

# I LINGUAGGI DI PROGRAMMAZIONE

## ISTRUZIONI MACCHINA

- **Linguaggio macchina**
  - può essere eseguito dalla CPU
  - dipende dal tipo di CPU
    - insieme di istruzioni macchina di una CPU: **instruction set**
- Istruzioni codificate in **configurazioni di bit**
  - es: **21 40 16 100 163 240**
    - 21: carica in un registro un valore 40
    - 16: carica su un altro registro valore 100
    - 163: se il primo valore è minore del secondo, procedi con istruzione ad indirizzo 240

## TIPOLOGIE DI ISTRUZIONI MACCHINA

- **Trasferimento dati**
  - LOAD
  - STORE
- **Operazioni aritmetiche e logiche** (eseguite da ALU)
  - ADD, SUB, MUL, DIV
  - AND, OR, NOT
- **Salti**
  - salto incondizionato
    - **JUMP**
  - salto condizionato
    - **JZ** (salto se valore è zero)
    - **JGZ** (salta se maggiore di zero)

## LINGUAGGIO ASSEMBLY

- Permette di scrivere il programma mediante dei nomi abbreviati
  - codici mnemonici
- Assembler: traduce linguaggio assembly in linguaggio macchina
- Permette di usare variabili
- Vantaggi
  - molto + comodo da usare del linguaggio macchina

- Svantaggi
  - richiede molte istruzioni per eseguire operazioni semplici
  - dipende dal set di istruzioni di ogni CPU

```
iload      rate
bipush     maxRate
if_icmpgt  tooMuch
```

## LINGUAGGIO DI ALTO LIVELLO

- Facile da usare per il programmare
- Non comprensibili direttamente dalla CPU
- Storia dei linguaggi di programmazione
  - FORTRAN (primo linguaggio di programmazione)
  - BASIC, COBOL
  - Pascal, C (nasce programmazione strutturata)
  - C++ (nasce programmazione ad oggetti)
  - Java
  - Python

## COMPILATORE

- Traduce linguaggio ad alto livello in istruzioni macchina
  - codice sorgente -> codice eseguibile
- Scrivi un programma una volta -> puoi compilarlo per ogni architettura

## ERRORI DI SINTASSI

- I linguaggi formali hanno una sintassi rigida
  - se fai un errore -> non esegue il programma

## IL LINGUAGGIO JAVA

- Nasce nel 1991
- **"Compile once, execute everywhere"**
  - compili il codice una volta e lo puoi eseguire su tutte le architetture
  - codice compilato per JVM (installato in tutti i dispositivi)
    - JVM esegue il codice compilato
- Presenta una ricca **libreria standard**
  - messa a disposizione dal Java

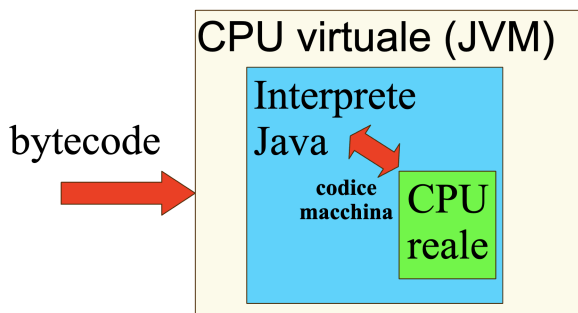
- **JVM**
  - Java Virtual Machine
  - comprende uno specifico set di istruzioni
- HotJava
  - browser che poteva eseguire programmi Java -> indipendentemente dalla piattaforma

## **COME COMPILARE IL CODICE**

```
javac NomeFile.java
```

## **ESECUZIONE PROGRAMMA**

- **Compilatore Java**
  - compila Java in bytecode
  - NomeFile.java -> NomeFile.class
- **Interprete Java**
  - traduce "al volo" bytecode in istruzioni macchina
  - NomeFile.class -> codice binario
  - può utilizzare le librerie Java grazie al JDK



## **LINGUAGGI COMPILATI VS LINGUAGGI INTERPRETATI**

- **Linguaggi compilati**
  - C++
  - massima **efficienza**
  - devono essere compilati per ogni processore
    - generano file eseguibile
- **Linguaggi interpretati**
  - Python
  - massima **portabilità**
  - il codice sorgente viene interpretato direttamente

- **Java**

- codice sorgente -> bytecode (codice macchina per JVM) -> codice macchina per CPU
  - il codice sorgente viene compilato
  - il bytecode viene interpretato (ma velocemente)
- linguaggio misto

### Domande

- Quali sono i due principali vantaggi del linguaggio Java?