

1 - Programmazione e problemi

Cos'è la programmazione?

È il termine con cui indichiamo l'esigenza di risolvere un problema per trasformarlo in un programma, di cui sia il computer l'esecutore stesso.

Cos'è un problema?

Il problema è l'impossibilità di ottenere qualcosa immediatamente, spetta al risolutore trovare una soluzione all'unico fine di raggiungere un obiettivo e produrre un risultato. Quindi è il cercare una soluzione ad una situazione per ottenere un risultato.

Come si risolve un problema con la programmazione?

La programmazione stessa consiste nel:

- Ricondurre il problema a problemi primitivi (cioè azioni primitive)
- Organizzare e utilizzare le "risorse" dell'elaboratore

Il programma è il prodotto finale di un lavoro che comincia dal formulare il problema e termina con una procedura eseguibile su un computer in tempi accettabili.

Problem solving

Il problem solving non è altro che il passaggio dalla formulazione del problema all'individuazione del metodo solutivo.

Per passare dalla formulazione all'individuazione abbiamo bisogno di:

- Una descrizione, anche parziale, di una situazione iniziale e di una situazione finale desiderata
- Un insieme di operatori applicabili a situazioni per trasformarle in nuove situazioni, espressi in un linguaggio comprensibile a chi descrive il problema

Nel problem solving troviamo 3 fasi fondamentali

1. Formulazione:

Nei problemi reali molte volte i dati possono essere insufficienti o sovrabbondanti, come le condizioni stesse che possono essere anche contraddittorie, in particolare questo avviene quando si parte da un problema relativo ad oggetti reali per ricondurlo ad un problema matematico.

Per poter risolvere i quesiti ci si basa sul processo dell'**analisi**, dove si scompone il problema e si individuano gli oggetti da operare e i risultati da ottenere.

Nella formulazione di un problema bisogna **chiarificare**, cioè creare una descrizione esplicita e completa di **COSA** deve essere risolto, quali sono i risultati che ci si attende, i dati in ingresso forniti dall'esecutore, vanno esplicitate proprietà e relazioni tra di essi, ipotesi sui dati (il loro tipo, se sono validi, se sono randomici e le condizioni) e sulle incognite(stesso discorso dei dati) ed infine vanno eliminate ambiguità, dettagli inutili o possibili errori.

Il chiarimento del problema avviene:

- Se chi propone il problema è disponibile
- Si pongono domande puntuali riguardo lo scopo del problema, i dati e le relazioni intercorrenti tra incognita e dati

Altrimenti facendo opportune ipotesi definendo dei campioni di dati

La fase di chiarifica **può** raffinare e definire meglio, eventualmente ricorrendo a delle ipotesi semplificative quanto non esplicitamente presente dal problema come fornita dal committente e soprattutto **non deve** disattendere o contravvenire a quanto esplicitamente riportato dal problema

2. Costruzione del metodo di soluzione: progettare un metodo composto da:

- Operazioni semplici disponibili nel linguaggio conosciuto dall'esecutore
 - Modalità secondo cui possono essere connesse e composte per realizzare operazioni più complesse
- Queste operazioni semplici vanno eseguite uno dopo l'altra per poter creare queste operazioni più complesse, un'operazione di **sequenzializzazione**

Il modello quindi per la costruzione di un metodo solutivo è l' **algoritmo**, cioè una serie di prescrizioni o azioni da compiere per risolvere un problema. Per poterne creare uno serve:

1. Esaminare il problema
2. Costruirne un'astrazione (trasformare i problemi in oggetti formali)
3. Rappresentarla formalmente
4. Individuare una sequenza di azioni che, eseguite, risolvano il problema nel mondo dell'astrazione
5. L'algoritmo a questo punto è pronto per essere eseguito dall'esecutore

Lo stesso algoritmo deve presentare le seguenti proprietà:

- **Finitezza:**
Le azioni di un algoritmo devono essere un insieme finito e una qualunque esecuzione dell'algoritmo deve terminare in un tempo finito
- **Generalità:**
L'algoritmo deve risolvere una classe di problemi, ovvero deve essere applicabile a qualsiasi insieme di dati appartenenti al dominio di definizione dell'algoritmo
- **Non ambiguità:**
L'algoritmo non deve essere costituito da azioni che si contraddicono

La descrizione dell'algoritmo deve essere:

- **Univoca:** non deve dare spazio ad interpretazioni errate
- **Completa:** devono essere previste esattamente tutte le azioni necessarie
- **Ripetibile:** deve poter essere eseguito da più esecutori, senza che il suo risultato sia diverso, quindi con garanzia di successo

Quest'ultima condizione deve essere rispettata per poter essere sempre eseguita da un esecutore qualunque

Un algoritmo deve essere codificato , cioè scritto in codice con un linguaggio di programmazione

3. **Esecuzione:** Processo dove il programma viene effettivamente eseguito, dopo che è stato compilato.
 - L'esecutore deve avere tutte le istruzioni che gli consentano di trasformare i dati di input in dati di output.
 - Gli errori nel processo di esecuzione sono errori logici, ovvero errori che trasformano i dati di input in risultati errati
 - Gli errori nel programma sono errori sintattici che impediscono l'esecuzione del programma

Cosa è un programma?

Il programma è la traduzione, in un linguaggio comprensibile all'elaborator, dell'algoritmo; Ovvero della descrizione del procedimento di soluzione, con indicazioni sui dati di ingresso e di uscita.

Tramite il programma si comunica all'elaboratore:

- Quali dati di ingresso deve trattare,
- Come deve operarci,
- Quali dati deve dare come risultati

Cosa significa risolvere un problema?

- **Scegliere l'astrazione**,definire un insieme di dati e oggetti che caratterizzano la realtà e che siano rilevanti
- **Scegliere la rappresentazione giusta** dei dati (interi, float etc.)
- **Individuare un procedimento algoritmico** tramite utilizzo di algoritmi (e poi programmi)
- **Scomporre eventualmente il procedimento in sotto procedimenti**