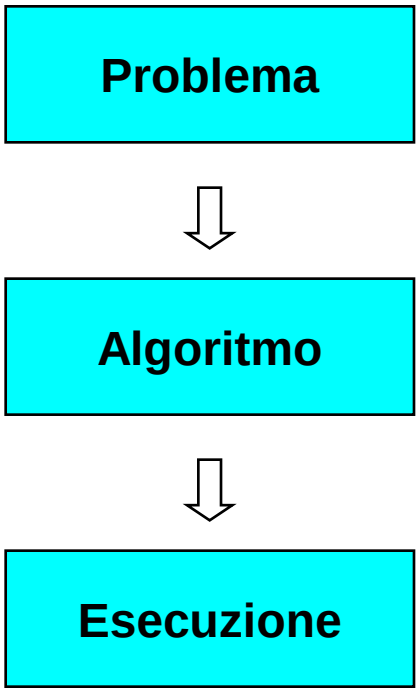
Linguaggi di Programmazione e Paradigmi

Prof. Francesco Accarino
IIS Altiero Spinelli Sesto San Giovanni

Come risolvere un problema

Le 3 tappe fondamentali di risoluzione di un problema algoritmico sono:



Formulazione del problema da risolvere

Costruzione del metodo di risoluzione del problema

Esecuzione dell'algoritmo progettato da parte di un esecutore (umano o meccanico)

L'esecutore calcolatore

Se l'esecutore dell'algoritmo è un **calcolatore**

è necessario **implementare** (o codificare) l'algoritmo tramite un opportuno

Linguaggio di Programmazione e ottenere quindi un **Programma**

La Programmazione

La **Programmazione** (in Informatica) è l'attività svolta per creare un **Programma**

Il linguaggio può essere:

- **direttamente** comprensibile dal calcolatore
- **non direttamente** comprensibile dal calcolatore

Cos'è un linguaggio

Un linguaggio è un insieme di parole e di metodi di combinazione delle parole comprese da una comunità di persone. (Linguaggio Umano)

- Vantaggi: Ricchezza espressiva;
- Svantaggi: Ambiguità Ridondanza.

Linguaggio Macchina (e Assembler):

Linguaggio legato alla struttura fisica del Calcolatore (CPU);

Potente e veloce ma i programmi sono molto lunghi e di difficile scrittura.

In linguaggio macchina una istruzione è rappresentata da codici binari che rappresentano l'operazione da svolgere e su quali operandi

Assembler	Linguaggio Macchina	
MOV AL,05	0011111000000101	Corrispondenza1 a 1

Linguaggio di Programmazione

- Programma:
  - formulazione di un algoritmo nei termini di un linguaggio di programmazione.
- Linguaggio di Programmazione:
  - Linguaggio intermedio fra il linguaggio macchina e il linguaggio naturale;
  - Descrive gli algoritmi con una ricchezza espressiva comparabile con quella dei linguaggi naturali, ma non è ambiguo;
  - Descrive gli algoritmi in modo rigoroso.

Lessico, Sintassi e Semantica

- Lessico: l'insieme di regole formali per la scrittura di parole in un linguaggio
- Sintassi: l'insieme di regole formali per la scrittura di frasi in un linguaggio, che stabiliscono cioè la grammatica del linguaggio stesso
- Semantica: l'insieme dei significati da attribuire alle frasi (sintatticamente corrette) costruite nel linguaggio

– Nota: una frase può essere sintatticamente corretta e tuttavia non avere significato!

I linguaggi di programmazione

I linguaggi di programmazione si possono suddividere fondamentalmente in tre categorie:

1. **Linguaggio Macchina**
2. **Linguaggio Assembler**
3. **Linguaggi ad alto livello**

Basso livello

Linguaggio Macchina

- E' una codifica utilizzata fino alla prima metà degli anni '50
- Gli algoritmi vengono codificati in word di 0 e 1 (word di bit)
- Il calcolatore esegue direttamente il programma

Dalla seconda metà degli anni '50 il linguaggio di codifica incomincia ad alzarsi di livello e viene affidato alla macchina il faticoso compito di tradurre il programma in word di bit…

