

ITIS-LS “Francesco Giordani” Caserta

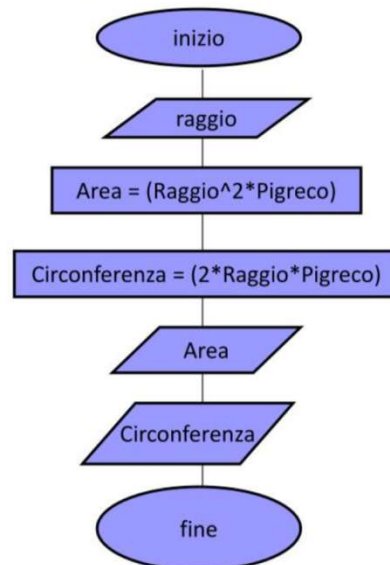
prof. Ennio Ranucci  
a.s. 2020-2021

Dal problema al risultato

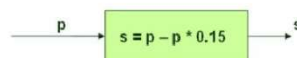
C++



Flow-chart: circonferenza e area



- **Problema:** Calcolare il prezzo scontato del 15%
- Input: il numero razionale p.
- Output: un numero razionale s ( $s = p - p \cdot 0.15$ )



## Problema

Dal vocabolario: Il problema è un quesito che attende una soluzione;

Difficoltà che richiede un adattamento o un comportamento particolare,

*2+2 è un problema per un bimbo di 1 elementare non lo è per un adulto.*

In matematica e in altre scienze, domanda con cui si chiede di trovare, sulla base di dati noti ed enunciati, dati non noti, logicamente deducibili dai primi.

Trovato il modo di risolvere il problema è prassi comune determinare un procedimento che lo risolva in un numero finito di passi. Questo procedimento è chiamato **Algoritmo**.

*Definizione non formale di Algoritmo:*

Un insieme di istruzioni che eseguite da un esecutore restituiscono il risultato.

*Definizione formale di Algoritmo:*

L'**algoritmo** è una strategia che serve per risolvere un problema ed è costituito da una sequenza **finita** di operazioni (dette anche **istruzioni**), consente di risolvere tutti i quesiti di una **stessa classe**.

[ In altre parole: un algoritmo non è altro che un procedimento che permette la risoluzione di specifici problemi mediante l'applicazione di una sequenza finita di precise istruzioni che, a loro volta, devono essere interpretate ed eseguite fino alla loro conclusione seguendo un ben preciso ordine.]

L'algoritmo deve rispettare le seguenti proprietà:

- i passi dell'algoritmo devono essere elementari, cioè non possono essere ulteriormente divisibili (**atomicità**);
- i passi dell'algoritmo non possono essere interpretati in altri modi (**non ambiguità**);
- l'algoritmo deve per forza essere svolto in un certo numero di specifici passi e, allo stesso tempo, deve richiedere in ingresso soltanto una determinata quantità di dati (**finitezza**);
- l'esecuzione dell'algoritmo deve terminare entro un certo periodo di tempo (**terminazione**);
- l'esecuzione dell'algoritmo deve portare ad un risultato univoco (**effettività**);
- ogni passo dell'algoritmo deve essere ben stabilito (**determinismo**).

In informatica per descrivere un algoritmo vengono utilizzati i seguenti due linguaggi formali:

