

Programmi

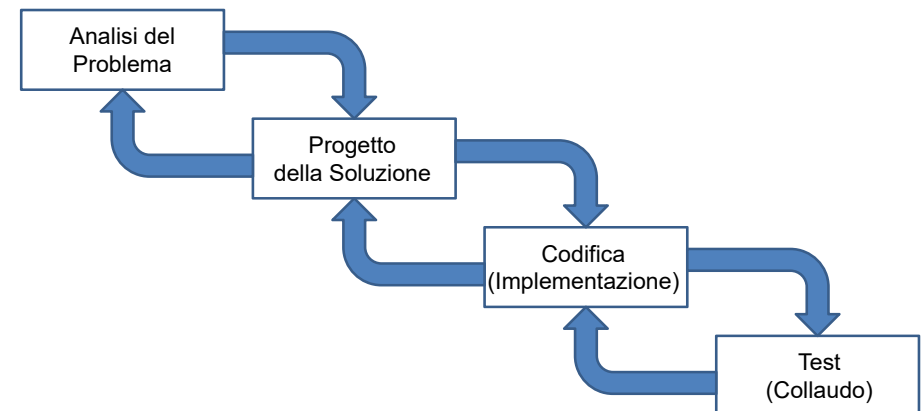
Sommario

- Obiettivo della programmazione e ciclo di sviluppo di programmi
- Istruzioni variabili e tipi
- Sottoprogrammi
- Strutture di controllo
- Ricorsione

Obiettivo

- Risoluzione di un problema mediante un calcolatore
 - Partire dalla descrizione del problema
 - in linguaggio naturale
 - per giungere alla stesura di un programma
 - nel linguaggio di programmazione scelto

Ciclo di sviluppo di programmi



Fasi di Sviluppo

- (Studio di fattibilità)
- Analisi
 - Chiarifica del problema
 - COSA si ha a disposizione ?
 - COSA si deve ottenere ?
- Progettazione
 - Individuazione di una strategia di soluzione
 - COME raggiungere l'obiettivo dato quello di cui si dispone ?
 - Scelta delle strutture di dati
- Codifica
 - Scrittura del programma
- Verifica (e correzione)
 - (Test) del programma
 - Rimanda ad una delle fasi precedenti
- Manutenzione
 - Correttiva, Adattativa, Migliorativa

Programmi

5

Programmazione (1)

- Descrizione del procedimento di soluzione di un problema ad un esecutore meccanico
- Poiché l'esecutore è meccanico, consiste nel
 - Ricondurre il problema da risolvere a problemi primitivi
 - Eseguibili come insieme di azioni primitive
 - Organizzare ed utilizzare le "risorse" dell'elaboratore

Programmi

6

Programmazione (2)

- Trasformazione della descrizione di un algoritmo in un messaggio
 - Insieme di istruzioni codificate in un linguaggio interpretabile da un esecutore
- Passa attraverso un'astrazione
 - Sia delle operazioni che il procedimento prevede
 - Sia degli oggetti su cui il procedimento deve operare

Programmi

7

Programma (1)

- Traduzione della procedura di soluzione in un linguaggio comprensibile alla macchina con indicazioni sui dati di ingresso e uscita
- Comunica al calcolatore **istruzioni operative**
 - Quali dati di ingresso deve trattare
 - Come deve operare su questi dati
 - Quali dati deve dare come risultato

Programmi

8

Programma (2)

- Procedura eseguibile su calcolatore, che rappresenta una soluzione ad un problema
 - Risultato di un lavoro di analisi e progetto che inizia dalla formulazione del problema
 - Corrisponde alla tripla (Dati, Algoritmo, Risultati)
 - [Wirth] Algoritmi + Strutture Dati = Programmi

Programma (3)

- Traduzione di un metodo di soluzione eseguibile in un linguaggio comprensibile alla macchina
 - Descrive come vanno elaborati insiemi di valori che rappresentano le entità del problema
 - Usa rappresentazioni simboliche per estendere l'applicabilità del metodo di soluzione a valori diversi
 - Uso di variabili

Dati

- Entità su cui lavora il programma
 - Costanti
 - Variabili
- Rappresentati come sequenze di bit
 - Nei linguaggi ad alto livello il programmatore può ignorare i dettagli della rappresentazione
 - Tipo di dato