Linguaggio Assembler

- Si avvicina molto al linguaggio macchina
- Definisce un set di operazioni elementari
- Esiste quasi una corrispondenza quasi biunivoca tra istruzione e numero di operazioni eseguite dal sistema
- Il calcolatore non esegue direttamente il programma

… e necessita di un **traduttore** (Assemblatore) che lo trasformi in word di bit (affinché il calcolatore lo possa eseguire)

Linguaggi ad alto livello

- Si avvicinano molto al linguaggio naturale
- Le istruzioni sono molto astratte e ognuna di esse può corrispondere ad un elevato numero di operazioni macchina
- Sono di facile utilizzo
- Il calcolatore non esegue direttamente il programma

… e necessita di un **traduttore** (Compilatore) che lo trasformi in word di bit (affinché il calcolatore lo possa eseguire)

Linguaggi ad alto livello

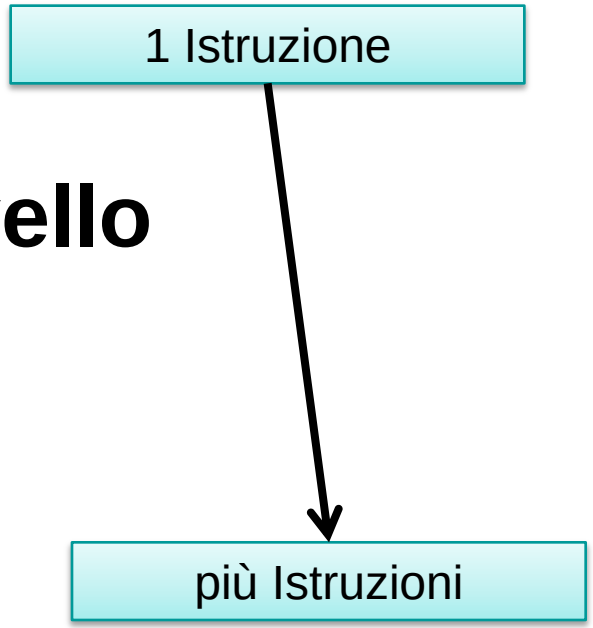
Esempi

- **Java**
- **C**
- **C++**
- **Pascal**
- **etc.**

Comparazione tra linguaggi

Linguaggio ad alto livello

1. Esci dalla stanza

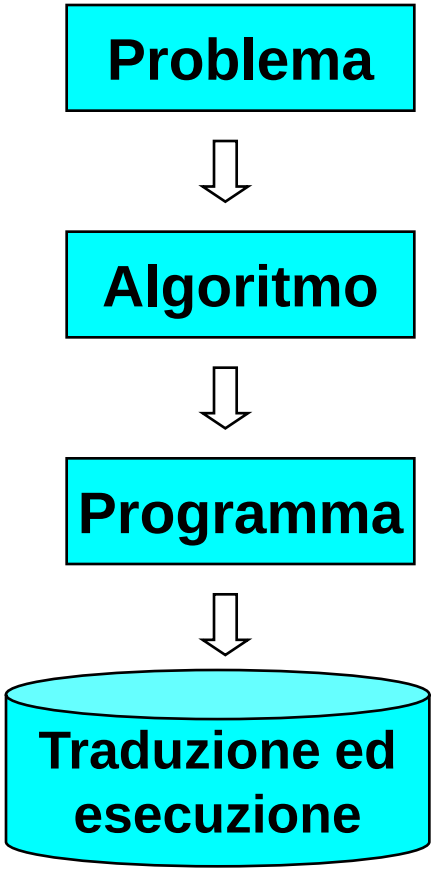


Linguaggio a basso livello

1. Individua la porta
2. Girati verso la porta
3. Cammina verso la porta
4. Gira la maniglia
5. Apri la porta ed esci

Come risolvere un problema

… su calcolatore



Formulazione del problema da risolvere

Costruzione del metodo di risoluzione del problema

Codifica dell'algoritmo in un Linguaggio di programmazione

Traduzione del programma in linguaggio macchina (dal momento che nessuno utilizza più il linguaggio macchina…) ed esecuzione

Un po' di terminologia…

- Programmare in un determinato linguaggio ad alto livello significa produrre un file di puro testo che prende il nome di **codice sorgente** (o semplicemente sorgente)