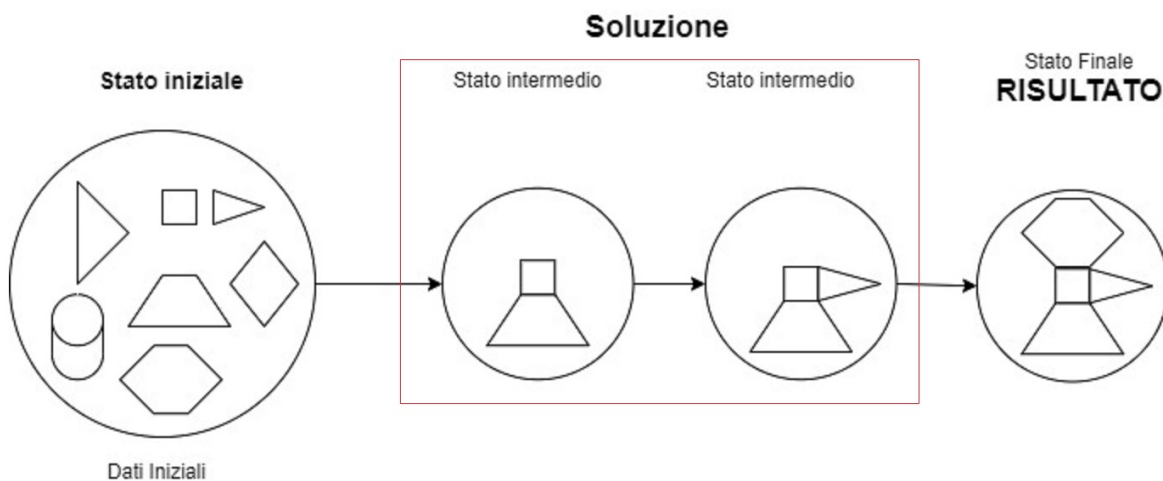
a) un linguaggio, basato sull'utilizzo di simboli grafici, per descrivere i singoli passi dell'algoritmo, detto linguaggio dei diagrammi a blocchi o **flow chart** (oppure diagramma di flusso);

b) un linguaggio, molto vicino al linguaggio naturale, detto linguaggio di progetto o **pseudolinguaggio** talvolta abbastanza vicino ad alcuni linguaggi di programmazione.

La descrizione dell'algoritmo effettuata con lo pseudolinguaggio prende il nome di pseudocodifica. La **pseudocodifica** è una fase intermedia che si frappone tra la fase di analisi/progettazione del problema e quella di codifica in un vero e proprio linguaggio di programmazione. Lo scopo principale della pseudocodifica è di portare l'utente ad esprimere le proprie istruzioni in una forma naturale, utilizzando frasi ed espressioni elementari della lingua italiana che però siano univocamente interpretabili. Ciò permette di concentrarsi sulla soluzione logica del problema invece che sulla forma e sui vincoli da rispettare nella sua enunciazione.

Un automa può essere un sistema informatico che risponde ai nostri input (digitazioni di dati, numeri, parole, ecc.....) fornendoci, come output una risposta o una azione differente a seconda della sua situazione interna.

**Un automa può essere un sistema informatico che riceve in ingresso i dati iniziali del problema, elabora i dati forniti(soluzione) e restituisce il risultato.**

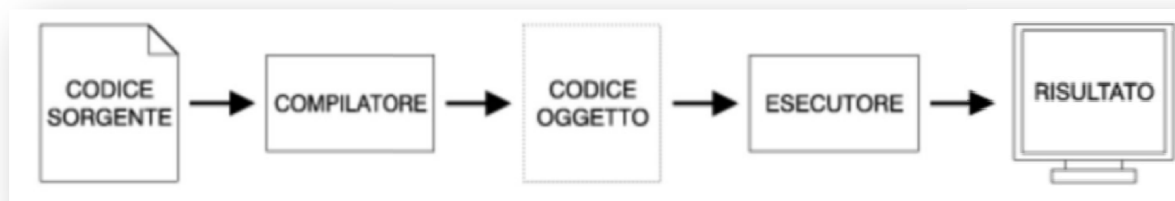
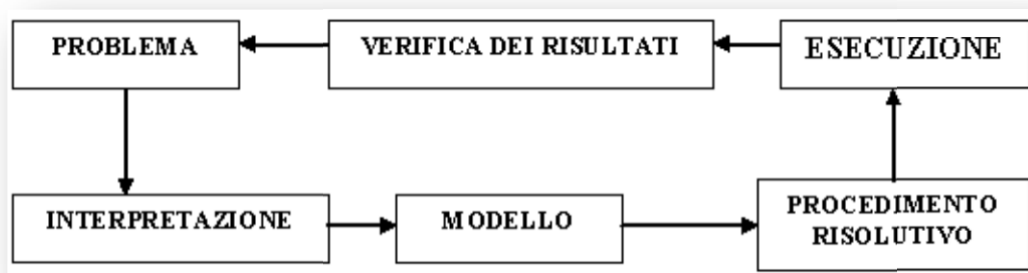
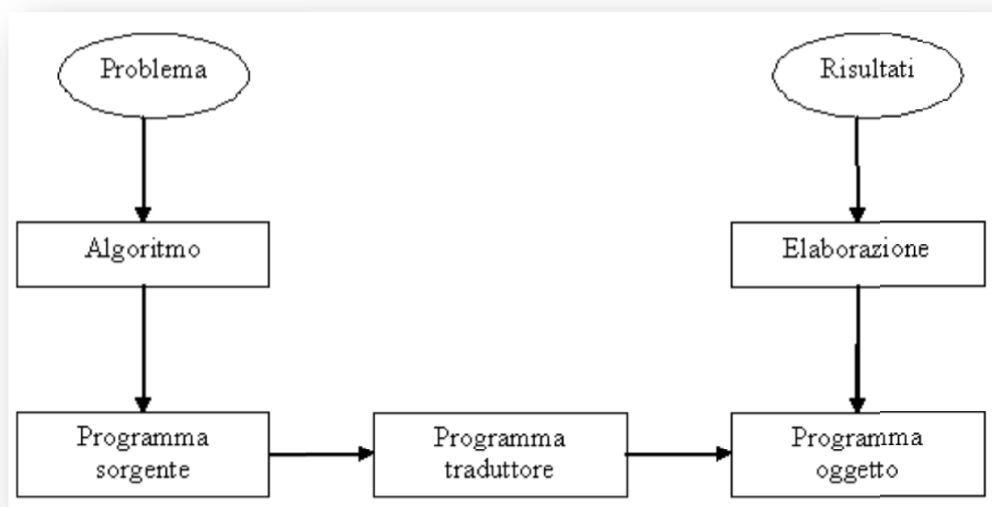


