



# Informatica

Capitolo 1 – Introduzione

Prof. Ivan Gentile

# Tipi di Linguaggi

- **Linguaggio macchina o linguaggio a basso livello**
  - Compreso dalla macchina
  - Ogni famiglia di processori ha il suo linguaggio macchina
  - Elementare
  - Difficile da programmare
- **Linguaggio ad Alto livello (C, C++, Java, Basic, Pascal, ...)**
  - Più vicini al nostro modo di ragionare
- **Traduttore**
  - Da alto livello a basso livello

# Chi interagisce col computer?

- **utenti:** quelli adoperano i software fatti da altri.
- **programmatori:** coloro che scrivono il software per gli utenti
- Dobbiamo essere programmatori
  - Utenti già lo siamo

# Parti di un programma

1. Chiedere i dati all'utente
  2. Fornire le istruzioni di calcolo al processore.
  3. Restituire i risultati all'utente
- Esempio: Calcola l'area del cerchio
    1. Dammi il raggio  $r$
    2. Calcola l'area  $A$  del cerchio come il prodotto di  $r$  per  $r$  per costante  $PI = 3.14$
    3. L'area del cerchio è  $A$

# Dati (1/2)

- In un programma sono coinvolti tanti **dati**
- Classificazione in base alla *modificabilità*
  - **Costanti** (PI): non variano *mai* (anche tra un'esecuzione e un'altra)
  - **Variabili** (r, A): *possono variare* (all'interno di una stessa esecuzione o tra esecuzioni diverse)
- Classificazione in base *all'utilizzo*
  - **Input** (r): forniti *al* programma
  - **Output** (A): forniti *dal* programma
  - **Interni** (PI): solo per elaborazioni del programma ma non interessa all'utilizzatore

## Dati (2/2)

- Una costante *deve essere* un dato interno
- Un dato di input o di output saranno *sempre* variabili
- Un dato può essere di input e di output

# Identificatori

- Un dato deve avere un nome univoco nel programma: **identificatore**
  - deve iniziare con una lettera,
  - non spazi
  - talvolta non sono permessi caratteri speciali come %& ,
  - in alcuni casi (C, Java) sono case sensitive in altri no (PASCAL, BASIC).
  - notazione **camel case**
    - Maiuscola come separatore di parole al posto dello spazio
    - valoreMassimo al posto di valore\_Massimo.
    - **Lower camel case**: iniziale minuscola valoreMassimo (useremo questa)
    - **Upper camel case**: iniziale maiuscola ValoreMassimo
  - notazione **snake**: parole separate da underscore (\_): valore\_massimo

# Tipo

- **Tipo:** indica l'insieme di appartenenza del dato (esempio intero, reale, carattere) e quali operazioni si possono fare su di esso.
  - Intero o reale: addizione, sottrazione, moltiplicazione, divisione, etc.
  - Carattere: confronto, concatenazione, etc.



# Perché identificatore e tipo

- In base al tipo il traduttore chiederà al computer di riservare una zona di memoria centrale sufficiente a contenere i possibili valori di quel dato
- L'identificatore è il nome di quella zona di memoria
- Tipo e modificabilità di un dato (costante e variabile) servono anche per aiutare il programmatore a non commettere errori
  - Se dico che un dato è costante e lo modifico il traduttore mi segnala errore
  - Se dico che un dato è carattere e lo moltiplico per un numero il traduttore mi segnala errore

# Operazioni di base: aritmetiche

- Sui numeri (interi e reali) si possono fare le classiche operazioni **aritmetiche**
  - + somma
  - - sottrazione
  - \* moltiplicazione
  - / divisione
    - Se i due operandi sono interi la divisione sarà **intera** (viene restituita solo la parte intera)
      - $3/2$  dà come risultato 1
    - Se almeno uno dei due è reale la divisione sarà quella decimale
      - $3.0/2$  dà come risultato 1.5

# Operazioni di base: assegnazione

- Il risultato di un calcolo spesso va *memorizzato* in una variabile
  - 2 moltiplicato per 3.14 per il raggio (r) lo voglio memorizzare in una variabile di nome c
  - Operazione di **assegnazione**
  - Spesso indicata con = (simbolo di uguale)
  - $c = 2 * 3.14 * r$
- **Non si tratta di un'uguaglianza!**
  - Si calcola quello che c'è a destra del simbolo e si *copia* nella variabile a sinistra
  - Infatti il Pascal usava il simbolo di := (due punti uguale)