- La traduzione in linguaggio macchina di un codice sorgente prende il nome di **codice oggetto**(o codice macchina)

Il traduttore

- E' necessario quando si programma utilizzando un linguaggio diverso dal linguaggio macchina
- Traduce il codice sorgente del programma in **istruzioni macchina** (word di bit)
- Le istruzioni macchina sono operazioni elementari che agiscono direttamente su registri e memorie della CPU
- Dipende dal linguaggio di programmazione usato e dal tipo di processore (CPU) montato sul calcolatore

Tipi di traduttore

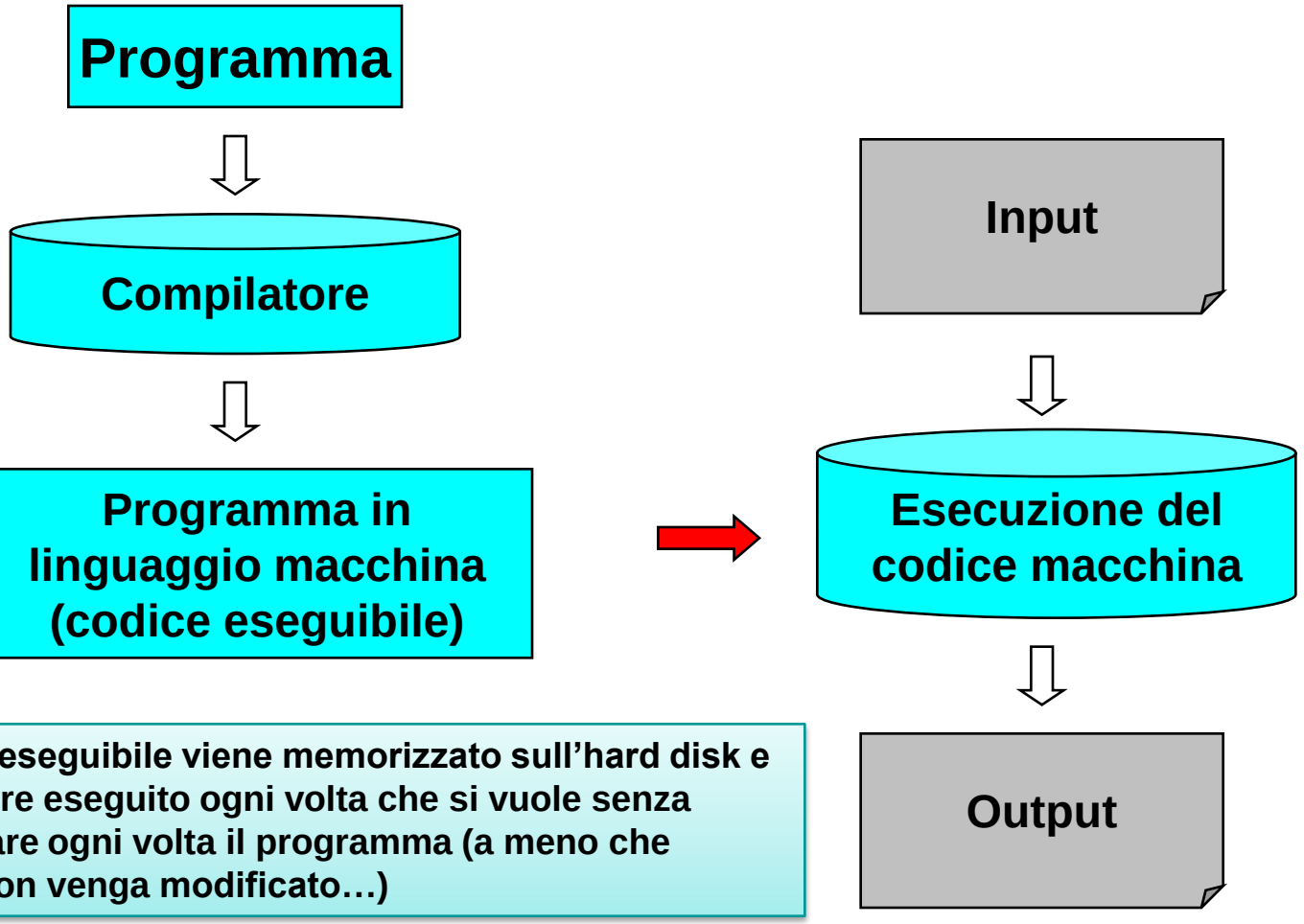
1. **Compilatore**

   Traduce in linguaggio macchina un programma scritto in uno specifico linguaggio ad alto livello