La soluzione è costituita dall'insieme degli stati intermedi. Il risultato è lo stato finale.

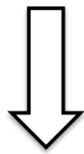
### *Differenza tra Algoritmo e Programma*

**ALGORITMO:** sequenza finita di passi necessari per risolvere un problema o eseguire una computazione. Solitamente esistono diversi modi per risolvere un problema e quindi diversi algoritmi.

**PROGRAMMA:** descrizione di un algoritmo scritto in un particolare linguaggio di programmazione.



**PROGRAMMATORE  
LINGUAGGIO SORGENTE**

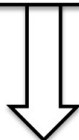


**Compilazione  
Traduce il codice sorgente  
nel codice macchina**

**COMPUTER  
CODICE MACCHINA**

**PROGRAMMATORE  
LINGUAGGIO SORGENTE**

**INTERPRETE**



**Interprete:  
Implementazione che  
esegue le istruzioni  
del linguaggio sorgente**

**COMPUTER  
CODICE MACCHINA**

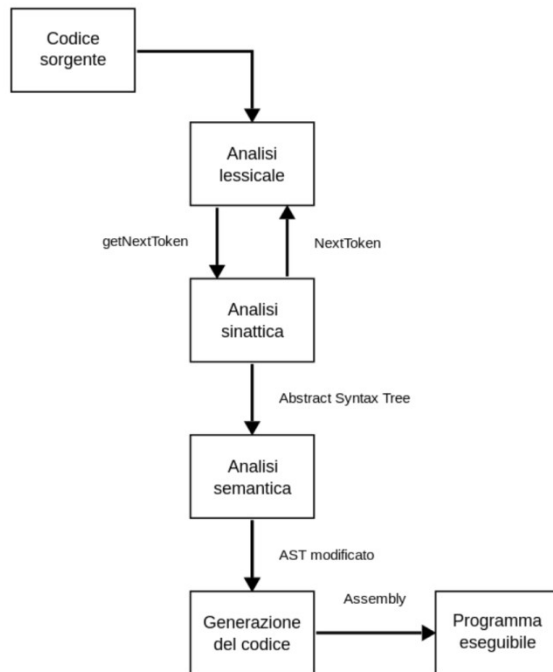
Un compilatore, in informatica, è un programma che traduce una serie di istruzioni scritte in un determinato linguaggio di programmazione (codice sorgente) in istruzioni di un altro linguaggio (codice oggetto). Questo processo di traduzione si chiama compilazione.

