

## La programmazione



## Sviluppo del software

- **problema**
- **idea (soluzione informale)**
- **algoritmo (soluzione formale)**
- **programma (traduzione dell'algoritmo in una forma comprensibile da un elaboratore elettronico)**
- **test (su molti casi, con particolare attenzione ai casi limite)**
- **documentazione (manuale utente + manuale del programmatore)**

## Algoritmo



**Un algoritmo può essere considerato un insieme di regole per svolgere un dato compito (risolvere un problema).**

**Il nome deriva dal matematico persiano Muhammad ibn Mūsa 'l-Khwārizmī (780-850).**

**Un algoritmo deve:**

- **terminare in un tempo finito**
- **produrre un effetto osservabile**
- **essere deterministico, ossia produrre gli stessi risultati a partire dalle stesse condizioni iniziali**

## Esecuzione di un algoritmo

**Vengono eseguite in sequenza le operazioni che lo costituiscono.**

**Esistono algoritmi che prevedono:**

- **una sequenza di esecuzione unica**
- **sequenze di esecuzione multiple**

### **Esempio: sequenza di esecuzione unica**

**Dato il valore di X, calcolare:  $Y = 5X + 3$**

**Sequenza di esecuzione:**

- 1. ricevo il valore di X**
- 2. multiplico X per 5 (sia Z il risultato)**
- 3. sommo 3 a Z (sia Y il risultato)**
- 4. visualizzo Y**

### **Esempio: sequenze di esecuzione multiple**

**Dato il valore di X, calcolare la radice quadrata di  $X+5$ .**

**Sequenza di esecuzione:**

- 1. ricevo il valore di X**
- 2. sommo 5 a X (sia Y il risultato)**
- 3a. se Y è positivo o nullo, calcolo la sua radice quadrata e la visualizzo**
- 3b. se Y è negativo, indico che è impossibile calcolare la sua radice quadrata**

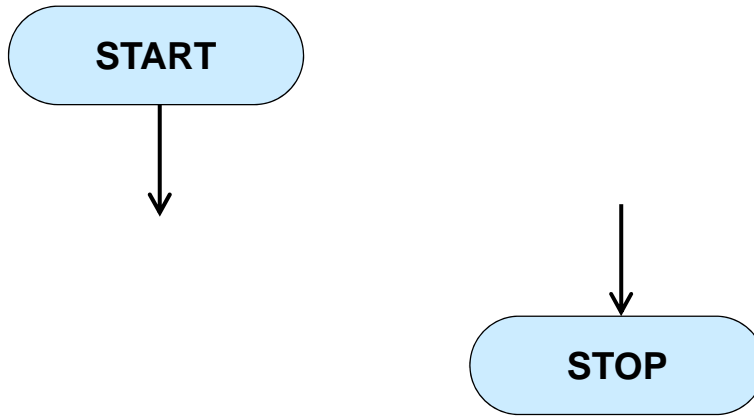
## **Formalizzazione di una soluzione**

- **pseudo-linguaggio (o pseudo-codice)**
  - **vantaggi: immediato**
  - **svantaggi: descrizione dell'algoritmo poco astratta, interpretazione più complicata**
- **diagrammi di flusso**
  - **vantaggi: più intuitivi perché usano un formalismo grafico, descrizione dell'algoritmo più astratta**
  - **svantaggi: richiedono l'apprendimento della funzione dei vari tipi di blocco**

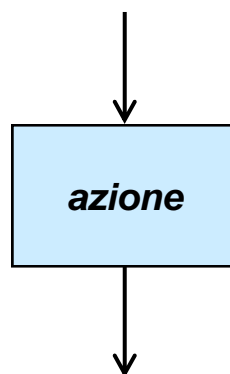
## **Diagrammi di flusso (flow-chart)**

- **metodo grafico per descrivere in modo formale un algoritmo**
- **blocchi base per descrivere:**
  - **azioni**
  - **decisioni (solo binarie, ossia della logica Booleana)**
- **archi orientati per descrivere la sequenza di svolgimento delle azioni**

