

Corso di Programmazione

Problem solving: Fasi di Sviluppo di un programma

Prof. ssa Teresa Roselli
teresa.roselli@uniba.it

Sviluppo di un Programma

- Risoluzione di un problema mediante un calcolatore
 - Partire dalla descrizione del problema
 - In linguaggio naturale
 - per giungere alla stesura di un programma
 - In un certo linguaggio di programmazione

Sviluppo di un Programma

Fasi

- (Studio di fattibilità)
- Analisi
 - Chiarifica del problema
 - Cosa
- Progettazione
 - Individuazione di una strategia di soluzione
 - Come
 - Scelta delle strutture di dati

Sviluppo di un Programma

Fasi

- Codifica
 - Scrittura del programma
- Verifica (e correzione)
 - Test del programma
 - Rimanda ad una delle fasi precedenti
- Manutenzione
 - Correttiva
 - Adattativa
 - Migliorativa

Sviluppo di un Programma

- Accumulo degli errori
 - Gli errori in ciascuna fase si ripercuotono su tutte le fasi successive
 - Più costosi
 - Ciascuna fase può aggiungere errori a quelli delle fasi precedenti
- Vari approcci
 - Differente sequenza di esecuzione delle fasi di sviluppo

Sviluppo di un Programma

- Documentazione di ogni fase
 - Necessaria per chi riprenderà in seguito il programma per modificarlo
 - L'autore stesso
 - Altre persone
- La documentazione in output da ciascuna fase è input per la fase successiva

Problema

Parti principali

- *Incognita/e* o *Obiettivo/i*
 - Ciò che si vuole trovare
- *Dati*
 - Ciò che è dato o conosciuto
- *Condizione*
 - Specifica come l'incognita è connessa ai dati
 - Per mezzo di quali relazioni

Problema

- Comprendere il problema
 - Il suo significato
 - Il suo scopo
- Vederne molto chiaramente le parti principali
 - Incognita
 - Condizioni
 - Dati

Problema

- Scopo di un problema
 - Trovare
 - Trovare, produrre, costruire, identificare, elencare, caratterizzare, ...
- un certo oggetto
 - L'*incognita* del problema
- che soddisfa la *condizione* del problema
 - Collega l'*incognita* ai *dati* del problema

Formulazione del Problema

- I problemi reali non sempre sono ben formulati
 - I dati
 - Sono sufficienti?
 - Sono sovrabbondanti?
 - La condizione
 - È sufficiente, è sovrabbondante?
 - È contraddittoria?

Formulazione del Problema

- Caso di problemi perfettamente determinati (ad es. un teorema)
 - Tutte le affermazioni contenute nell'ipotesi del teorema sono essenziali per dimostrare la tesi
 - Se la dimostrazione non tiene conto di una qualsiasi parte dell'ipotesi è senz'altro errata
 - Analogamente: se una parte della condizione di un problema che “chiede di trovare” non viene presa in considerazione, si finisce col risolvere un problema diverso da quello originale

Formulazione del Problema

- Un problema *chiaramente enunciato* deve specificare
 - La *categoria* a cui appartiene l'incognita
 - Che genere di cosa bisogna trovare
 - Esempio: un numero, una parola, ecc.
 - La *condizione* che deve soddisfare l'incognita
 - Il sottoinsieme di quegli oggetti che soddisfano la condizione rappresenta le soluzioni
 - I *dati*

Formulazione del Problema

- Quasi tutti i problemi sono mal formulati
 - Non chiaramente enunciati
 - Omissioni
 - Ambiguità
 - Imprecisioni
- Possibilità di non capire bene le richieste del problema
 - Necessità di esplicitare le ipotesi

Formulazione del Problema

Esempio

- Problema
 - Calcolare l'*altezza* di uno stabile avente n piani di h metri
- Obiettivo
 - Trovare il valore dell'*altezza complessiva* dello stabile
- Dati
 - *Numero di piani* dello stabile (n)
 - *Altezza dei piani* (h)

Formulazione del Problema

Esempio

- *Imprecisione ed ambiguità*
 - Nell'obiettivo (che rappresenta l'incognita)
 - Nei dati
- *L'incognita* è un valore numerico (espressa nella stessa unità di misura dei valori in ingresso) che rappresenta l'altezza complessiva di uno stabile

Formulazione del Problema

Esempio

- Cosa si intende per *altezza complessiva*?
 - la distanza fra suolo e parapetto del terrazzo?
 - o questo è escluso?
- Cosa si intende per numero di piani?
 - il piano terra costituisce un piano?
 - ed è compreso nel dato n fornito?
- Cosa si intende per *altezza dei piani*?
 - la distanza che intercorre fra pavimento e soffitto?
 - o fra due solai successivi?