Il compilatore traduce l'intero codice sorgente in linguaggio macchina (.exe). Il compilatore durante l'esecuzione del programma exe non fa nulla, quindi può non risiedere in memoria.

L'interprete traduce una istruzione alla volta e la esegue, quindi durante l'esecuzione del programma deve essere in memoria.

# Fasi tipiche della compilazione

---



## ▶ **Analisi lessicale**

- ▶ Attraverso un analizzatore lessicale il compilatore divide il codice sorgente in tanti pezzetti chiamati token. I token sono gli elementi minimi (non ulteriormente divisibili) di un linguaggio, ad esempio parole chiave (for, while), nomi di variabili (pippo), operatori (+, -, «)

## ▶ **Analisi sintattica**

- ▶ L'analisi sintattica prende in ingresso la sequenza di token generata nella fase precedente ed esegue il controllo sintattico. Il controllo sintattico è effettuato attraverso una grammatica.

## ▶ **Analisi semantica**

- ▶ L'analisi semantica si occupa di controllare il significato delle istruzioni presenti nel codice in ingresso. Controlli tipici di questa fase sono il type checking, ovvero il controllo di tipo, controllare che gli identificatori siano stati dichiarati prima di essere usati e così via. Come supporto a questa fase viene creata una tabella dei simboli che contiene informazioni su tutti gli elementi simbolici incontrati quali nome, scope, tipo (se presente) etc.
-

## Elementi fondamentali della programmazione

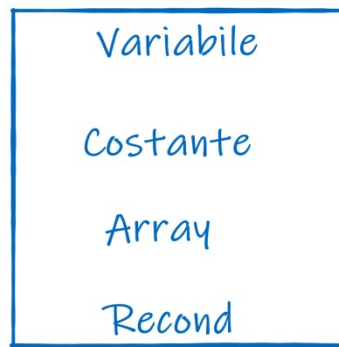
Strutture che controllano  
il flusso delle istruzioni

