DISPENSE DI PROGRAMMAZIONE

Modulo 2 – Dall'algoritmo al programma

- **Delega della esecuzione di un procedimento di calcolo: il ruolo del linguaggio.**
- **Linguaggi a basso e ad alto livello.**
- **Linguaggi imperativi. Cenni sulla loro evoluzione.**
- **Linguaggi e grammatiche. Forma di Backus-Naur (BNF); Diagrammi sintattici.**
- **Cenni sui traduttori**

Ha collaborato alla editazione Tosini Serena

UN PROGRAMMA È UN MESSAGGIO DI COMUNICAZIONE TRA L'UOMO E LA MACCHINA.

Per una COMUNICAZIONE DIRETTA tra due interlocutori bisogna assicurarsi che l'estensore del messaggio, al momento della formulazione dello stesso, e il ricevitore, al momento della ricezione, diano al messaggio EGUAL SIGNIFICATO.

Altrimenti

Si deve ricorrere ad una COMUNICAZIONE INTERMEDIATA da un TRADUTTORE

QUANDO SI RICORRE AD UN TRADUTTORE?

- **SE ESTENSORE E RICEVITORE USANO (CONOSCONO) LINGUAGGI DIFFERENTI;**

- **SE ESTENSORE E RICEVITORE HANNO UN DIVERSO GRADO DI CONOSCENZA DEL LINGUAGGIO USATO PER COMUNICARE;**

- **SE MANCANO ADEGUATE CONVENZIONI TRA I DUE PER UNA INTERPRETAZIONE UNIVOCA DEL MESSAGGIO.**

Basso livello

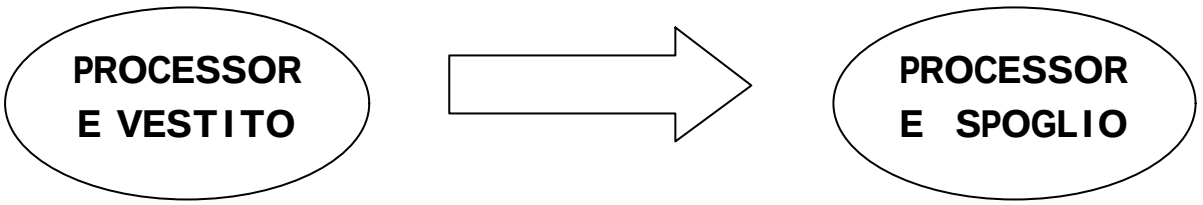
LINGUAGGIO MACCHINA	**DIRETTAMENTE ESEGUIBILE**
LINGUAGGIO ASSEMBLATIVO	**OCCORRE L'ASSEMBLATORE**
LINGUAGGIO DI PROGRAMMAZIONE	**OCCORRE INTERPRETE oppure COMPILATORE**

Alto livello

LINGUAGGIO NATURALE

- **LA MACCHINA COMPRENDE IL LINGUAGGIO MACCHINA, COSTITUITO DA UN INSIEME DI ISTRUZIONI ELEMENTARI. È DEFINIBILE COME PROCESSORE SPOGLIO.**

- **PER FACILITARE IL COMPITO DI PROGRAMMARE LA SOLUZIONE DI UN PROBLEMA,SI UTILIZZANO LINGUAGGI A PIÙ ALTO LIVELLO. UNA MACCHINA IN GRADO DI COMPRENDERE UN LINGUAGGIO L AD ALTO LIVELLO È UN PROCESSORE VESTITO.**



Linguaggio di programmazione **TRADUZIONE** **Linguaggio macchina**

COMPITO DEL PROGRAMMA TRADUTTORE:

EFFETTUARE LA TRADUZIONE

DA LINGUAGGIO L

IN LINGUAGGIO MACCHINA.

Io ho andato

E' ERRATA SINTATTICAMENTE

Il tavolo sta ballando

E' CORRETTA SINTATTICAMENTE (FORMA), MA E' ERRATA SEMANTICAMENTE (SIGNIFICATO)

LE REGOLE SINTATTICHE DEI LINGUAGGI DI PROGRAMMAZIONE

- **NON SONO MOLTO NUMEROSE**
- **SONO ABBASTANZA SEMPLICI**
- **VANNO RISPETTATE IN MODO RIGIDO**

IL PROGRAMMA TRADUTTORE CONTROLLA LA CORRETTEZZA SINTATTICA.