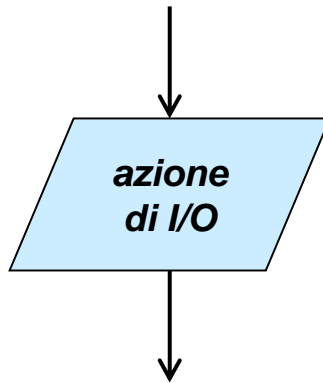
### Blocchi di inizio e fine



### Blocco di azione



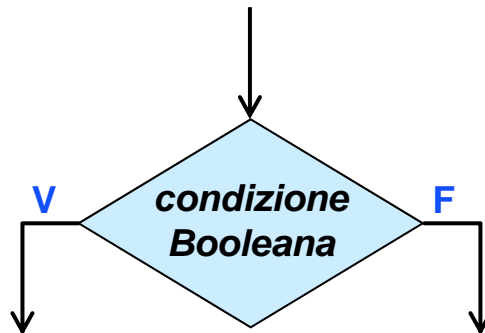
## Blocco di Input/Output



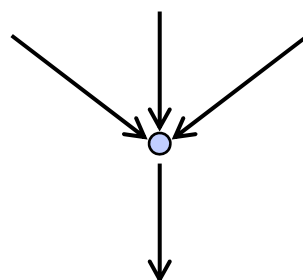
## Blocco di inizializzazione



## Blocco di decisione binaria



## Connettore



## Regole

- **uno ed un solo blocco START**
- **uno ed un solo blocco STOP**
- **tutti gli archi devono avere origine e fine in un blocco**

## Diagrammi di flusso strutturati

Un diagramma di flusso è detto *strutturato* se contiene solo un insieme predefinito di strutture:

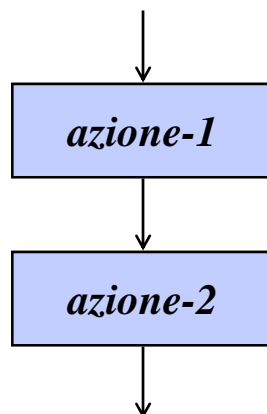
- **sequenze**
- **decisioni**
  - **IF-THEN-ELSE**
  - **IF-THEN**
- **cicli**
  - **WHILE**
  - **REPEAT**

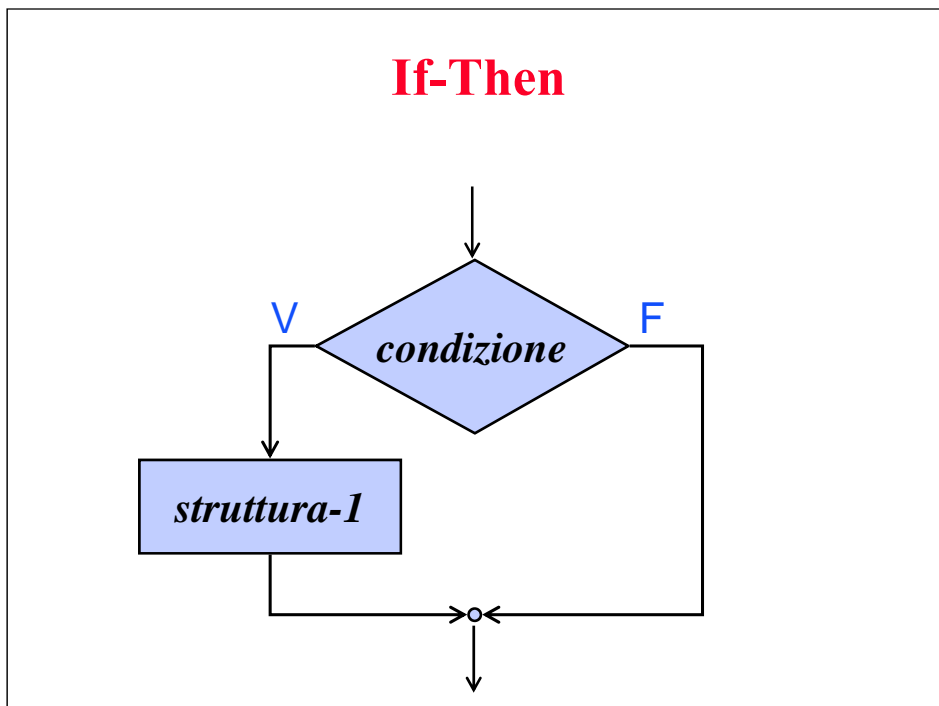
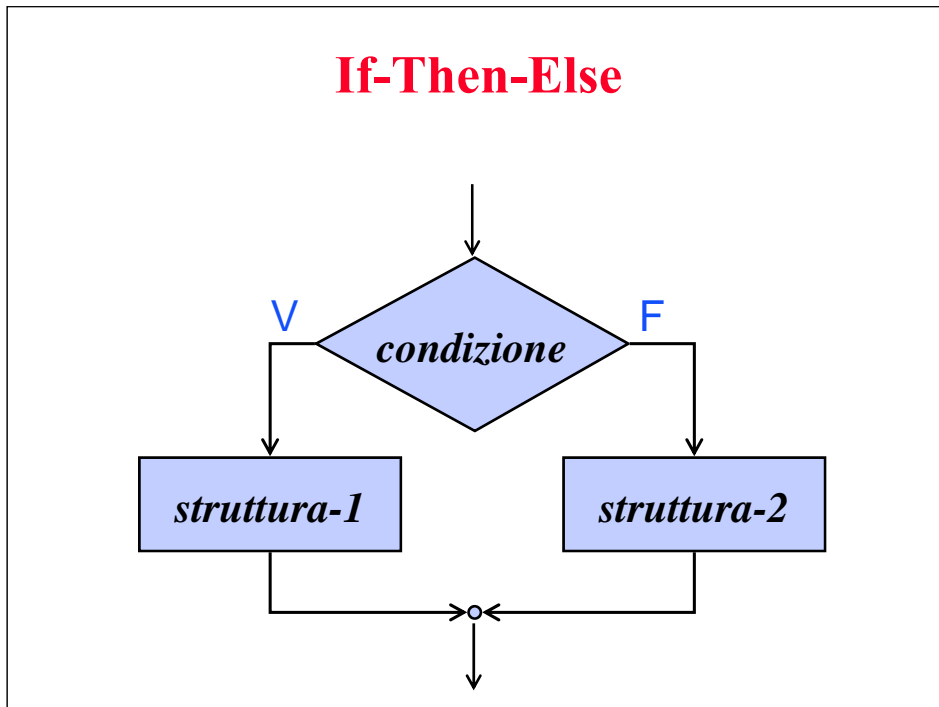