Sequenza

Selezione

Iterazione

Strutture dati

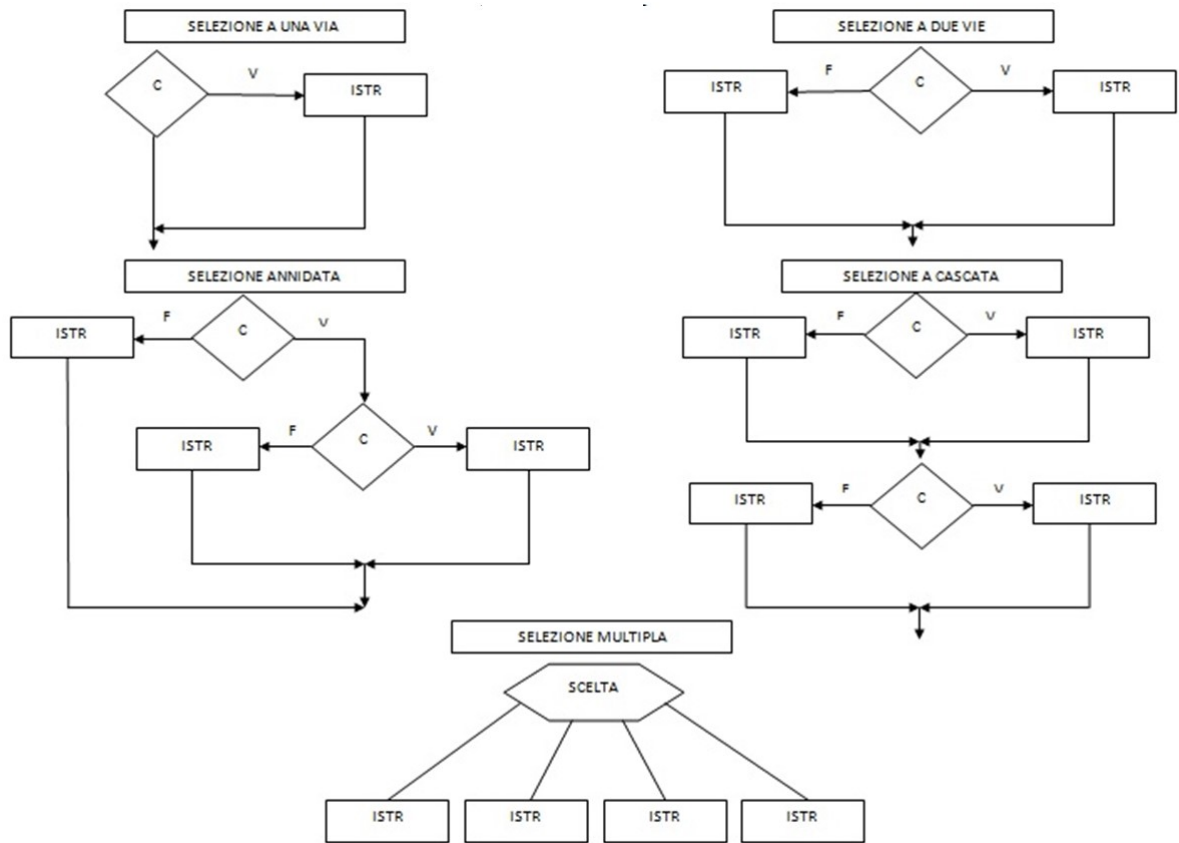


RAM



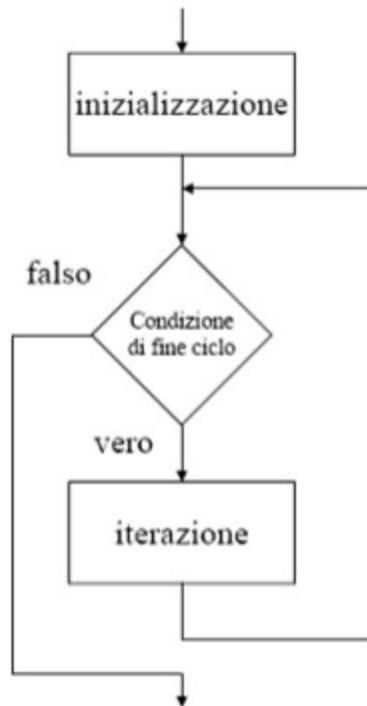
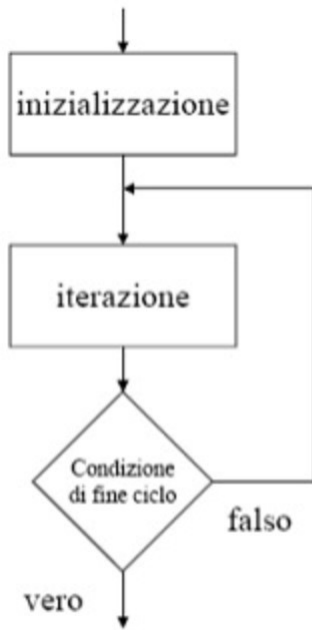
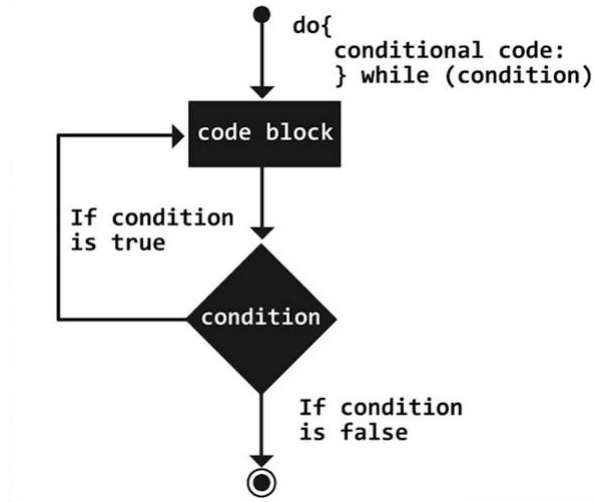
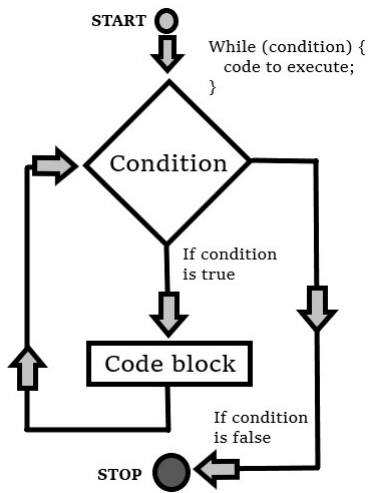
memoria di massa