AD OGNI ISTRUZIONE (FRASE DEL LINGUAGGIO DI PROGRAMMAZIONE) DEVE CORRISPONDERE UNA UNICA INTERPRETAZIONE.

I TRADUTTORI OPERANO :

- **SULLE ISTRUZIONI**
- **SULLE STRUTTURE DI DATI**

LE ISTRUZIONI SONO MOLTO PIÙ POTENTI CHE NEL LINGUAGGIO MACCHINA E LE STRUTTURE DI DATI (LISTE, SEQUENZE, ALBERI, etc.) NON SONO DIRETTAMENTE DISPONIBILI A LIVELLO MACCHINA MA DEVONO ESSERE RAPPRESENTATE IN TERMINI DI BIT, INDIRIZZI E LEGAMI TRA LOCAZIONI.

I <u>TRADUTTORI</u> convertono i programmi espressi in linguaggio ad alto livello in programmi espressi in linguaggio macchina.

Le istruzioni macchina corrispondono ad operazioni elementari che "VEDONO" direttamente gli elementi costitutivi della macchina, quali i registri o gli indirizzi di memoria.

BISOGNA DOTARSI DEL TRADUTTORE SPECIFICO DEL LINGUAGGIO DI PROGRAMMAZIONE CHE SI VUOLE ADOPERARE.

I traduttori si distinguono in:

Compilatori:

producono (se durante l'analisi sintattica non sono stati rilevati errori) la traduzione completa del programma (SORGENTE o SOURCE) in linguaggio macchina (OGGETTO o OBJECT); questa viene memorizzata sul disco fisso in modo da poter essere successivamente eseguita.

Traduzione ed esecuzione avvengono in differita.

Interpreti:

eseguono la traduzione in progressione: ogni istruzione in linguaggio ad alto livello viene analizzata e tradotta in linguaggio macchina e segue immediatamente la esecuzione da parte del processore, si prosegue con la istruzione successiva fino alla fine del programma.

Traduzione ed esecuzione sono effettuate "simultaneamente".

