

IMPLEMENTARE

Il modello di un sistema, che descrive una situazione, dovrà avere alcune caratteristiche per essere “*implementato*” (i.e. portato) sul calcolatore:

le sue variabili devono essere *registrabili* su memoria
le variabili devono poter essere messe in *relazione* tra loro

Dovrò formalizzare il problema in modo “*conveniente*”, così da poterlo implementare sul calcolatore.

La Formalizzazione

All'interno del problema in questione:

individuare le *variabili* in gioco

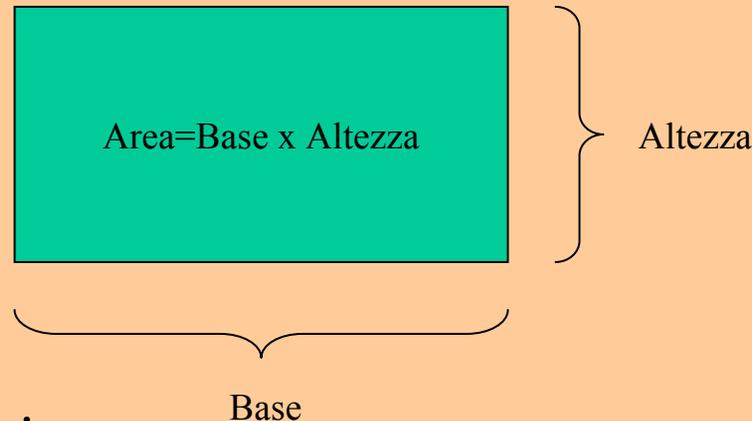
capire i *valori* da fargli assumere in funzione dello stato

formalizzare le *relazioni*

L'area del rettangolo

Il problema si pone con:

2 *variabili*: BASE (b) , ALTEZZA (h) ed AREA (a)
i *valori* dovranno essere forniti dall'utente (*input*)
la *relazione*: $a = b \times h$ (*output*)



NB: non ho costanti

L'area e la circonf. del cerchio

Il problema si pone con:

1 *variabile*: RAGGIO (r), AREA (a), CIRC (c)

1 *costante*: $\text{PIG} = 3.14$

il *valore* dovrà essere fornito dall'utente (*input*)

le *relazioni*: $a = \text{PIG} \times r^2$ (*output1*)

$c = 2 \text{PIG} \times r$ (*output2*)



Operazioni

Ho individuato i seguenti **DATI**:

1. le *variabili*
2. le *costanti*
3. assegnato “*nomi*” alle variabili e costanti (identificatori)

Ho scritto le *relazioni* usando i nomi che sono **AZIONI** che in genere sono di tipo ARITMETICO o LOGICO

Metodo

La metodologia di lavoro riguarderà poi la:

1. definizione del “*punto*” di partenza (start)
2. definizione della *sequenza delle operazioni da svolgere*
3. esatta definizione della *modalità* per svolgere ogni singola operazione
4. definizione del “*termine*” (stop)

Pensiamo ad una ricetta da cucina.

ALGORITMO

Per **ALGORITMO** (da Al-Khuwarizmi, algebrista arabo del IX° secolo) intendiamo:

*la definizione di un insieme finito di **istruzioni**, che devono essere eseguite per portare a termine un dato compito e per raggiungere un risultato definito in precedenza*

NB: La definizione implica la presenza di un esecutore

LE ISTRUZIONI DELL'ALGORITMO

Le istruzioni devono:

1. essere *realizzabili* dall'esecutore
2. essere *precise* e non ambigue
3. avere durata *limitata*
4. Avere un risultato *osservabile*
5. essere di tipo *deterministico*
6. essere *elementari*