Istruzioni

- Operative
 - Lavorano su rappresentazioni delle entità del problema
- Dichiarative
 - Consentono di definire come interpretare e rappresentare tali entità in termini di variabili nel programma
 - Totale caratterizzazione mediante la definizione di:
 - Un nome
 - Un tipo

Istruzioni Dichiarative

- Definiscono le aree di memoria in cui sono conservati i dati cui fa riferimento un algoritmo
 - Predispongono le posizioni di memoria da utilizzare
 - Associano un nome a ciascuna di esse
 - Identificatore
 - Determinano il tipo di dati che vi possono essere memorizzati
 - Insieme dei valori permessi
 - Insieme delle operazioni applicabili

Tipi di Istruzioni

- Un linguaggio di programmazione dispone di:
 - Istruzioni di ingresso
 - Permettono all'esecutore di conoscere informazioni fornite dall'esterno
 - Istruzioni di uscita
 - Permettono all'esecutore di notificare all'utente i risultati ottenuti dall'elaborazione
 - Istruzioni operative
 - Permettono di effettuare calcoli o, comunque, operazioni sulle entità astratte rappresentanti gli elementi del problema
 - Strutture di controllo

Istruzioni di Ingresso/Uscita

- Livello di descrizione dell'algoritmo
 - Necessità di indicare i dati su cui operare
- Livello di programma
 - Necessità di comunicare i dati e i risultati
 - Istruzioni di lettura e scrittura

Istruzioni di Uscita

- Consentono di notificare all'utente il valore di una variabile del programma
 - Visualizzazione, stampa, ...
- Attivano un'operazione di scrittura
 - Copiatura su un supporto esterno
 - carta, nastri e dischi magnetici, display, ...
 - del contenuto di un'area di memoria denotata dal nome della variabile che compare nell'istruzione di scrittura
- Esempio: print y

Istruzioni Dichiarative Variabili

- Forniscono una lista contenente i nomi scelti per le variabili e i tipi corrispondenti
 - Convenzione: indicare tutte le variabili
- Necessarie nei linguaggi di programmazione
 - Nel programma si fa riferimento agli indirizzi delle aree di memoria in cui sono conservati i dati

Variabile (1)

- Nome simbolico per denotare un'area di memoria e, tramite essa, il valore contenuto
 - Contiene una rappresentazione di un oggetto su cui l'algoritmo opera
 - Memorizzazione di un valore
 - Istruzioni di ingresso
 - Istruzioni di assegnamento

Variabile (2)

- Caratterizzata da
 - Un nome
 - Identificatore
 - Un valore
 - Un tipo
 - Attributo che specifica l'insieme di valori che la variabile può assumere

Variabile (3)

- Rappresenta una locazione di memoria del computer, contraddistinta da uno specifico indirizzo, che contiene il valore su cui applicare le istruzioni del programma
- Un identificatore denota una coppia
 - Posizione di memoria
 - Quantità in essa contenuta
 - Una limitazione nel rappresentare dati di tipo numerico o alfanumerico viene dalle dimensioni limitate della memoria

Tipo (1)

- Attributo di una variabile che ne specifica ed individua:
 - L'insieme dei valori che la var. può assumere
 - L'insieme di operazioni effettuabili sulla var.
 - Il modo con cui ci si può riferire alla var.
- Per parlarne in termini formali è utile introdurre il concetto di algebra di dati
 - Esempio:
Stipendio: numero intero non negativo

Tipo (2)

- Tripla
 $T = \langle D, C, O \rangle$
 - Dominio
 - Costanti
 - Operatori
 - Funzioni
 - Predicati

Tipo: Esempio

- Tipo dei complessi
 - Dominio
 - Sottoinsieme dei numeri complessi
 - Costanti
 - Parte reale
 - Parte immaginaria
 - Operazioni
 - Addizione $+$: (complesso X complesso) \rightarrow complesso
 - Non ci interessiamo della rappresentazione interna dei valori e delle operazioni