Formulazione del Problema

- Tutti i problemi enunciati a parole contengono delle ipotesi semplificatrici sottintese
 - Necessità di un certo lavoro preliminare di interpretazione e di astrazione da parte di colui che risolve il problema
 - In particolare questo avviene quando si parte da un problema relativo ad oggetti reali per ricondurlo ad un problema matematico

Formulazione del Problema

Esempio

- Un aeroplano di pattuglia percorre 220 miglia all'ora in atmosfera tranquilla. Esso trasporta benzina per 4 ore di volo sicuro. Se decolla in pattuglia contro un vento di 20 miglia orarie, di quanto può allontanarsi per ritornare sano e salvo?
 - È sottinteso:
 - Che si suppone che il vento continui a soffiare con la stessa intensità durante tutto il volo
 - Che l'aeroplano voli in linea retta
 - Che il tempo necessario per cambiare direzione nel punto estremo sia trascurabile
 - ...e così via

Formulazione di un Problema

- NON iniziare a risolvere un problema prima di averlo *capito*
 - Lo si è capito quando si è in grado di riformulare il problema indicando chiaramente
 - Le *incognite*
 - I *dati*
- e spiegando
 - La *condizione* (ovvero l'insieme delle relazioni che collegano i dati con le incognite)

Chiarifica

- Riformulazione del problema – considerandolo risolto – cercando di vedere con chiarezza, in ordine conveniente, tutte le relazioni che devono intercorrere fra le incognite ed i dati, al fine di soddisfare l'obiettivo del problema.
- Tutte queste relazioni rappresentano la condizione del problema.
 - Se n è il numero delle incognite, ottenere n equazioni

Chiarifica Esempio

- Un fattore ha polli e conigli. Questi animali hanno 50 teste e 140 zampe. Quanti polli e quanti conigli ha il fattore?
 - Bisogna trovare 2 numeri, p e c (*incognite*), che rappresentano il numero dei polli e il numero dei conigli rispettivamente
 - Sono forniti il numero delle teste (50) ed il numero delle zampe (140) (*dati*)

$$\begin{array}{l} p + c = 50 \\ 2p + 4c = 140 \end{array} \quad \left. \vphantom{\begin{array}{l} p + c = 50 \\ 2p + 4c = 140 \end{array}} \right\} \text{ *condizione* }$$

- 2 = nro zampe per ciascun pollo
 - 4 = nro zampe per ciascun coniglio
- $$\left. \vphantom{\begin{array}{l} 2 = \text{nro zampe per ciascun pollo} \\ 4 = \text{nro zampe per ciascun coniglio} \end{array}} \right\} \text{ Ulteriori dati emersi dall'analisi del problema }$$

Chiarifica

- 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

Chiarifica

- La fase di chiarifica
 - PUÒ raffinare e definire meglio, eventualmente ricorrendo a delle ipotesi semplificative, quanto non esplicitamente presente nella traccia del problema come fornita dal committente
 - NON DEVE disattendere o contravvenire a quanto esplicitamente riportato nella traccia del problema come fornita dal committente

Chiarifica

- Generalmente per essere in grado di comprendere il problema bisogna avere delle conoscenze pertinenti
 - Dominio del problema
 - Esempio: problemi geometrici
 - Il teorema di Pitagora, la proporzionalità dei lati nei triangoli simili, formule per aree e volumi, ecc.
 - In generale si fa ricorso alle definizioni

Analisi

- Input: Traccia del problema
- Processo: Chiarire con precisione *cosa* vuole il problema (non *come* va fatto)
 - Obiettivo del problema
 - Dati a disposizione
 - Risultati desiderati
 - Ambiguità e imprecisioni nella definizione del problema
 - Obiettivo, dati
 - Possibilità di capire male le richieste
- Output: Descrizione dettagliata discorsiva del problema

Analisi

Punti chiave

- Dati (input)
 - Formato o tipo, ordine, limiti sui valori, limiti sul volume, fine dei dati, ipotesi di ordinamento
- Risultati (output)
 - Contenuto, formato, ordine, limite sul volume
- Errori e Casi limite
 - Tipi di errori (di input e/o di elaborazione) ed azioni da intraprendere
 - Campioni di input e di output corrispondenti

Analisi Ambiguità

- Alcune informazioni su cui è necessario fare delle assunzioni
 - Ordine dell'input
 - $8/7 = 8$ luglio o 7 agosto?
 - Limiti sui valori
 - $0 < \text{età} < 125$
 - Errore di elaborazione
 - $\forall \Delta < 0$ nelle equazioni di II grado

Analisi Esempio

- Data una lista di numeri, stampare il 1° numero della lista, il 2° e così via fino al più grande dei valori presenti nella lista