## Teorema di Böhm - Jacopini

*Qualunque diagramma di flusso è sempre trasformabile in un diagramma di flusso strutturato equivalente a quello dato.*

### Sequenza

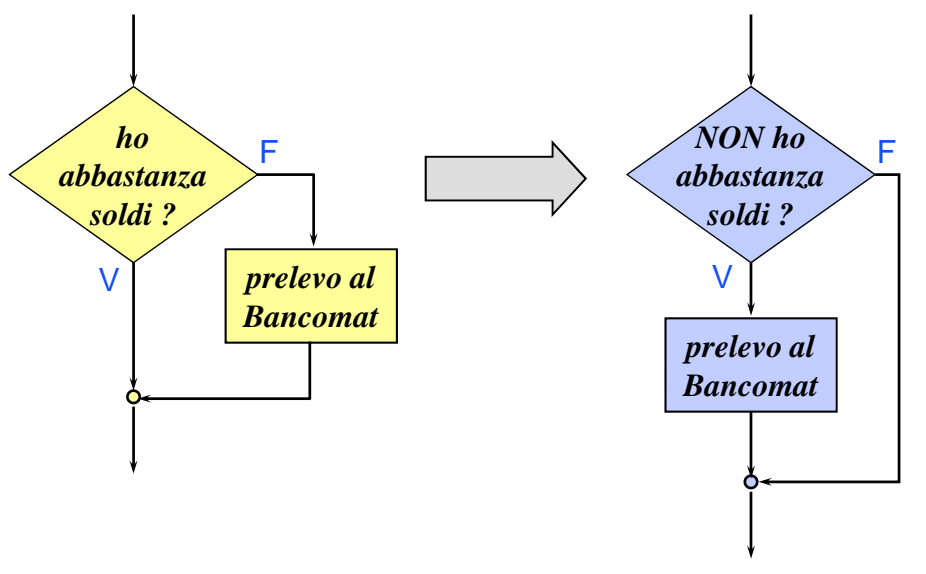


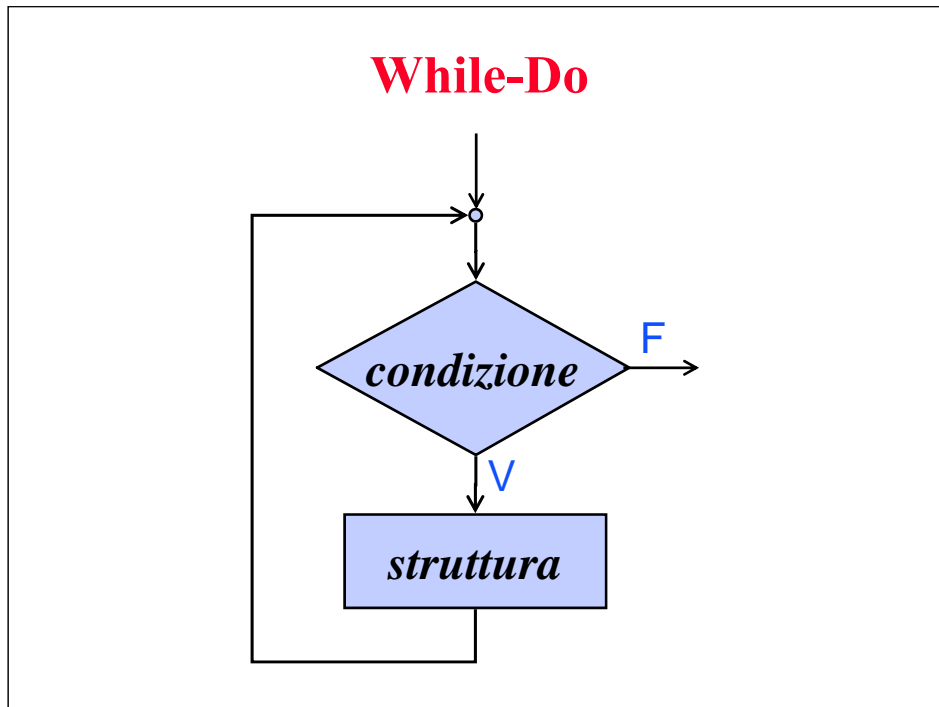