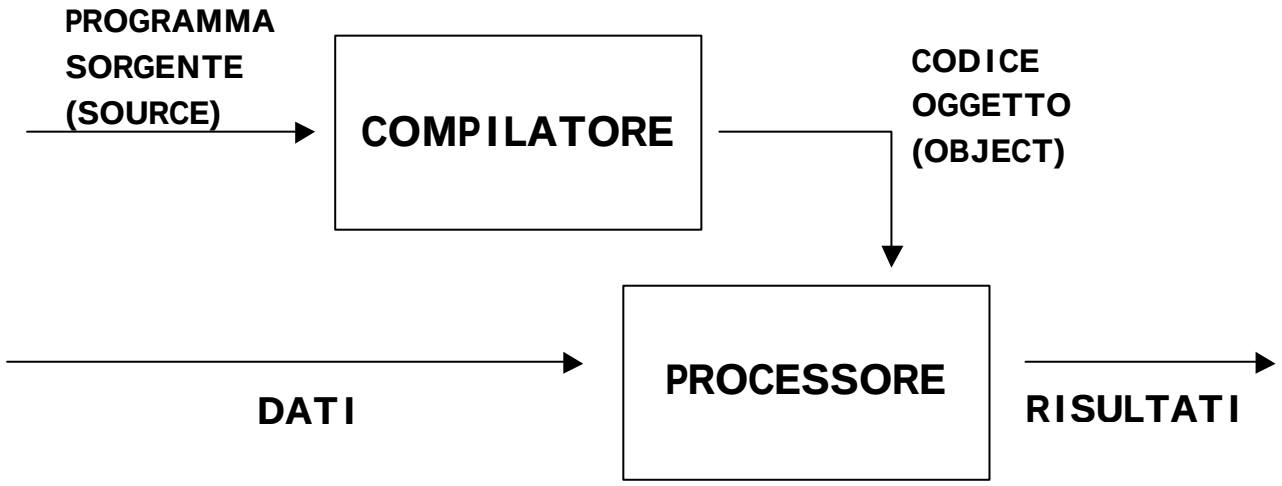
INTERPRETI E COMPILATORI

VANTAGGI

1) Semplicità
2) Interattività

SVANTAGGI

1) Inefficienza

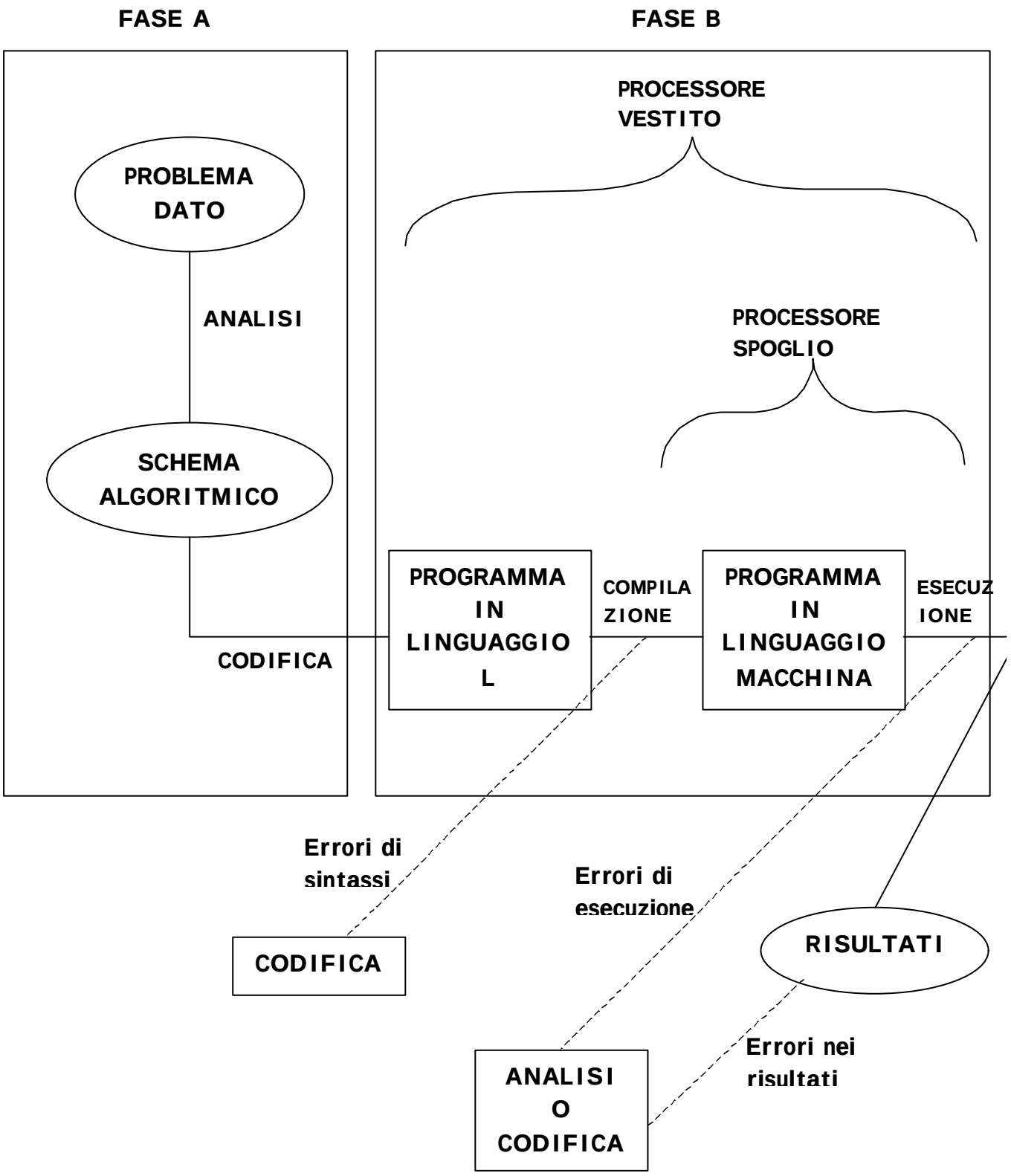


VANTAGGI

1) Efficienza
2) Possibile ottimizzazione del codice oggetto

SVANTAGGI

1) Nessuna interattività


FASE A
FASE B
PROCESSORE VESTITO
PROBLEMA DATO
ANALISI
PROCESSORE SPOGLIO
SCHEMA ALGORITMICO
PROGRAMMA IN LINGUAGGIO L
COMPILAZIONE
PROGRAMMA IN LINGUAGGIO MACCHINA
ESECUZIONE
CODIFICA
Errori di sintassi
Errori di esecuzione
CODIFICA
RISULTATI
ANALISI O CODIFICA
Errori nei risultati

UN SISTEMA DI CALCOLO PUÒ ESSERE VISTO COME UNA GERARCHIA DI COMPONENTI SOFTWARE E HARDWARE

SW APPLICATIVO (es. pacchetti ad usi speciali)
SW DI SISTEMA (sistemi operativi, traduttori, etc.)
HW (CPU, Memorie, dispositivi di I/O, ecc.)