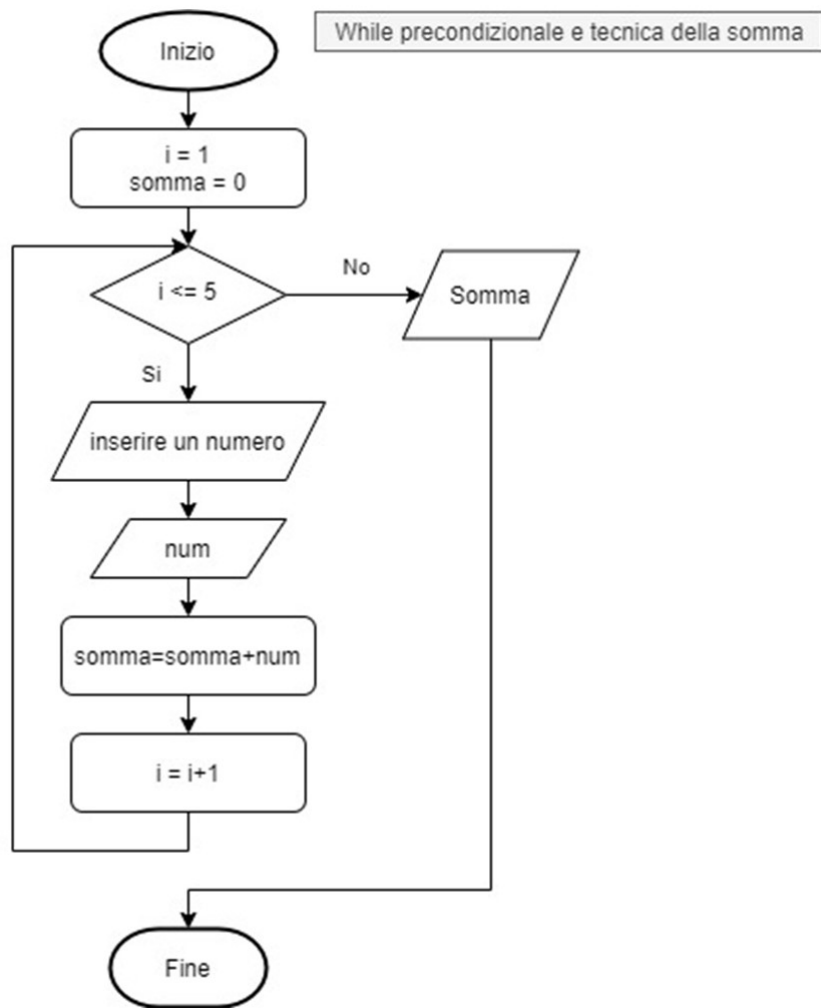
# SELEZIONE





# ITERAZIONE





```

int i=1; int somma=0;
while(i<=5)
{
  cout<<"inserire il " << i <<"^ numero: ";
  cin>>num;
  somma=somma+num;
  i++;
}
cout<<"la somma dei numeri inseriti e': "<<<somma;
  
```

## Von Neumann

---

- Il modello di **Von Neumann** è alla base della struttura dei computer attuali
- Due componenti principali
  - **Memoria**, dove sono memorizzati i programmi e i dati
  - **Unità centrale di elaborazione**, che ha il compito di eseguire i programmi immagazzinati in memoria prelevando le istruzioni (e i dati relativi), interpretandole ed eseguendole una dopo l'altra

