

LABORATORIO DI PROGRAMMAZIONE
Corso di laurea in matematica

7 – VARIABILI E COSTANTI
 Dipartimento di Matematica e Applicazioni
 Università degli Studi di Napoli Federico II

wpage.unina.it/lapegna

1

Marco Lapegna
 Laboratorio di Programmazione
 7 – Variabili e costanti

dalla prima lezione

Concetto di algoritmo

Sequenza di istruzioni non ambigue che in un numero finito di passi risolvono una classe di problemi

Un primo linguaggio: il flow chart

- Ovale (inizio e fine)
- Rettangolo (comandi)
- Rombo (verifica condizione)

2

Marco Lapegna
 Laboratorio di Programmazione
 7 – Variabili e costanti

esempio: calcolo della circonferenza

Si vuole **calcolare la circonferenza del cerchio** per molteplici valori del raggio:

- **Raggio = 1**
 $circonferenza = 2 \pi 1$
- **Raggio = 2.5**
 $circonferenza = 2 \pi 2.5$
- **Raggio = 5**
 $circonferenza = 2 \pi 5$

Alcuni dati sono **costanti** (2 e π)
 Altri dati sono **variabili** (raggio e circonferenza)

raggio

↓

algoritmo

↓

circonferenza

3

Marco Lapegna
 Laboratorio di Programmazione
 7 – Variabili e costanti

input e output in flow chart

E' necessario introdurre due nuovi simboli

- **Input:** scheda perforata

Il numero dato in input e' associato al nome riportato all'interno del simbolo

- **Output:** tabulato di stampante

Il valore associato al nome riportato all'interno del simbolo e' reso disponibile all'esterno

4

Marco Lapegna
Laboratorio di Programmazione
7 - Variabili e costanti

uno pseudo linguaggio per gli algoritmi

Alcuni svantaggi del flow-chart

- poco adatto alla descrizione degli algoritmi, perché molto diverso dalla struttura dei linguaggi di programmazione
- Tende a produrre algoritmi in maniera disordinata e poco leggibile

Il **pascal-like** è un **linguaggio per la descrizione degli algoritmi** (anche chiamato pseudo linguaggio) derivato dal Pascal, e costringe a sviluppare algoritmi in maniera strutturata

```

graph TD
    A([inizio]) --> B[read R]
    B --> C["C = 2 * pi * R"]
    C --> D[print C]
    D --> E([fine])
            
```

begin circonferenza

read R

$C = 2 * \pi * R$

print C

end circonferenza

5

Marco Lapegna
Laboratorio di Programmazione
7 - Variabili e costanti

le variabili

Nell'algoritmo precedente

- I valori di 2 e π sono **costanti** e definiti per ogni esecuzione
- I valori di C e R non sono definiti e rappresentano quantità **variabili** da esecuzione a esecuzione

C e R indicano generici numeri reali

variabili di tipo reale

6

Marco Lapegna
Laboratorio di Programmazione
7 - Variabili e costanti

istruzioni read e print

Dal punto di vista pratico, una **variabile** è il **nome dato ad una locazione di memoria**

L'istruzione **read R** legge un valore dall'unità di input e lo **assegna alla variabile R**

L'istruzione **print C** visualizza sull'unità di output il **contenuto della variabile C**

memoria

read R → 5 R

memoria

R 5

C 31.4 → print C → 31.4

7

Marco Lapegna
Laboratorio di Programmazione
7 - Variabili e costanti

assegnazione

L'istruzione

$C = 2 * \pi * R$

- **Preleva** dalla memoria il contenuto della variabile R
- Lo **moltiplica** per 2π
- **Memorizza** il risultato nella variabile C

memoria

R 5

C 31.4

ALU

$2 * \pi * 5$

Osservazioni

- Il segno "=" è una operazione di **assegnazione**
- La variabile R deve essere **definita** (cioè avere un valore)
- L'assegnazione è sempre da destra verso sinistra

8

Marco Lapegna
Laboratorio di Programmazione
7 - Variabili e costanti

dichiarazione delle variabili

Dal punto di vista pratico, una **variabile** e' il **nome dato ad una locazione di memoria**

Poiche' **la sequenza di bit** presente nella locazione di memoria **ha significato diverso** a secondo del tipo di dato contenuto

Esempi:

var R, C : real (R e C variabili reale)
var N : integer (N variabile intera)
var nome : character (nome variabile carattere)
var L : logical (L variabile logica)

In pascal-like e' necessario dichiarare il tipo delle variabili

R

0	1	0	1	0	0	0	0
---	---	---	---	---	---	---	---

Se R integer
R=80 (rappr. binaria)

Se R character
R=P (rappr. ASCII)

Se R real
R=0.625 (rappr. reale con
- 5 cifre mantisse
- 3 esponente)

9

Marco Lapegna
Laboratorio di Programmazione
7 - Variabili e costanti

calcolo della circonferenza (seconda versione)

La dichiarazione delle variabili consente al calcolatore **di interpretare correttamente il contenuto delle variabili**

La dichiarazione delle variabili **va fatta prima dell'utilizzo** delle variabili stesse

Si ottiene quindi il seguente algoritmo

```
begin circonferenza
  var R, C: real
  read R
  C = 2 * pi * R
  print C
end circonferenza
```

10

Marco Lapegna
Laboratorio di Programmazione
7 - Variabili e costanti

le costanti

In una istruzione del tipo " R = P " come distinguere

- Assegna a R il **contenuto della variabile P**
- Assegna a R il **carattere P**

Analogamente, nell'istruzione " C = 6 ", come distinguere

- Assegna a C il **numero intero 6**
- Assegna a C il **carattere 6**

Necessita' di definire un modo per **usare i diversi tipi di costanti** (intero, reale, carattere, logico)

