

Corso di Programmazione

Programmazione

I parte

Dati – Istruzioni

Prof.ssa Teresa Roselli

`teresa.roselli@uniba.it`

Comunicazione dell'algoritmo all'elaboratore

- Linguaggi non ambigui e sequenziali
 - Comprensibili alla macchina
- Requisiti per la descrizione
 - Univoca
 - Non dà adito ad interpretazioni errate
 - Completa
 - Prevede tutte le azioni necessarie
 - Ripetibile
 - Garantisce un buon risultato se eseguita da più esecutori con medesime caratteristiche

Programmazione

- 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

Programmazione

- 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

Programmazione

- Metodologie
 - Astrazione
 - Oggetti
 - Azioni
- Tecniche
- Strumenti
 - Linguaggi
 - Ambienti

Programma

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

Programma

- 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)
- Algoritmi + Strutture Dati = Programmi
 - [Wirth]

Programma

- 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 (2+3)
 - Usa rappresentazioni simboliche per estendere l'applicabilità del metodo di soluzione a valori diversi
 - Uso di *variabili* (x+y)

Dati

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

Tipi di Istruzioni

- Un linguaggio di programmazione deve poter disporre 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

Tipi di Istruzioni

- Istruzioni dichiarative
 - Consentono di definire come rappresentare le entità del problema in termini di variabili nel programma
 - Totale caratterizzazione attraverso la definizione di:
 - Un nome
 - Un tipo
- Strutture di controllo

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

Istruzioni di Ingresso/Uscita

- A livello di descrizione dell'algoritmo
 - Soddisfano la necessità di indicare i dati su cui operare
- A livello di programma
 - Soddisfano la necessità di comunicare i dati e i risultati
 - Istruzioni di lettura e scrittura

Istruzioni Dichiarative

- Forniscono una lista contenente i nomi scelti per le variabili e i tipi corrispondenti
 - Convenzione: indicare *tutte* le variabili
- Necessarie nei linguaggi di programmazione

Variabile

- Identificatore di variabile: nome simbolico per denotare un'area di memoria
- Individua un oggetto su cui l'algoritmo opera
- La memorizzazione di un valore nell'indirizzo di memoria associato ad una variabile avviene secondo uno dei seguenti modi
 - Istruzioni di ingresso
 - Istruzioni di assegnamento

Variabile

- 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 di variabile 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

Variabile

- Caratterizzata da
 - Nome
 - *Identificatore*
 - Indirizzo
 - Valore
 - Tipo
 - Attributo che specifica l'insieme di valori che la variabile può assumere

Istruzioni di Ingresso

- Permettono di acquisire informazioni dall'esterno che vengono inserite nelle locazioni di memoria delle variabili dichiarate nel programma
- Attivano un'operazione di lettura
 - Assegnazione del valore letto da un supporto esterno
 - tastiera, dischi magnetici, dischi ottici ...
 - ad un'area di memoria individuata dal nome che compare nell'istruzione di lettura
- Esempio: read x

Assegnazione di valori a variabili

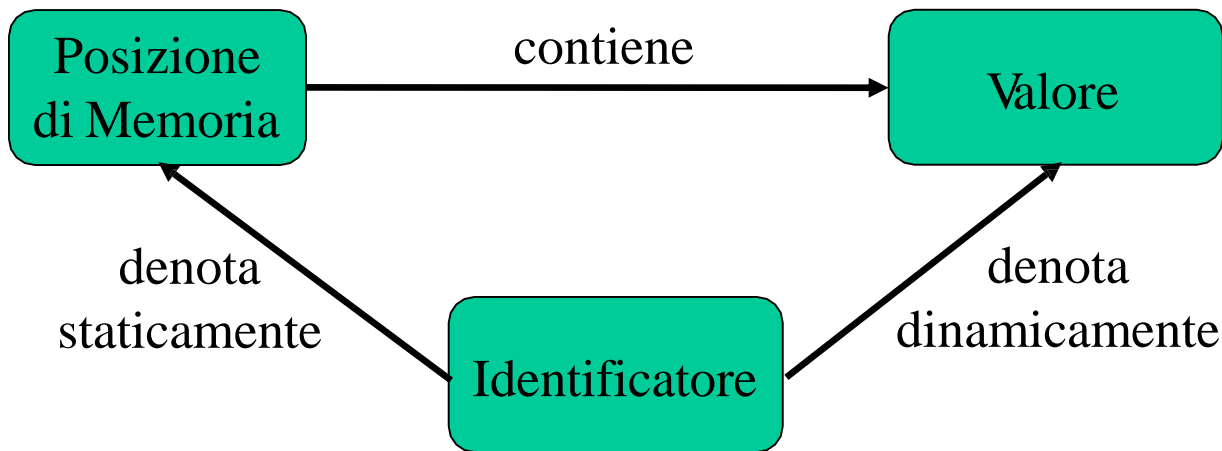
- Avviene in 2 fasi
 - Produzione di un nuovo valore
 - Assegnamento di quel valore alla variabile
- L'operazione di assegnazione viene indicata con $\leftarrow :=$ a seconda dei diversi linguaggi di programmazione
 - Confusione fra uguaglianza e assegnazione

Assegnazione

- 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 nella locazione di memoria riservata alla variabile a sinistra
 - La memorizzazione del valore ottenuto nella locazione di memoria riservata alla variabile implicata nell'assegnamento cancella qualunque valore contenuto in precedenza

Legami degli Identificatori

- Identificatore – Posizione di memoria
 - Legame statico
- Identificatore – Valore
 - Legame dinamico nel programma



Legami degli Identificatori

- La dinamicità del legame identificatore-valore consente di scrivere senza contraddizioni assegnazioni del tipo

$x \leftarrow x + 1$ (non è una uguaglianza)

- Alla posizione di ~~mem~~ *memoria* identificata da x assegna il valore ottenuto calcolando la somma del valore già memorizzato e 1

x 10

x 11

$x = x + 1$

↖ indirizzo ↗ valore

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, display, dischi magnetici ...
 - del contenuto di un'area di memoria denotata dal nome della variabile che compare nell'istruzione di scrittura
- Esempio: write y

Assegnazione

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
**, ↑, ...	“	Elevamento a potenza

Espressioni Logiche o Predicati

- Usate in:
 - Assegnazioni che riguardano 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

Strutture di Controllo

- Obbligatorie
 - Sequenza
 - Selezione binaria
 - Una Iterazione illimitata
- Opzionali
 - Selezione multipla
 - L'altra iterazione illimitata
 - Iterazione limitata