2. **Interprete**

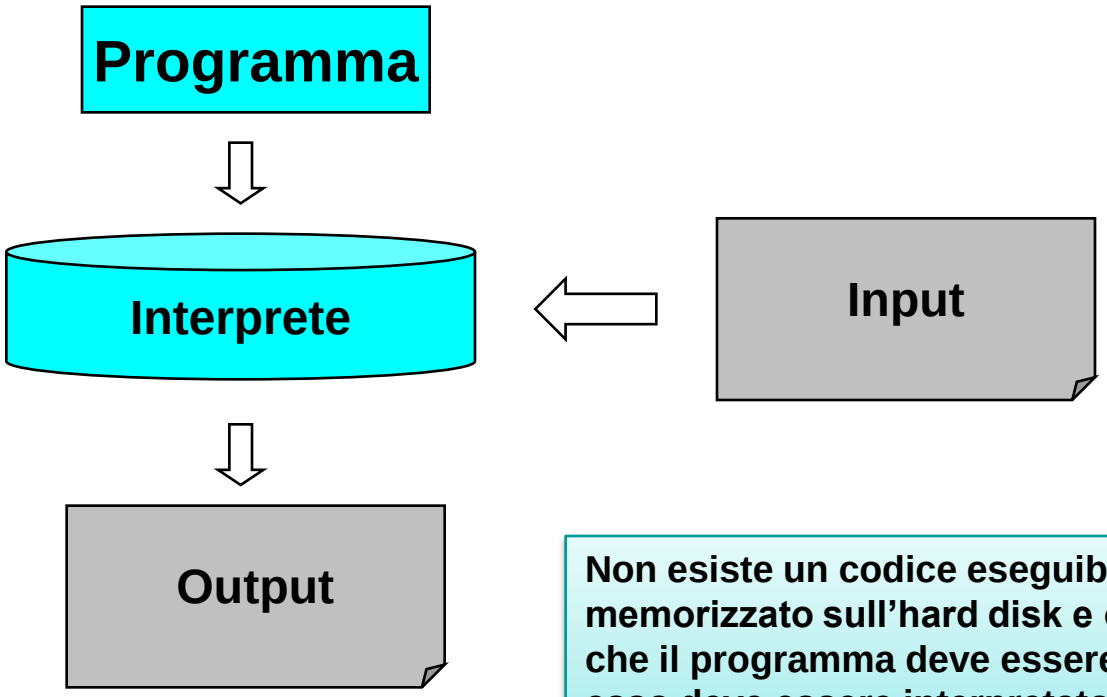
   Analizza e traduce il programma scritto in linguaggio ad alto livello e contemporaneamente compie le corrispondenti elaborazioni

Il compilatore



Il codice eseguibile viene memorizzato sull'hard disk e può essere eseguito ogni volta che si vuole senza ricompilare ogni volta il programma (a meno che questo non venga modificato…)

L'interprete



Non esiste un codice eseguibile memorizzato sull'hard disk e ogni volta che il programma deve essere eseguito, esso deve essere interpretato

Compilatore vs Interprete

Compilatore

- Esecuzione efficiente
- Codice eseguibile non portabile
- Ogni modifica del programma richiede una nuova compilazione

Interprete

- Esecuzione meno efficiente
- Programma portabile
- Esecuzione immediata ad ogni modifica

La traduzione dei programmi Java

1. Un compilatore Java produce un codice intermedio, il "**Byte Code**"

   che è codice portabile a basso livello, simile all'assembler, indipendente dall'hardware e invisibile al programmatore Java

2. Un interprete traduce ed esegue il "**Byte Code**"

Gli errori di programmazione

Gli errori in programmazione si possono dividere in:

- Errori di sintassi
- Errori logici
- Errori in *runtime* (di esecuzione)

Errori di sintassi

Gli **errori di sintassi** sono gli errori ortografici che si commettono nella fase di scrittura del programma nel linguaggio scelto.