3 8 2 25 13 19

- Necessità di una fase di chiarifica del problema

Analisi Esempio

- Input: lista di numeri
 - Insieme composto da non più di 100 valori interi positivi (>0).
 - Un valore fittizio nullo, aggiunto in coda all'insieme, definisce la fine dei dati
 - 0 non fa parte dei valori
 - I dati di ingresso non sono ordinati rispetto al valore
- Campione di Ingresso: 1 7 3 9 5 8

Analisi Esempio

- Output: elenco di valori
 - Si vuole la stampa di una *colonna* di numeri che corrispondono ai numeri di ingresso nello *stesso ordine*
 - La colonna inizia con il primo numero e termina quando l'ultimo numero stampato è il più grande dell'intero insieme di ingresso
 - Verranno stampati *al massimo* un numero di valori *pari alla numerosità* dell'insieme di input se il più grande è l'ultimo
- Campione di Uscita: 1 7 3 9 (incolonnati)

Analisi Esempio

- Input
 - Quali numeri?
 - Reali, interi
 - Limiti sui valori?
 - Qual è il minimo ammissibile? Qual è il massimo?
 - Ad es., se rappresentassimo età non avrebbero senso dei valori negativi
 - Se fossero anni di nascita sarebbero fuori range tutti i valori > 2002
 - Limiti sul volume?
 - Quanti al massimo potranno essere i dati?

Analisi Esempio

- Come si riconosce la fine dei dati?
- Sono ordinati?
 - Qual è il criterio di ordinamento?
 - Crescente, decrescente, non crescente, non decrescente
 - Lessicografico, ...
 - Rispetto a cosa (per dati non atomici)?
 - Cognome, anno di nascita, ...
- Controlli sui dati in ingresso
 - Da realizzarsi nel programma

Analisi Esempio

- Errori e casi limite
 - Valori negativi
 - Scartare il dato, oppure
 - Prendere il valore assoluto
 - Troppi dati (più di 100)
 - Prendere in considerazione solo i primi 100
 - Presenza di più massimi
 - Fermarsi in output alla prima occorrenza del massimo

Analisi Esempio

- Mancanza di dati (lista vuota)
 - Prevedere il controllo adeguato da programma
- Mancanza del flag di fine dati
 - Non rilevabile da programma
- N.B.:
 - Per ogni situazione di errore individuata va almeno inviato un messaggio di notifica all'utente

Progettazione

- Si occupa di definire una strategia di soluzione che porti ad ottenere il risultato desiderato
 - Necessita di una definizione chiara del *cosa* si vuole ottenere
- Organizza la soluzione in un insieme di moduli cooperanti
 - Tecniche basate sul metodo di soluzione di problemi consistente nello scomporre un problema in sottoproblemi più semplici

Progettazione Tecniche di Sviluppo

- Top-Down
 - Raffinamento successivo della procedura di soluzione
 - Raffinamento della descrizione dei dati
- Bottom-Up
 - Sfrutta algoritmi codificati già esistenti raggruppandoli per adattarli a nuove situazioni
- Ibrida
 - Basata su una cooperazione fra le tecniche precedenti

Progettazione

- Input: Analisi del problema
- Processo: Chiarire con precisione *come* va ottenuto lo scopo voluto dal problema
 - Strategia di soluzione
 - Strutture di dati
- Output: Descrizione dettagliata della struttura della soluzione

Progettazione

- Individuare una possibile scomposizione in sottoproblemi più semplici
- Descrivere un procedimento risolutivo per ciascun sottoproblema
- Riportare l'algoritmo per il problema originario
 - Attenzione: l'algoritmo dipende sempre dal modo in cui sono organizzati i dati

Progettazione

Esempio

- Problema
 - Trovare in una agendina il numero telefonico di una persona di cui sia noto il cognome
 - *Dati di ingresso*
 - Agendina e cognome
 - *Dati di uscita*
 - Numero telefonico
 - Caso particolare: se nell'agenda non esiste quel cognome?
Risposta “non trovato”

Progettazione Esempio

- Organizzazione di un'agenda
 - Pagine consecutive
 - Ordinate alfabeticamente in base all'intestazione
 - Una singola lettera alfabetica
 - Un gruppo di lettere consecutive
 - Sono presenti tutte le 26 lettere!

Progettazione

Esempio

- Ogni pagina contiene cognomi ed i corrispondenti numeri telefonici relativi alla propria intestazione
 - All'interno di ciascuna pagina, i cognomi non sono generalmente ordinati ma raggruppati
 - Esempio: **GH**
 - Gruppo dei cognomi che iniziano con G o con H
 - Non ordinati

Progettazione

Esempio

- Descrivere il metodo da usare per risolvere questo problema
 - Algoritmo di ricerca in una agenda
 - Occorre:
 - Pensare “*come*” si effettua la ricerca in un’agenda telefonica
 - Dare una descrizione *sistematica*