11

Marco Lapegna
Laboratorio di Programmazione
7 - Variabili e costanti

le costanti

- Le costanti di **tipo carattere** sono racchiuse tra una **coppia di apici**
Esempio: 'casa' , 'Napoli' , 'Maria'
- Le costanti di **tipo logico** sono racchiuse tra una **coppia di punti**
Esempio: .true. , .false.
- Le costanti di **tipo intero** sono **sequenze di cifre** eventualmente con segno
Esempio: +80 , -45 , 981
- Le costanti di **tipo reale** sono **sequenze di cifre con il punto decimale**
Esempio: 45.23 , 0.0032 , -23.12

Per il tipo reale e' possibile utilizzare anche la notazione esponenziale
Esempio: 12.3e5 , 0.e0 , 0.1e-6

12

Marco Lapegna
Laboratorio di Programmazione
7 - Variabili e costanti

calcolo della circonferenza (terza versione)

Nell'algoritmo per il calcolo della circonferenza il valore del simbolo (costante) π e' ambiguo

Specificando il valore come una costante di tipo reale si ottiene la versione finale dell'algoritmo

```

begin circonferenza
  var R, C: real
  read R
  C = 2 * 3.1415926 * R
  print C
end circonferenza

```

13

Marco Lapegna
Laboratorio di Programmazione
7 - Variabili e costanti

definizione delle variabili

Definizione = Assegnazione di un valore ad una variabile

Puo' avvenire in due modi

- Tramite **lettura** (istruzione read)
- Tramite **assegnazione** (operatore =)

Utilizzando una ulteriore variabile (PG), l'algoritmo per il calcolo della circonferenza puo' essere riscritto come riportato

```

begin circonferenza
  var R, C, PG: real
  read R
  PG = 3.1415926
  C = 2 * PG * R
  print C
end circonferenza

```

14

Marco Lapegna
Laboratorio di Programmazione
7 - Variabili e costanti

alcune regole di base

Tutte le variabili devono essere **definite prima di essere utilizzate**

Il valore dell'espressione deve essere dello **stesso tipo di quello della variabile a cui e' assegnato**

```

begin circonferenza
  var R, C, PG : real
  read R
  C = 2 * PG * R
  print C
end circonferenza

```

ERRORE !! : La variabile PG non e' definita

```

begin circonferenza
  var R, C : real
  var PG : integer
  read R
  PG = 3.141592
  C = 2 * PG * R
  print C
end circonferenza

```

ERRORE !! : La variabile PG non e' usata in modo corretto

15

Marco Lapegna
Laboratorio di Programmazione
7 - Variabili e costanti

esempio: area del trapezio

Osservazione:
Nella definizione di una variabile, **prima** si valuta l'espressione al secondo membro, e **dopo** si assegna il valore ottenuto alla variabile al primo membro

Una variabile puo' essere **utilizzata piu' volte** ridefinendo il suo valore nel corso dell'algoritmo

Esempio:
Algoritmo per il calcolo dell'area di un trapezio

Dati di input: Bmag, Bmin, H
Dati di output: Area

```

begin trapezio
  var Bmag, Bmin : real
  var H, Area : real
  read Bmag, Bmin
  read H
  Area = Bmag + Bmin
  Area = Area * H
  Area = Area / 2
  print Area
end trapezio

```

16

Marco Lapegna
Laboratorio di Programmazione
7 - Variabili e costanti

esecuzione dell'algoritmo (1)

Calcolo dell'area di un trapezio con
Bmag = 10 , Bmin= 8 , H = 3
(Area = 27)

Fig.1: esecuzione di
`read Bmag, Bmin`

Fig.2: esecuzione di
`read H`

17

Marco Lapegna
Laboratorio di Programmazione
7 - Variabili e costanti

esecuzione dell'algoritmo (2)

Fig.3: esecuzione di
`Area = Bmag + Bmin`

Fig.4: esecuzione di
`Area = Area * H`

18

Marco Lapegna
Laboratorio di Programmazione
7 - Variabili e costanti

esecuzione dell'algoritmo (3)

Fig.5: esecuzione di
`Area = Area / 2`

Fig.6: esecuzione di
`print Area`

19

Marco Lapegna
Laboratorio di Programmazione
7 - Variabili e costanti

riassumendo

- Il pascal-like e' uno pseudo linguaggio per la descrizione degli algoritmi
- Le variabili sono nomi dati alle locazioni di memoria, i cui valori possono essere modificati durante l'algoritmo attraverso il loro nome
 - Le variabili vanno dichiarate
 - Le variabili vanno definite prima di essere usate

**Sono elementi universali degli algoritmi
presenti in tutti i linguaggi di programmazione**

20

Marco Lapegna
Laboratorio di Programmazione
7 - Variabili e costanti

un po' di storia (7)

Donald Knuth (1938)

- Laureato in matematica negli Stati Uniti, e' professore universitario a Stanford.
- Pone le basi dello sviluppo dell'analisi rigorosa della complessita' computazionale degli algoritmi e sistemato le tecniche matematiche per la verifica della loro correttezza
- E' l'autore di "the art of computer programming", un'opera in piu' volumi, ancora non tutti pubblicati, considerata la "Bibbia" degli informatici.
- E' autore anche del TeX, un sistema di composizione tipografica divenuto oggi uno standard per la pubblicazione di articoli scientifici.
- Vince il premio Turing nel 1974 per i suoi contributi all'analisi degli algoritmi e per il disegno dei linguaggi di programmazione



Donald Knuth nel 1988



Una edizione di "The Art of Computer Programming"