## If-Else

- **non esiste un blocco If-Else ...**
- **... perché non è necessario!**
- **basta usare un blocco If-Then in cui la condizione sia stata invertita (negata)**

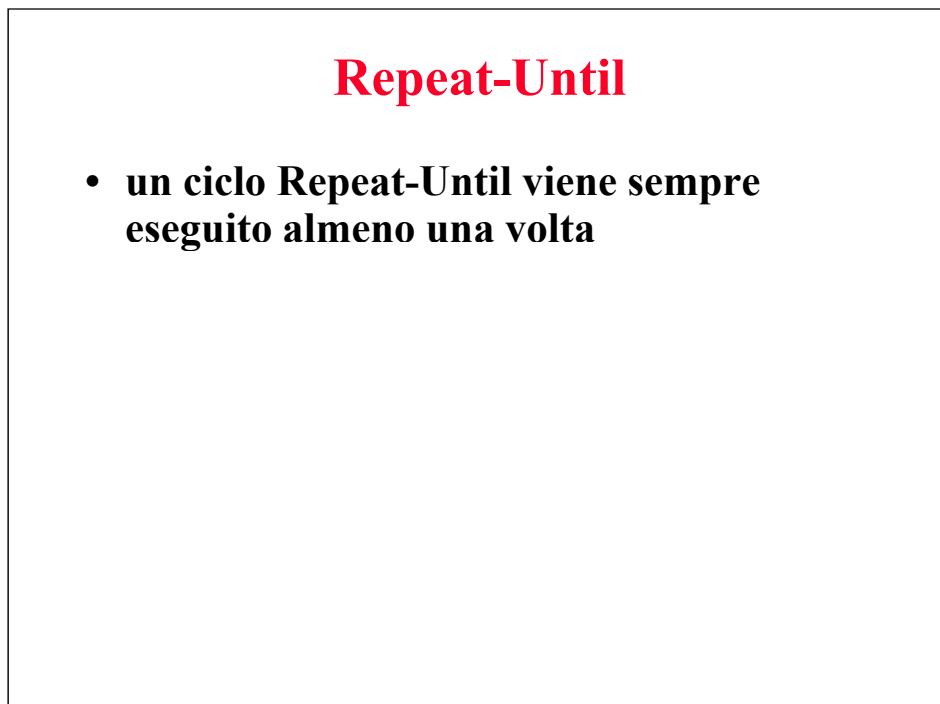
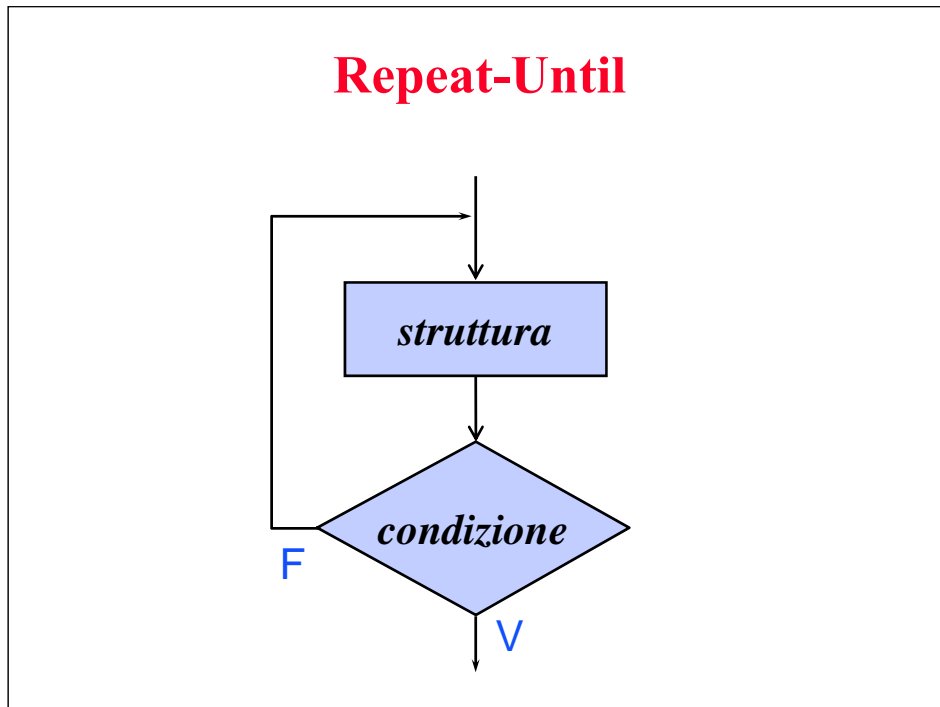
## If-Else (esempio)