IL SW APPLICATIVO INDICA UN INSIEME DI PROGRAMMI DA UTILIZZARE PER APPLICAZIONI PARTICOLARI. SONO GENERALMENTE SCRITTI IN LINGUAGGIO AD ALTO LIVELLO, FORNITI DIRETTAMENTE DA COSTRUTTORI DI COMPUTER O DITTE SPECIALIZZATE NELLA PRODUZIONE DI PROGRAMMI.

IL SW DI SISTEMA È L'INSIEME DEI PROGRAMMI CHE FORNISCONO SERVIZI E SVOLGONO FUNZIONI VITALI PER IL SW APPLICATIVO.

I TRADUTTORI FANNO PARTE DEL SOFTWARE DI SISTEMA, LIVELLO INTERMEDIO DELLA GERARCHIA SW-HW

I LINGUAGGI POSSONO ANCHE DISTINGUERSI IN:

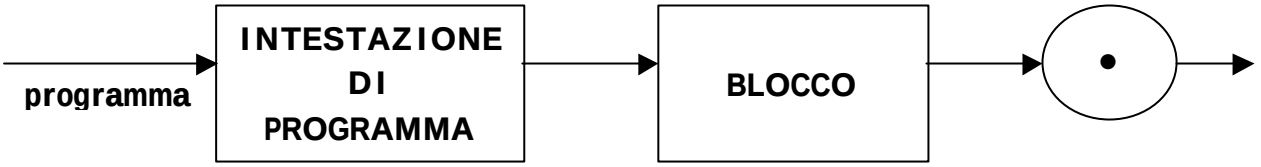
- **PROCEDURALI**

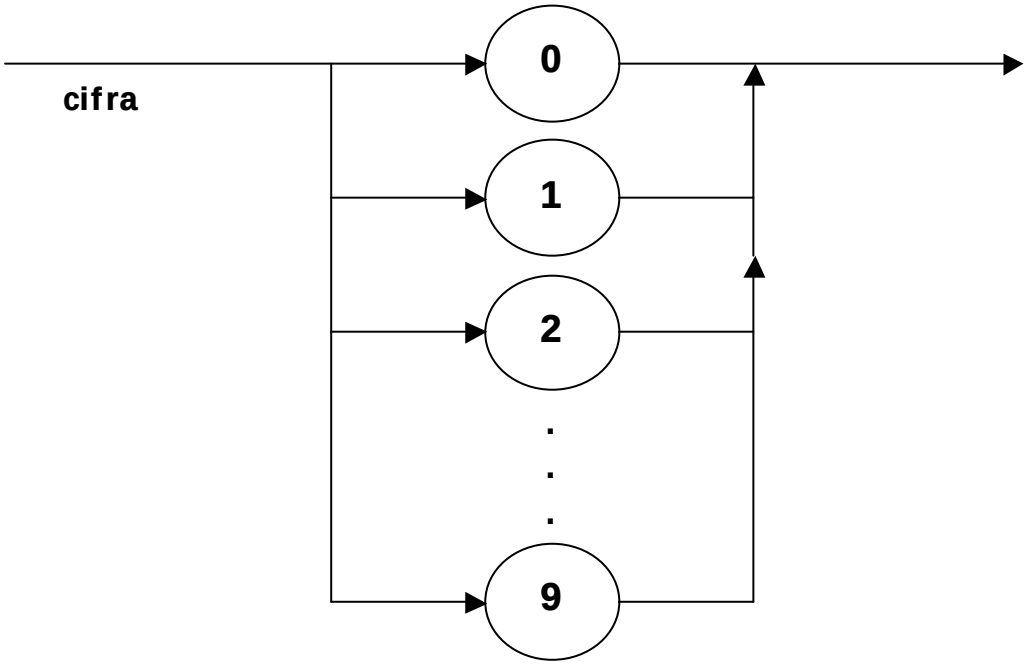
CHE DESCRIVONO IL PROCEDIMENTO DI SOLUZIONE SPECIFICANDO LA SUCCESSIONE TEMPORALE DELLE OPERAZIONI DA SVOLGERE

- **NON PROCEDURALI**

ALCUNI ESPRIMONO LE PROPRIETÀ DEI RISULTATI DA OTTENERE.
Es: trovare la radice quadrata di un valore y con l'istruzione:
" *trovare x tale che x*x sia uguale ad y*"

PER DESCRIVERE LA SINTASSI DI UN LINGUAGGIO SI POSSONO USARE LE CARTE o DIAGRAMMI SINTATTICI





Gli *elementi terminali* sono racchiusi in CERCHI o OVALI;
sono usate FRECCE ORIENTATE come *connettori*;
le *categorie sintattiche* sono racchiuse in RETTANGOLI.