Sono i più facili da correggere in quanto il programma non viene compilato e il compilatore stesso li identifica automaticamente tramite messaggi di errore.

Il programma non può essere dunque eseguito finché tutti gli errori sintattici non vengono eliminati.

Errori di sintassi

Ad esempio scrivere

```
System.out.pintln(“Hello world\n”);
```

al posto di

```
System.out.println(“Hello world\n”);
```

è un errore di sintassi

Errori logici

Gli **errori logici** sono gli errori che si commettono nella fase di progettazione dell'algoritmo prima di scrivere il programma nel linguaggio scelto.

Sono dovuti ad una mancata comprensione del problema o dei vincoli che devono soddisfare i dati in input.

Sono difficili da correggere in quanto il compilatore non è in grado di identificarli.

Se un programma viene compilato ed eseguito senza messaggi di errore, ciò NON significa che sia corretto!

Errori logici

Ad esempio scrivere

```
media=somma5Valori/2;
```

al posto di

```
media=somma5Valori/5;
```

è un errore logico se in realtà il programma deve calcolare la media tra 5 valori

Errori in *runtime*

Gli **errori in *runtime*** sono gli errori che possono verificarsi nella fase di esecuzione del programma anche se il programma è corretto (assenza di errori logici) e viene compilato (assenza di errori di sintassi).

Sono difficili da correggere in quanto il compilatore non è in grado di identificarli.

Errori in *runtime*

Ad esempio se durante la fase di esecuzione di un programma si tenta di effettuare una divisione per 0, si ha un errore in *runtime*

A=5

B=0

C=A/5

Gli errori di programmazione

In conclusione:

- Gli errori di sintassi bloccano il compilatore che non riesce a produrre il codice eseguibile
- Gli errori in *runtime* bloccano l'esecuzione del programma
- Gli errori logici non bloccano nulla ma l'output del programma non è corretto

Un pò Di terminologia

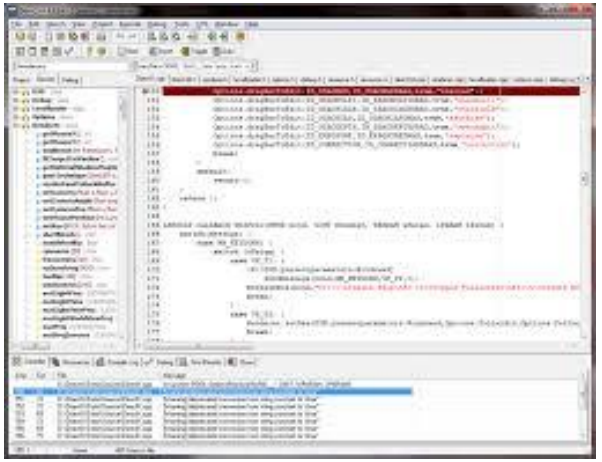
- SDK: **Software Development Kit**
  pacchetto di sviluppo per applicazioni", e sta a indicare genericamente un insieme di strumenti per lo sviluppo e la documentazione di software.

- Contiene normalmente:
  - un compilatore, per tradurre il codice sorgente in un eseguibile;
  - librerie standard dotate di interfacce pubbliche dette API - Application programming interface;
  - documentazione sul linguaggio di programmazione per il quale l'SDK è stato sviluppato e sugli strumenti a disposizione nell'SDK stesso;

Un pò Di terminologia

- IDE: **integrated development environment**
  Normalmente è uno strumento software che consiste di più componenti, da cui appunto il nome *integrato*:

  - un editor di codice sorgente;
  - un compilatore e/o un interprete;
  - Un Linker
  - un tool di building automatico;
  - (solitamente) un debugger.



Un pò Di terminologia

- Frame Work

  In informatica, e specificatamente nello sviluppo software, un **framework** è una struttura logica di supporto su cui un software può essere progettato e realizzato, facilitandone lo sviluppo riducendo il codice da scrivere.
- Contiene:
  - Un IDE
  - Svariate librerie
  - Svariati strumenti di progettazione

I paradigmi di programmazione

Un **paradigma di programmazione** è un modello concettuale che fornisce la "struttura" di un programma.

