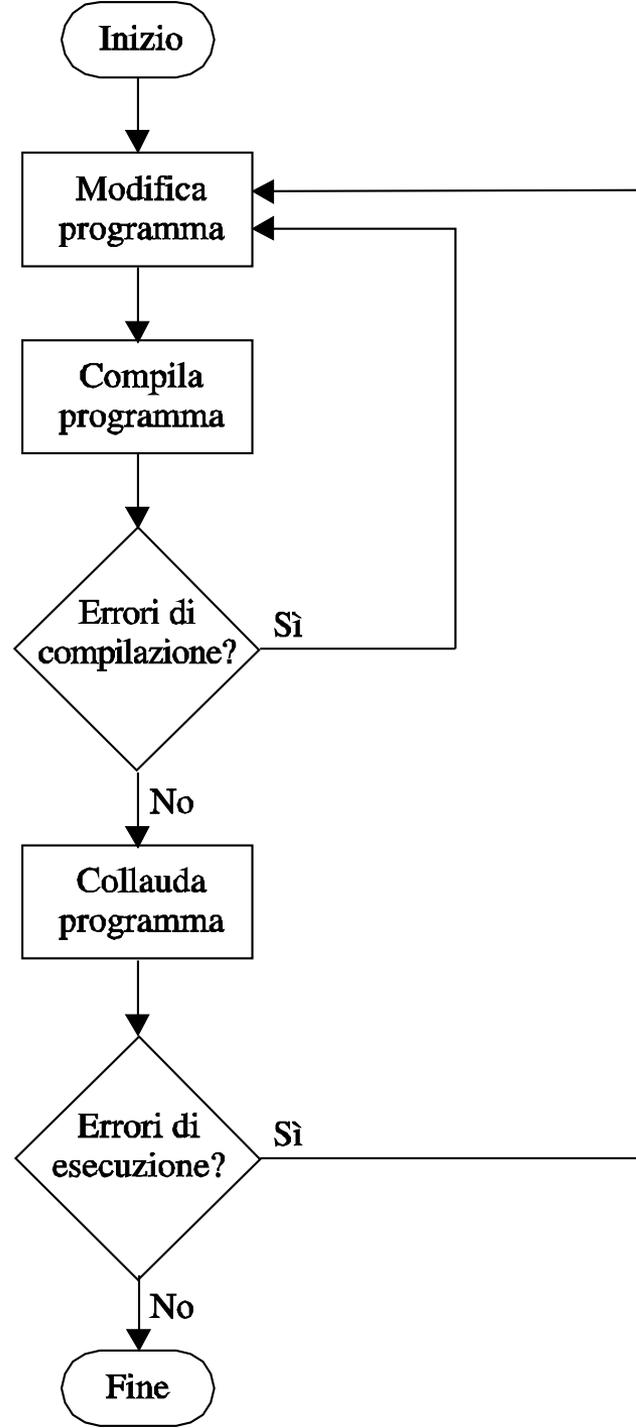


Figura 15
Il ciclo modifica-compila-collauda