E' un formalismo che consente di rappresentare gli algoritmi mediante semplici istruzioni in linguaggio «parlato»

Delle volte viene chiamato *metalinguaggio*, e l'algoritmo scritto tramite metalinguaggio viene chiamato *pseudocodice*.

- Si può passare facilmente da diagramma di flusso a pseudocodice, e viceversa.
- In sostanza, uno pseudocodice appare come una sequenza di istruzioni scritte in linguaggio parlato che specificano le operazioni da svolgere per poter eseguire l'algoritmo

Pseudocodice

```
var a
leggi a
var i := 1
ripeti finché (i<=10){
    a = a + i
    i = i + 1
}
stampa a
```

- Ogni istruzione va scritta su una riga
- Come nei diagrammi di flusso, può essere necessario introdurre delle variabili «temporanee» (ad esempio, la variabile i), necessarie alla risoluzione del problema.

Pseudocodice - Espressioni

- Sequenze di variabili e costanti combinate fra loro mediante operatori
- Gli operatori possono essere aritmetici, ad esempio come somma (+), sottrazione (-), prodotto (*), divisione (/), resto della divisione (%), radice quadrata (sqrt)
- Espressioni relazionali: danno come risultato vero o falso. Sono, ad esempio, minore (<), maggiore (>), uguale (= =), diverso (! =)
- Espressioni logiche: danno anch'esse come risultato vero o falso, e sono **and**, **or**, **not**
- Nella valutazione di una espressione, si sostituisce ad ogni variabile il suo valore attuale e si eseguono le operazioni secondo un ordine prestabilito da regole di precedenza (possono comparire parentesi).
- A tutte le variabili che compaiono nell'espressione deve essere stato associato un valore prima della valutazione dell'espressione

Creazione variabili

- Le variabili possono essere create tramite l'istruzione **var**

Esempio:

`var a` → crea una variabile di nome «a»

Ingresso (lettura, read, input)

- Come nel diagramma di flusso, si utilizza il comando **leggi**

QUINDI in sostanza, per poter leggere un numero da tastiera e salvarlo in un contenitore, vanno fatte in ordine le seguenti istruzioni:

1. `var a` → crea variabile di nome a
2. `leggi a` → leggi da tastiera e salva in a

Uscita (stampa, print, output)

- **Esecuzione dell'istruzione:**

Si calcolano i valori delle espressioni e si trasmettono all'unità di uscita (ad esempio, il video)

- Come nel diagramma di flusso, si utilizza il comando **scrivi**

QUINDI se vogliamo stampare a video il valore di una variabile, o il risultato di un'espressione, ci basta l'istruzione

- scrivi a
- scrivi $a+b+1$
- scrivi "ciao mondo! " → scrive esattamente quello che c'è scritto tra le virgolette
- scrivi 'a capo' → per andare a capo sul video

Assegnamento

- **Esecuzione dell'istruzione:**

Si calcola il valore dell'espressione a destra del simbolo «:=» e lo si salva (*assegna*) nella variabile indicata a sinistra del simbolo «:=»

$a := 5$

- «Calcola il valore dell'espressione (5, non c'è bisogno di fare nulla) ed assegnalo alla variabile a »

$i := i + 1$

- «Calcola il valore dell'espressione ($i + 1$) ed assegnalo alla variabile i »

- Esempio: se i vale 6 prima di eseguire l'istruzione, calcolo $6+1$ (7) e lo salvo nella variabile i ; quando l'istruzione è completata, la variabile i vale 7.

Istruzioni condizionali

- **Esecuzione dell'istruzione:**

Si valuta la condizione specificata all'interno delle parentesi tonde: se è verificata, si prosegue le istruzioni immediatamente successive (*ramo then*) racchiuse nella parentesi graffe, altrimenti (se non è verificata) si prosegue con le istruzioni tra parentesi graffe dopo l'istruzione «altrimenti» (*ramo else*)

se ($i \leq 10$)

{...fai qualcosa...} → *ramo then*

altrimenti

{...fai qualcos'altro...} → *ramo else*

- «Calcola il valore dell'espressione $i \leq 10$: se è vero, esegui le istruzioni nel ramo then, altrimenti prosegui per il ramo else»

Esempio (uguale a quello su diagrammi di flusso)

1. var A, B
2. leggi A
3. leggi B
4. var C
5. se (A<B)
6. { C = A }
7. altrimenti
8. { C = B }
9. stampa C

- *Cosa fa questo algoritmo?*
 - Risposta: legge due numeri da tastiera, calcola il minimo tra i due e stampa a video il minore dei due

Iterazione o Ripetizione

Esprime la ripetizione di una sequenza di istruzioni.

Nel caso più generale (ripetizione enumerativa), è costituita da:

- **Inizializzazione:** assegnazione dei valori iniziali alle variabili caratteristiche del ciclo (viene eseguita una sola volta);
- **Controllo:** determina, in base al valore delle variabili che controllano l'esecuzione del ciclo se il ciclo deve essere ripetuto o meno;
- **Corpo:** esecuzione delle istruzioni fondamentali del ciclo che devono essere eseguite in modo ripetitivo;
- **Modifica:** modifica dei valori che controllano l'esecuzione del ciclo (eseguito ad ogni iterazione).

Iterazione

Comando :

ripeti finché (condizione)
{ ... istruzioni ... }

Controllo la condizione, se è vera si eseguono le istruzioni all'interno delle parentesi graffe, altrimenti si prosegue senza eseguirle

Esempio

(sempre uguale a quello sulle slides dei diagrammi di flusso)

```
1. var n
2. leggi n
3. var i := 1           → Inizializzazione
4. ripeti finché (i<=n) → Controllo
5. {
6.     stampa i         → Corpo
7.     i := i + 1      → Modifica
8. }
```

Possibili esercizi da fare

Mediante pseudocodice, descrivere le istruzioni da eseguire per poter completare la richiesta:

- Leggere da tastiera due numeri e stampare a video la loro somma
- Leggere da tastiera due numeri e stampare a video il maggiore dei due
- Leggere da tastiera due numeri, calcolarne il prodotto usando sequenza di somme anziché usare l'operatore $*$, e stampare a video il risultato
- Calcolare il massimo di una sequenza di numeri interi letti da tastiera