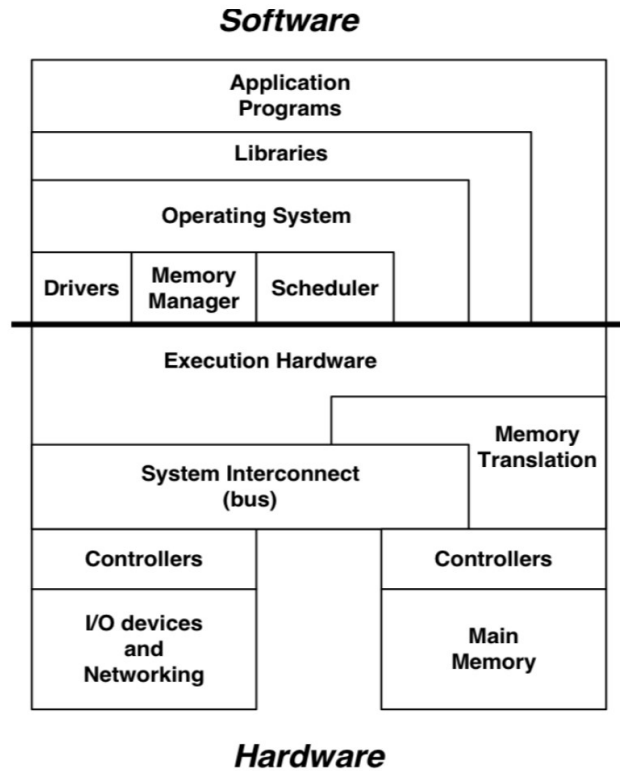
## Ciclo Fetch-Execute

---

- **Fetch**: L'istruzione da eseguire viene prelevata dalla memoria e trasferita all'interno della CPU
- **Decode**: L'istruzione viene interpretata e vengono avviate le azioni interne necessarie per la sua esecuzione
- **Data Fetch**: Sono prelevati dalla memoria i dati sui quali eseguire l'operazione prevista dalla istruzione
- **Execute**: È portata a termine l'esecuzione dell'operazione prevista dall'istruzione
- **Store**: È memorizzato il risultato dell'operazione prevista dall'istruzione

# Astrazione

- Sistemi: livelli di astrazione
  - “levels of abstraction with well defined interfaces”
- Meccanismo per controllare e dominare la complessità dei sistemi



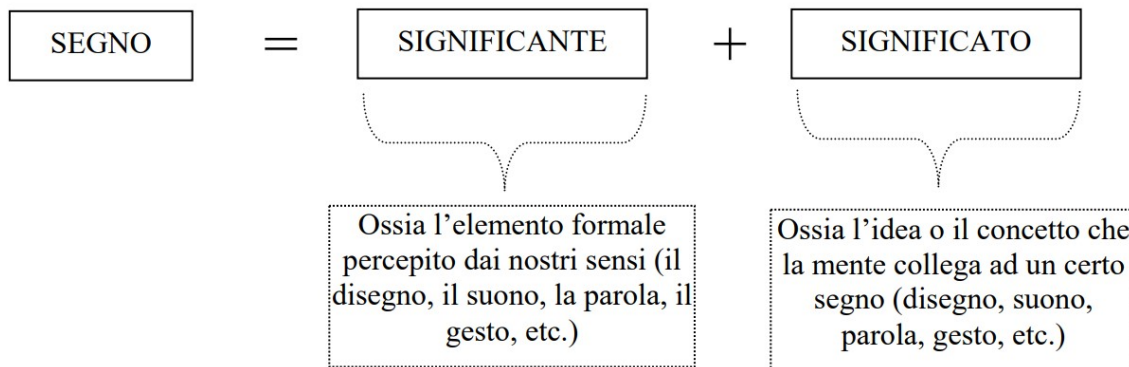
## LINGUAGGI NATURALI E LINGUAGGI FORMALI

Per linguaggio si intende un codice ossia un insieme di segni e di regole che rendono possibile la comunicazione tra coloro che lo usano.

E' possibile quindi avere:

- Codici verbali (ossia le lingue come l'italiano, l'inglese, etc.);
- Codici gestuali (ossia il linguaggio dei sordomuti, il linguaggio degli aviatori, etc.);
- Codici musicali
- Codici iconici (ossia la segnaletica stradale, la barra delle applicazioni del computer)

Il segno è il mezzo attraverso il quale si trasmettono particolari messaggi (sentimenti, pensieri, emozioni, informazioni, etc.)