ALTRO FORMALISMO ADEGUATO È LA BACKUS-NAUR FORM (BNF).

< programma > ::= < intestazione di programma > < blocco >.

< intestazione di programma > ::= PROGRAM < identificatore > (< parametri di programma>);

LINGUAGGI PROCEDURALI o IMPERATIVI
(basati sull'assegnamento)

I CONCETTI FONDAMENTALI SONO:

- **VARIABILE**
-
- **ASSEGNAMENTO**

ALCUNI LINGUAGGI IMPERATIVI (STRUTTURATI e NON)

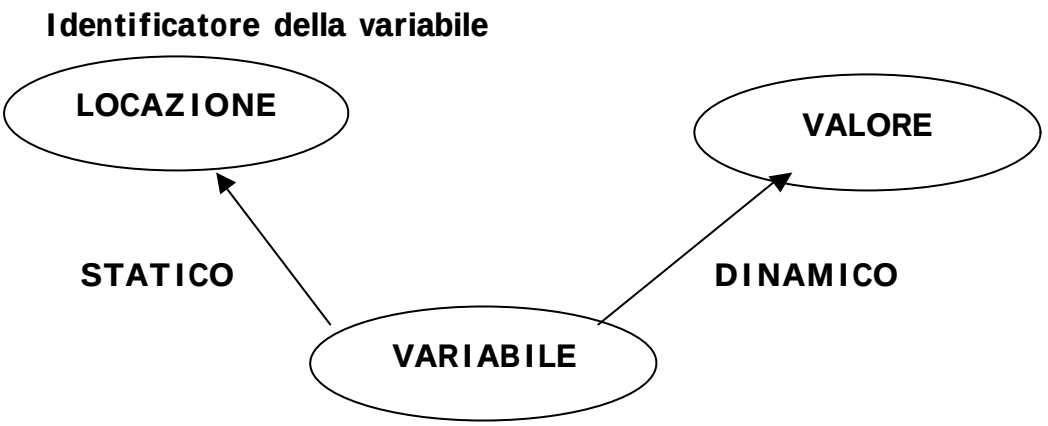
BASIC
FORTRAN
FORTRAN IV
STRUFORT
PL/1
MODULA 2
PASCAL

CONCETTO DI VARIABILE

PER DESCRIVERE I PROCESSI, QUALUNQUE LINGUAGGIO DI DESCRIZIONE SI ADOTTI, È NECESSARIO FAR RIFERIMENTO AGLI OGGETTI IMPLICATI NEI PROCESSI TRAMITE NOMI SIMBOLICI.

AD OGNI ESECUZIONE DI UN PROCESSO SI USANO I VALORI ASSOCIATI AI NOMI SIMBOLICI CHE COMPAIONO NELL'ALGORITMO CHE LO DESCRIVE

Es.
RICETTA = descrive le azioni sugli ingredienti
ESECUZIONE di una ricetta = adopero gli ingredienti reali, che saranno trasformati azione dopo azione.



UNA VARIABILE DENOTA UNA COPPIA COSTITUITA DA
UNA LOCAZIONE
E
DAL VALORE CONTENUTO IN ESSA

IL LEGAME tra variabile ed indirizzo È STATICO
MENTRE
IL LEGAME tra variabile e valore È DINAMICO

Esempio

1) X ← 10

2) X ← X + 1

prima di 1) quanto vale X ? ?

dopo 1) X varrà 10

dopo 2) X varrà 11

IN UN PROGRAMMA (traduzione equivalente dell'algoritmo in un linguaggio di programmazione) UNA VARIABILE È CARATTERIZZATA DA:

- **UN IDENTIFICATORE (nome formato in base alle regole sintattiche del linguaggio di programmazione usato)**
- **UN VALORE**
- **UN TIPO (è un attributo che specifica l'insieme dei possibili valori della variabile e degli operatori applicabili)**

E RAPPRESENTA UNA LOCAZIONE DI MEMORIA, CONTRADDISTINTA DA UNO SPECIFICO INDIRIZZO, CONTENENTE IL VALORE SU CUI APPLICARE LE ISTRUZIONI DEL PROGRAMMA.