CARATTERISTICHE DELL'ALGORITMO

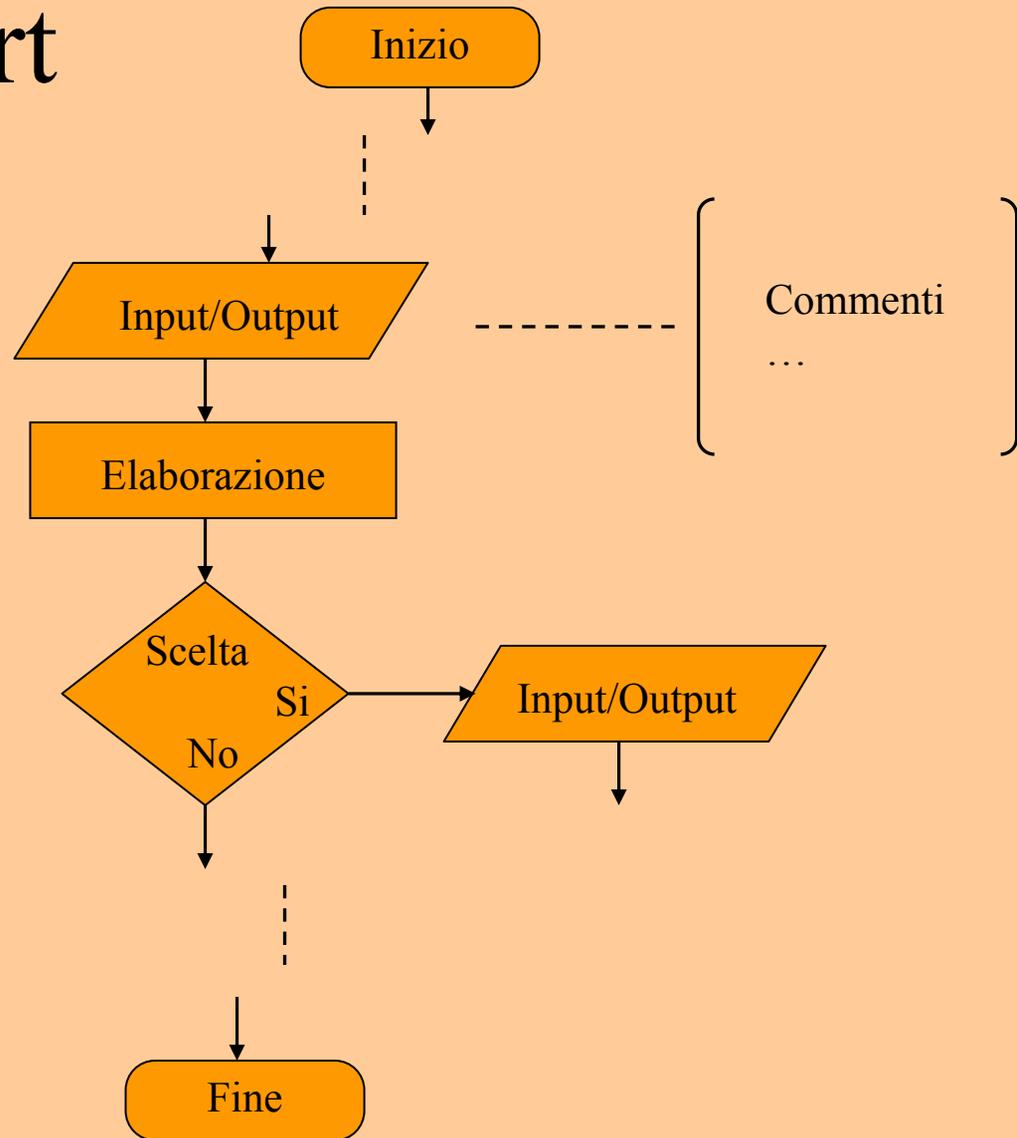
L'algoritmo dovrà:

1. avere un *numero finito* di istruzioni dall'esecutore
2. essere *completo*
3. essere *riproducibile*

Che linguaggio utilizzo per comunicarlo all'esecutore?

in prima istanza un linguaggio **pseudo-naturale**, sintetico ed efficace, in secondo luogo “**grafico**” per me, per i miei collaboratori, per l'esecutore vedremo...

Il Flow-chart



La PSEUDO-codifica

Potrebbe essere qualcosa del tipo:

1. INIZIO
2. LEGGI b
3. LEGGI h
4. Calcola $c = 2 * \text{PIG} * r$
5. ...
6. Calcola $a = \text{PIG} * \text{quadrato}(r)$
7. Stampa c, a
8. FINE

IMPLEMENTARE su calcolatore

L'algoritmo scritto in PSEUDO-codice o Flow-chart dovrà essere
“implementato” per l'esecutore:

dovremo usare un *linguaggio appropriato!*

La PSEUDO-codifica

Ci ha suggerito qualcosa del tipo:

1. INIZIO
2. LEGGI b
3. LEGGI h
4. Calcola $c = 2 * \text{PIG} * r$
5. ...
6. Calcola $a = \text{PIG} * \text{quadrato}(r)$
7. Stampa c, a
8. FINE

Un primo esempio pascal

```
PROGRAM cerchio;
```

```
    CONST
```

```
        PIG = 3.14;
```

```
    VAR
```

```
        r,c,a: REAL;
```

```
BEGIN
```

```
    Write('inserire il raggio:');
```

```
    Readln(r);
```

```
    c := 2*PIG*r;
```

```
    a := PIG*Sqr(r);
```

```
    Writeln('la circonferenza vale:':30,c:8:2,'il raggio:',r:8:2);
```

```
END.
```