Quando parliamo di segni artificiali ci riferiamo alla convenzione che nasce all'interno di una comunità di persone per stabilire il nesso o collegamento tra significante e significato; tale convenzione potrebbe essere sempre messa in discussione. (Es. il fischiare così come inteso da noi o dagli americani)

Quando parliamo di segni naturali ci riferiamo a quelli in cui il collegamento tra significante e significato; è direttamente collegato ad un rapporto innato tra causa ed effetto. (Es. il fumo è il segno del fuoco, poiché il fuoco è la causa del fumo) Tra tutti le tipologie dei linguaggi possibili distinguiamo:

**LINGUAGGI VERBALI** ossia quelli usati dall'uomo per comunicare quotidianamente.

I segni sono rappresentati dalle parole (segni artificiali). Il loro significante varia se ci riferiamo al linguaggio scritto (ossia le lettere dell'alfabeto) oppure al linguaggio orale (ossia i suoni - fonemi - emessi da chi parla per formare le parole). Il loro significato è attribuito per convenzione.

**LINGUAGGI NON VERBALI** ossia quelli che non usano la parola come mezzo di comunicazione e possono essere di 5 tipi:

- linguaggi basati sulla vista (dai geroglifici egizi, ai cartelli stradali);
- linguaggi basati sul tatto (es. braille); - linguaggi basati sull'udito;
- linguaggi basati sull'olfatto;
- linguaggi basati sul gusto;

I linguaggi verbali, scritti ed orali, sono anche detti linguaggi naturali (italiano, tedesco, inglese, francese, etc.).

Sono caratterizzati da un proprio alfabeto ossia l'insieme dei simboli (i caratteri alfanumerici) che ognuno di

essi possiede per formare i segni.

Mettendo insieme i simboli otteniamo le parole.

L'insieme di tutte le parole consentite in un linguaggio prende il nome di vocabolario o lessico.

La grammatica di un linguaggio comprende l'alfabeto e l'insieme delle regole per formare le parole e raggruppare le frasi.

La sintassi è un insieme di regole che definisce la scrittura di frasi formalmente corrette.

La semantica definisce il significato delle parole e delle frasi.

Una delle principali caratteristiche dei linguaggi naturali è la presenza di sinonimi, ambiguità e talvolta eccezioni. Inoltre essi si "evolvono" aggiungendo nuove parole o significati a quelle già esistenti. Es. Volenteroso può essere sostituito dal punto di vista semantico con zelante, operoso, attivo, dinamico. Pesante può avere significato diversi a seconda del contesto della frase usata (si può dire "pesante" di un film ma anche di una pietra). Se dico: "Ieri ho visto Giuseppe con un conoscente" vuol significare che Giuseppe era in compagnia di un conoscente oppure io stesso ero in compagnia di un conoscente? Inoltre in entrambi i casi il conoscente era di Giuseppe oppure il mio?

Nei linguaggi naturali ad ogni simbolo non corrisponde un unico significato. Un simbolo può trasmettere molteplici significati in un singolo contesto.

Appare evidente che nel comunicare istruzioni ad un esecutore automatico ossia nella comunicazione uomo-macchina non vi possano essere ambiguità, eccezioni, incertezze, interpretazioni e devono essere controllati da regole prefissate.

Quindi ecco, in opposizione ai linguaggi naturali, l'introduzione dei linguaggi formali (formale sta per rigorosamente definito) ossia di quei linguaggi usati dall'uomo che utilizzano simboli astratti univocamente interpretabili applicati a situazioni concrete. Nei linguaggi formali accade che: Ad ogni simbolo corrisponde uno ed un solo significato;

- Un simbolo trasmette sempre lo stesso significato qualsiasi sia il contesto di utilizzo;
- Una volta definito, il codice non può essere modificato dagli elementi che lo utilizzano.
- I linguaggi di programmazione ossia quei linguaggi in grado di tradurre gli algoritmi in istruzioni comprensibili dall'elaboratore sono ovviamente esempi di linguaggi formali.

Quindi con il termine programma si intende la trasformazione dell'algoritmo che risolve una classe di problemi in una sequenza di frasi sintatticamente e semanticamente corrette.

Tali frasi sono formate utilizzando la grammatica (ossia l'alfabeto, il vocabolario o lessico, e le regole) del linguaggio di programmazione scelto. Le singole frasi consentite nel linguaggio di programmazione scelto prendono il nome di istruzioni