Dichiarazione di Tipo: Utilità

- Definizione del dominio di applicazione del programma
- Comprensione funzionamento dell'algoritmo
- Verifica correttezza del programma
 - Compilatore
 - Programmatore
- Definizione dello spazio di memoria necessario
- Rappresentazione interna
 - Esempio: 18 e 18.0

Tipi Standard

- Tipi più comuni di variabili
 - Interi
 - Reali
 - Logici
 - Caratteri
- Valori rappresentabili limitati
 - Dimensioni della memoria che dovrà ospitarne le variabili
 - Tipo numerico
 - Tipo alfanumerico

Istruzioni

- Nella maggior parte dei linguaggi di programmazione sono presenti diverse tipologie di istruzioni
 - Ingresso/Uscita
 - Assegnamento
 - Strutture di controllo

Istruzioni di Ingresso

- Permettono di acquisire informazioni dall'esterno, inserendole in opportune variabili del programma
- Attivano un'operazione di lettura
 - Assegnazione del valore letto su un supporto di memorizzazione esterno
 - schede, nastri e dischi magnetici, ...
 - ad un'area di memoria individuata dal nome che compare nell'istruzione di lettura
- Esempio: get x

Assegnamento di valori a variabili

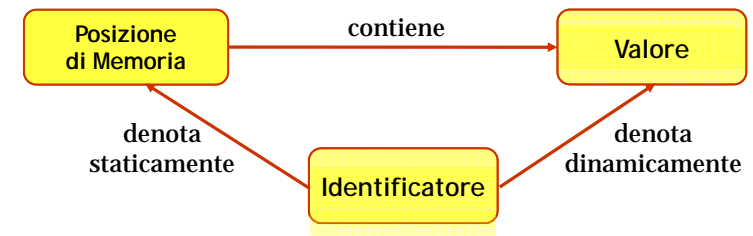
- Avviene in 2 passi
 - Produzione di un nuovo valore
 - Assegnamento di quel valore alla variabile
- L'operazione di assegnamento è indicata con simboli diversi a seconda dei diversi linguaggi di programmazione
 - := (Pascal)
 - = (C, C++, Java, ...)
 - ...
- Confusione fra uguaglianza e assegnazione

Assegnamento

- Per produrre il valore si possono usare espressioni (aritmetiche o logiche) il cui risultato è un singolo valore
- Il valore prodotto dall'espressione a destra viene memorizzato nell'area di memoria riservata alla variabile a sinistra
 - La memorizzazione del valore ottenuto nella locazione di memoria riservata alla variabile implicata nell'assegnamento rimpiazza qualunque valore contenuto in precedenza

Legami degli Identificatori

- Identificatore – Posizione di memoria
 - Statico
- Identificatore – Valore
 - Dinamico nel programma



Legami degli Identificatori

- La dinamicità del legame identificatore-valore consente di scrivere senza contraddizioni assegnazioni del tipo
- $x = x + 1$
 - Alla posizione di memoria identificata da x assegna il valore ottenuto calcolando la somma del valore già memorizzato e 1

Assegnamento Computo dei Valori

- Le operazioni di assegnazione possono implicare calcoli complessi
 - Espressioni aritmetiche
 - Espressioni logiche e predicati

Espressioni Aritmetiche

- Formate da associazioni di variabili e costanti secondo regole opportune e attraverso l'applicazione di definiti operatori numerici

Simbolo	Tipo di valore cui dà luogo	Operazione
+	Numerico	Somma
-, ...	"	Differenza
*, ^, ...	"	Prodotto
/, ÷, DIV,	"	Divisione
***, ^, ...	"	Potenza

Espressioni Logiche o Predicati

- Usate in:
 - Assegnazioni fatte ad una variabile logica
 - Strutture di controllo che comportano la verifica di condizioni

Simbolo	Tipo di valore cui dà luogo	Operazione
NOT, AND, OR	Logico	Connettivi logici
=	"	Uguaglianza
≠	"	Diversità
<	"	Minoranza
>	"	Maggioranza
≤	"	Minore o uguale
≥	"	Maggiore o uguale

Costanti

- Dati il cui valore viene definito inizialmente e non varia per tutta l'esecuzione del programma
 - Accessibili solo in lettura
- Garanzia nell'uso
 - Impossibile eseguire assegnazioni sugli identificatori corrispondenti