

Programmazione ad oggetti

2016-17

La conoscenza che non entra nella carne
è solo rumore!

Analisi e programmazione

- Tramite un elaboratore si possono risolvere problemi di varia natura: emissione di certificati anagrafici, gestione dei c/c di un istituto di credito, prenotazioni ferroviarie...
- Il problema deve essere formulato in modo opportuno, perché sia possibile utilizzare un elaboratore per la sua soluzione
- Per analisi e programmazione si intende l'insieme delle attività preliminari atte a risolvere problemi utilizzando un elaboratore, dalla formulazione del problema fino alla predisposizione dell'elaboratore
- Scopo dell'analisi: definire un algoritmo
- Scopo della programmazione: definire un programma

Analisi e programmazione

- **Algoritmo:** elenco finito di istruzioni, che specificano le operazioni eseguendo le quali si risolve una classe di problemi
- Un particolare problema della classe viene risolto utilizzando l'apposito algoritmo sui dati che lo caratterizzano
- Un algoritmo non può essere eseguito direttamente dall'elaboratore
- **Programma:** *ricetta* che traduce l'algoritmo ed è direttamente comprensibile, pertanto eseguibile, da parte di un elaboratore
- **Linguaggio di programmazione:** linguaggio rigoroso che permette la formalizzazione di un algoritmo in un programma

Le fasi del procedimento di analisi e programmazione

- Problema
- ANALISI
- Algoritmo
- PROGRAMMAZIONE
- Programma
- ELABORAZIONE
- Risultati

Relazioni tra problema, analisi, algoritmo, programmazione, programma, dati ed elaborazione

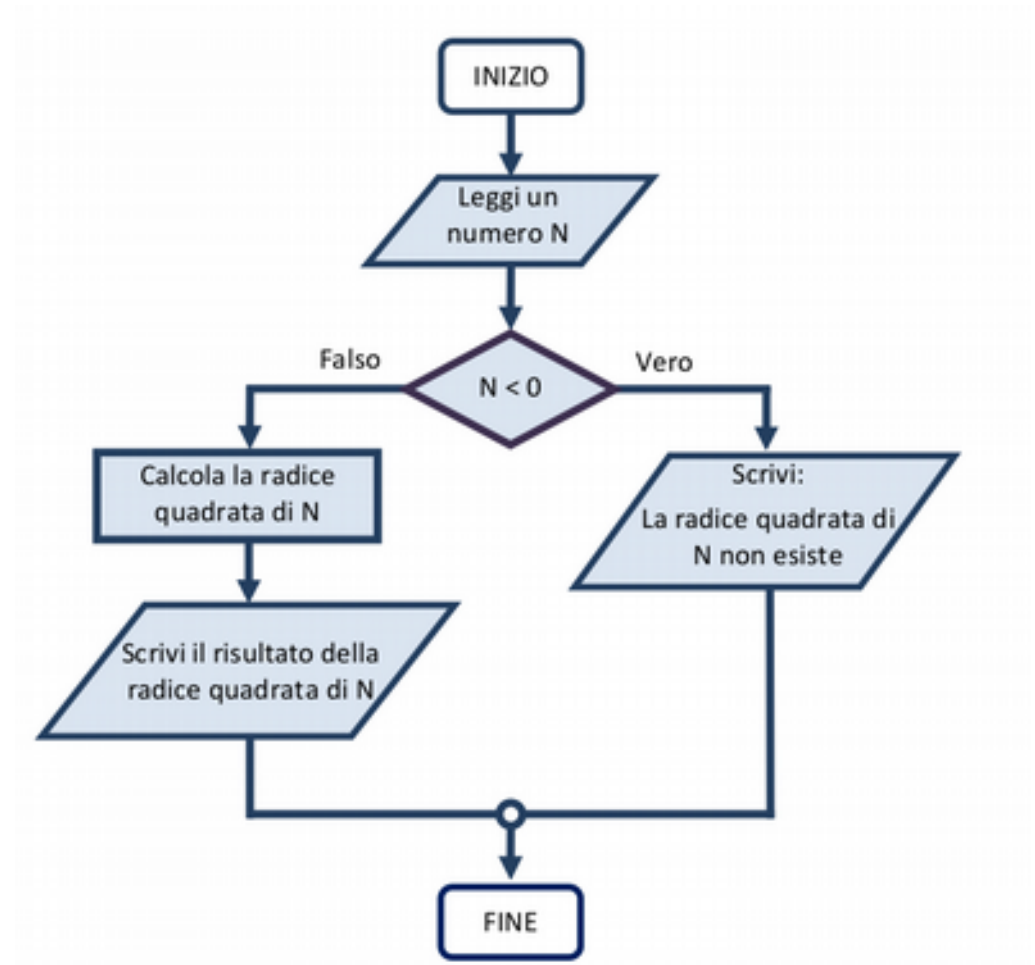
Dati

Proprietà degli algoritmi





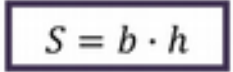
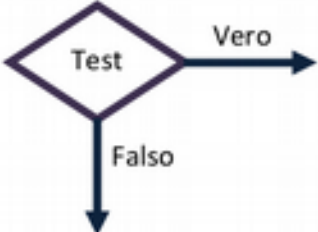


- **Finitezza:** ogni algoritmo deve essere finito, cioè ogni singola istruzione deve poter essere eseguita in tempo finito ed un numero finito di volte
- **Generalità:** ogni algoritmo deve fornire la soluzione per una classe di problemi; deve pertanto essere applicabile a qualsiasi insieme di dati appartenenti all'insieme di definizione o dominio dell'algoritmo e deve produrre risultati che appartengano all'insieme di arrivo o codominio
- **Non ambiguità:** devono essere definiti in modo univoco i passi successivi da eseguire; devono essere evitati paradossi, contraddizioni ed ambiguità; il significato di ogni istruzione deve essere univoco per chiunque esegua l'algoritmo
- **Eseguibile:** le istruzioni devono poter essere eseguite materialmente dall'esecutore
- **Deterministico:** partendo dagli stessi dati iniziali deve portare sempre allo stesso risultato finale indipendentemente dall'esecutore
- **Completo:** deve contemplare tutti i casi che si possono verificare durante l'esecuzione

Rappresentare algoritmi

- Vedi esempi di algoritmi
- Vedi focus slide su algoritmi



Rappresentare algoritmi

Oggetto grafico	Denominazione	Significato
	Punto di partenza	Rappresenta un'azione che avvia il processo
	Punto di fine	Rappresenta un'azione che conclude il processo
	Leggi	Rappresenta un'azione di ingresso dati
	Scrivi	Rappresenta un'azione di uscita dei risultati
	Elaborazione	Rappresenta il comando o calcolo da eseguire
	Test	Rappresenta la scelta fra due possibili percorsi
	Linea di flusso	Indica la direzione del percorso del flusso
	Connessione	Rappresenta il punto d'inserimento nel grafico (generalmente contiene una lettera o un numero d'ordine)

- Vedi esempi di algoritmi
- Vedi focus slide su algoritmi