

# Concetti di base

## 1 Informatica

Si occupa dell'informazione e del suo trattamento automatico tramite mezzi

- fisici (computer)
- logici (algoritmi)

## 2 Computer

Una macchina elettronica *programmabile* al fine di svolgere diverse funzioni.

È un tipo di *esecutore*.

## 3 Programmazione

La programmazione si suddivide in due fasi:

1. individuazione del processo risolutivo del problema (*algoritmo*)
2. codifica dell'algoritmo in un insieme di *istruzioni* per un *esecutore*, il quale solitamente è un linguaggio di programmazione

Può anche essere definita come l'organizzazione di istruzioni elementari, direttamente comprensibili dall'esecutore, in strutture complesse (*programmi*) per svolgere determinati compiti.

## 4 Algoritmo

Insieme ordinato di passi

- eseguibili
- non ambigui
- che definiscono un processo che termina

Un algoritmo codifica tutta l'“intelligenza” necessaria per la risoluzione di un problema: l'esecutore può quindi svolgerlo in modo “meccanico”.

#### 4.1 Esempio

Algoritmo di Euclide: calcola il MCD di due numeri interi  $x$  e  $y$

1. calcola il resto della divisione di  $x$  per  $y$
2. se il resto è diverso da 0 ricomincia dal passo 1 con  $x \leftarrow y$  e  $y \leftarrow$  resto altrimenti prosegui
3. MCD è il valore attuale di  $y$

#### 4.2 Controesempio

1. crea un elenco di tutti i numeri primi
2. ordina l'elenco in modo decrescente
3. preleva il primo

Non è un algoritmo perché le istruzioni non sono effettivamente eseguibili:

- non è possibile creare un elenco infinito (passo 1)
- non esiste un numero primo più grande di tutti gli altri (passi 2 e 3)

### 5 Programma

Una sequenza di *istruzioni elementari* che un computer è in grado di comprendere ed eseguire.

È un'espressione concreta di un algoritmo astratto. Per questo, la scrittura di un programma è successiva all'individuazione dell'algoritmo.

## 6 Macchina di Von Neumann

Modello ideale di esecutore, introdotto nel 1946.

È composta da

- memoria, che contiene
  - le istruzioni del programma
  - i dati utilizzati
- processore, l'esecutore “vero e proprio”
  - ripete un ciclo *fetch-decode-execute*
  - ha un proprio *linguaggio macchina*, che definisce il formato delle istruzioni

## 7 Linguaggio macchina

Ogni istruzione è rappresentata da una sequenza di bit, che codificano

- l'operazione da eseguire
- gli operandi coinvolti (registri, locazioni di memoria, costanti, ecc.)

Il linguaggio macchina ha anche una rappresentazione simbolica, chiamata *assembler*.

### 7.1 Svantaggi

- è necessario conoscere i dettagli dell'architettura
- è impossibile trasportare i programmi a macchine diverse
- i programmi scritti in linguaggio macchina sono difficili da comprendere, modificare e correggere perché
  - vengono utilizzati “trucchi” legati a particolari caratteristiche dell'architettura
  - la struttura logica del codice è nascosta

## 8 Linguaggi ad alto livello

Risolvono gli svantaggi del linguaggio macchina tramite la definizione di esecutori astratti di livello più alti.

Uno dei principali vantaggi è la **portabilità del sorgente**.

## 8.1 Implementazione

L'implementazione di una macchina astratta, che comprende un linguaggio  $L$ , su una macchina reale  $M$  può avvenire in due modi:

**compilatore:** traduce un programma dal linguaggio  $L$  al linguaggio macchina di  $M$

**interprete:** simula direttamente la macchina astratta:

1. legge un'istruzione del programma in linguaggio  $L$
2. esegue le istruzione di  $M$  corrispondenti

## 9 Java Virtual Machine

La traduzione del linguaggio Java avviene tramite un approccio ibrido, il quale sfrutta sia la compilazione che l'interpretazione:

1. Il codice Java viene tradotto in codice per un *esecutore astratto a basso livello*, la **Java Virtual Machine (JVM)**. Tale codice viene chiamato **bytecode**.
2. Il bytecode viene interpretato.

Questa tecnica consente di ottenere anche la **portabilità dell'eseguibile**.