UNA VARIABILE ASSUME VALORE QUANDO NELLA LOCAZIONE AD ESSA RISERVATA VIENE MEMORIZZATO IL VALORE.

CIÒ PUÒ AVVENIRE TRAMITE:

- **ISTRUZIONI DI INGRESSO**
- **ISTRUZIONI DI ASSEGNAMENTO**

POICHÈ NEL PROGRAMMA SI FA RIFERIMENTO ALLE VARIABILI, CIOÈ AGLI INDIRIZZI DELLE AREE DI MEMORIA IN CUI SONO CONSERVATI I DATI,

UN LINGUAGGIO DI PROGRAMMAZIONE DEVE DISPORRE ANCHE DI ISTRUZIONI CHE:

- **PREDISPONGONO LE POSIZIONI DI MEMORIA DA UTILIZZARE**
- **ASSOCIANO UN NOME A CIASCUNA DI ESSE**
- **DEFINISCONO IL TIPO DI DATI CHE POSSONO ESSERE MEMORIZZATI (INSIEME DI VALORI PERMESSI PER UNA VARIABILE)**

SOLITAMENTE VIENE ADOTTATA LA SEGUENTE CONVENZIONE LINGUISTICA:

TUTTE LE VARIABILI DEVONO ESSERE INDICATE IN UNA OPPORTUNA ISTRUZIONE CHE FORNISCE UNA LISTA CONTENENTE I NOMI SCELTI PER LE VARIABILI E I TIPI CORRISPONDENTI (ISTRUZIONI DICHIARATIVE)

TIPI DI ISTRUZIONI

- **Istruzioni di ASSEGNAMENTO**
- **Istruzioni di INGRESSO**
- **Istruzioni di USCITA**
- **Istruzioni di CONTROLLO e STRUTTURE DI CONTROLLO (sequenze, selezioni, iterazioni)**

+

- **Istruzioni DICHIARATIVE (tutti i linguaggi di programmazione dispongono di convenzioni per specificare nome e tipo per ciascuna variabile)**

ISTRUZIONE DI ASSEGNAMENTO

Una istruzione di assegnamento determina:

- **La produzione di un <u>NUOVO VALORE</u>**
- **L'assegnamento di quel valore alla <u>VARIABILE</u>**

L'espressione (aritmetica o logica) viene valutata e il VALORE ottenuto viene MEMORIZZATO nella LOCAZIONE di memoria riservata alla VARIABILE da assegnare (che compare sempre alla sinistra del simbolo di assegnamento, che potrà differire dal linguaggio a linguaggio)

La variabile che appare alla SINISTRA del simbolo di assegnamento assume il valore di ciò che appare alla destra.

x ← y è diverso da y ← x

ISTRUZIONI DI INGRESSO

Una istruzione di ingresso attiva una operazione di LETTURA che si configura come l'ASSEGNAZIONE del VALORE letto da un supporto di MEMORIZZAZIONE ESTERNO (schede, nastri e dischi magnetici, etc.) ad un'AREA DI MEMORIA INTERNA individuata dal NOME della variabile che compare nella istruzione di ingresso.

leggi x **read (x)**

prima della lettura: x vale ?

immissione dall'ESTERNO di un valore

dopo la lettura : x vale ciò che è stato immesso

ISTRUZIONI DI USCITA

Una istruzione di uscita attiva una operazione di SCRITTURA che si configura come la "COPIATURA" su un SUPPORTO ESTERNO (carta, nastri e dischi magnetici, schermo, ecc.) del CONTENUTO di una AREA DI MEMORIA INTERNA denotata dal NOME della variabile che compare nella istruzione di scrittura.

scrivi y write (y)

Prima della scrittura y 92

Dopo la scrittura y 92

Stampa visualizzata su SUPPORTO ad es. CARTACEO

LINGUAGGI DI PROGRAMMAZIONE

ASTRAZIONE (a vari livelli)

ELABORATORI CONVENZIONALI

VARIABILE: UN NOME PER RIFERIRSI AD UNA LOCAZIONE DI MEMORIA

ASSEGNAMENTO: MODIFICA DEL VALORE DELLE LOCAZIONI DI MEMORIA