## While-Do

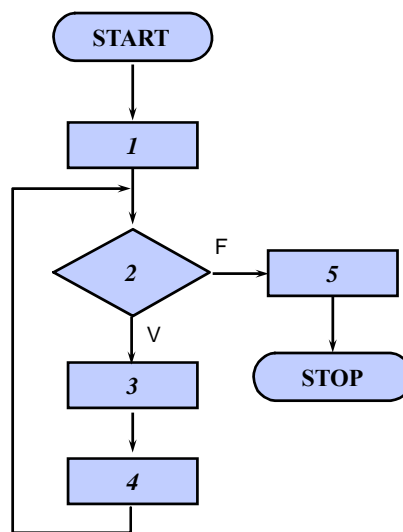
- **la parte ciclica viene eseguita quando la condizione è vera**
- **se abbiamo un ciclo che viene eseguito quando la condizione è falsa, allora occorre trasformarlo in un While-Do mettendo la condizione negata**
- **un ciclo While-Do può essere eseguito zero o più volte**
- **viene eseguito zero volte quando la condizione è subito falsa**



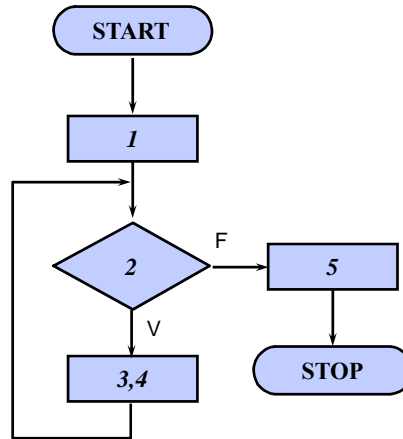
## Verifica di strutturazione

- P1. etichettare ogni blocco**
- P2. sostituire ad ogni insieme strutturato un blocco avente come etichetta l'unione delle etichette dei blocchi che lo costituiscono**
- P3. se al passo P2 si è fatta almeno una sostituzione, ripetere il passo P2**
- P4. se alla fine si ottiene un diagramma lineare (una sequenza), allora il diagramma originale è strutturato**

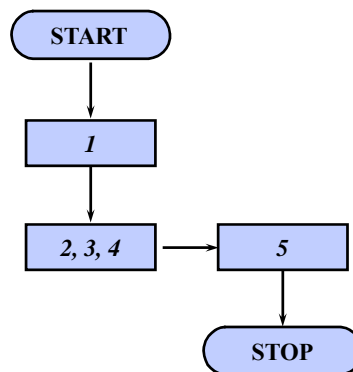
## Esempio: diagramma strutturato



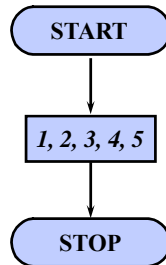
### Esempio: diagramma strutturato



### Esempio: diagramma strutturato



## Esempio: diagramma strutturato



## Esempio: diagramma non strutturato

*Il ciclo non ha  
un test all'inizio  
o alla fine*