I principali paradigmi di programmazione sono:

- Programmazione Procedurale (o Imperativa)
- Programmaione Funzionale
- Programmazione Logica
- Programmazione Object-Oriented

Programmazione Procedurale

Un **programma è composto da istruzioni che realizzano trasformazioni di certo insieme di variabili in un certo momento dell'esecuzione:**

- C
- Pascal
- etc.

Programmazione Funzionale

Un **programma è una funzione che viene valutata per ottenere un risultato:**

- Lisp
- SML
- etc.

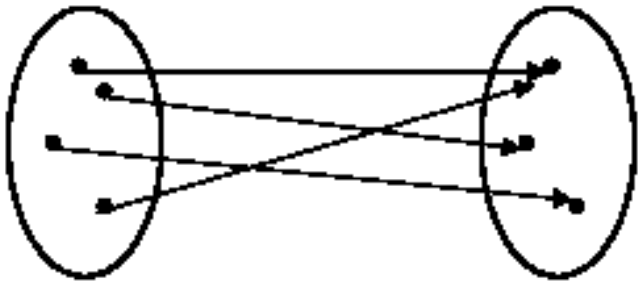
Programmazione Funzionale

Concetto primitivo: funzione

– Una **funzione** è una regola di associazione tra due insiemi, che associa a ogni elemento del primo insieme (dominio) un elemento del secondo insieme (codominio).

– La definizione di una funzione ne specifica il **dominio**, il **codominio** e la **regola di associazione**.

Ad esempio: **incr: N → N**
incr(x) = x + 1.

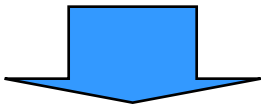


– Dopo averne dato definizione, una funzione può essere **applicata** a un elemento del dominio (argomento) per restituire l'elemento del codominio ad esso associato (**valutazione**): **incr(3)** = **4**.

Programmazione Funzionale

L'unica operazione utilizzata nel paradigma funzionale è l'applicazione di funzioni.

Il ruolo dell'interprete di un linguaggio funzionale è valutare il programma e ***produrre un valore***: nel paradigma funzionale "puro" il valore di una funzione è determinato ***soltanto dal valore dei suoi argomenti, e non da variabili*** (assenza di effetti collaterali).



Il concetto di variabile utilizzato è quello di variabile matematica → valori non mutabili (nessun assegnamento).

L'essenza della programmazione funzionale è
combinare funzioni

Programmazione Funzionale

Programma = Funzione

La struttura del programma consiste nella definizione di una funzione.

L'esecuzione del programma consiste nella valutazione di tale funzione.

La struttura dati fondamentale è la lista

Un programma è definito come una lista (programma *coincide* con struttura dati)

Programmazione Logica

Un **programma è un insieme di fatti e regole e la sua esecuzione equivale alla realizzazione di una dimostrazione:**

- Prolog
- LDL
- etc.

Paradigma logico

Concetto primitivo: deduzione logica

Base: logica formale

Obiettivo: formalizzare il ragionamento

Programmare significa:

– descrivere il problema con formule del linguaggio

– interrogare il sistema, che effettua deduzioni sulla base della conoscenza rappresentata

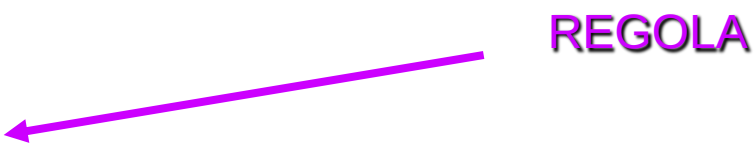
Linguaggio Prolog

ESEMPIO: due individui sono colleghi se lavorano per la stessa ditta

collega(X,Y):- lavora(X,Z), lavora(Y,Z), diverso(X,Y).

FATTI

lavora(emp1,ibm).
lavora(emp2,ibm).
lavora(emp3,txt).
lavora(emp4,olivetti).
lavora(emp5,txt).
:- collega(X,Y).



Programmazione Object-Oriented

Un **programma è un insieme di oggetti (astrazioni della realtà) dotati di proprietà (dati) e metodi (procedure) che comunicano tramite scambio di messaggi:**

- Java
- C++
- etc.