Programmazione Procedurale

Strutture di Controllo
Istruzioni Iterative (Cicli)

versione 4.0

1

Sommario

- Introduzione
- Tipologie di Ciclo
- Istruzione WHILE in C++
- Istruzione FOR in C++
- Nidificazioni
- Esercizi

2

Introduzione

- In molti casi
  - ⇨è necessario ripetere più volte in un programma le stesse operazioni su dati diversi
  - ⇨es: lettura dei dati di molti studenti
  - ⇨sarebbe opportuno evitare di duplicare le istruzioni
- Istruzioni iterative
  - ⇨servono esattamente a questo scopo

3

Introduzione

- Istruzioni Iterative o “Cicli”
  - ⇨consentono di ripetere l’esecuzione di una porzione di programma più volte
  - ⇨l’esecuzione procede fino a quando una condizione opportuna è verificata
  - ⇨quando la condizione non è più verificata, l’esecuzione si interrompe e si procede con le istruzioni successive

4

Introduzione

- Parti fondamentali di un ciclo
  - ⇨"corpo del ciclo": istruzioni di cui ripetere l'esecuzione
  - ⇨"condizione del ciclo": condizione che regola l'esecuzione del ciclo
- Corpo del ciclo
  - ⇨istruzione semplice o blocco di istruzioni
- Condizione del ciclo
  - ⇨espressione a valori booleani

5

Introduzione

- Istruzioni di ciclo
  - ⇨due istruzioni disponibili nei linguaggi procedurali
  - ⇨Istruzione while ("ciclo while")
  - ⇨Istruzione for ("ciclo for")

6

Introduzione

ATTENZIONE
allo studio dei cicli

- ATTENZIONE
  - ⇨come affrontare lo studio dei cicli
- Ciclo while
  - ⇨semplice capire la semantica
  - ⇨difficile utilizzarlo correttamente in pratica
- Ciclo for
  - ⇨leggermente più articolata la semantica
  - ⇨semplice da utilizzare in pratica

7

Tipologie di Iterazione

- Prima di illustrare le due istruzioni
  - ⇨cerchiamo di capire che tipo di ripetizioni (iterazioni) sono normalmente richieste nelle applicazioni
- Tipologie di iterazione
  - ⇨“iterazione aperta”
  - ⇨“iterazione chiusa” (o “a conteggio”)
  - ⇨corrispondono ad esigenze diverse di programmazione

8

Tipologie di Iterazione

- Iterazione aperta
  - ⇨non è possibile stabilire a priori quante volte dovranno essere eseguite le operazioni da ripetere
  - ⇨il tutto dipende dal valore di una condizione
  - ⇨possono verificarsi 0 ripetizioni (la condizione non è verificata la prima volta)
  - ⇨oppure indefinite ripetizioni (la condizione è sempre verificata)

9

Tipologie di Iterazione

- Iterazione chiusa o a conteggio
  - ⇨è noto prima che l'iterazione cominci quante volte è necessario ripetere le operazioni
- Realizzazione pratica
  - ⇨attraverso una tecnica di conteggio
  - ⇨viene utilizzata una variabile intera per "contare" le ripetizioni (>>)

10

Tipologie di Iterazione

- Normalmente
  - ⇨il while si può usare sia per programmare iterazioni aperte che chiuse
  - ⇨il for solo per iterazioni chiuse
  - ⇨di conseguenza il while è un'istruzione più generale
- In effetti
  - ⇨in alcuni linguaggi la distinzione è rigida (es: FORTRAN)
  - ⇨in C++ non esiste una vera distinzione (>>)
  - ⇨ma ragioneremo alla maniera tradizionale

11

Istruzione WHILE

- Sintassi:

  `while (<condizione>) <corpo>`
- Dove
  - ⇨<condizione> è un'espressione booleana
  - ⇨<corpo> è un blocco di istruzioni
- Esempio:

```
int x; cin >> x;
while (x > 0) {
  cout << sqrt(x) << endl;
  cin >> x;
}
```

12

Istruzione WHILE

- Semantica
  - ⇨quando viene incontrato il ciclo, viene valutata la condizione
  - ⇨se la condizione viene trovata vera viene eseguito il corpo del ciclo
  - ⇨al termine dell'esecuzione del corpo, si torna in cima e si valuta nuovamente la condizione
  - ⇨se la condizione viene trovata falsa il ciclo si interrompe e si passa all'istr. successiva

13

Istruzione WHILE

- Supponiamo
  - ⇨che l'utente digiti la sequenza 25 9 144 0
- In questo caso
  - ⇨verranno eseguite le seguenti istruzioni

espansione del ciclo

```
int x;
cin >> x;
while (x > 0) {
  cout << sqrt(x) << endl;
  cin >> x;
}
```

>> Animazione

```
int x;
cin >> x; // 25
// x > 0 ? TRUE
cout << sqrt(x) << endl;
cin >> x; // 9
// x > 0 ? TRUE
cout << sqrt(x) << endl;
cin >> x; // 144
// x > 0 ? TRUE
cout << sqrt(x) << endl;
cin >> x; // 0
// x > 0 ? FALSE
```

14

Istruzione WHILE

- Numero di esecuzioni del corpo
  - ⇨ci sono casi in cui il corpo del ciclo non viene mai eseguito; es: digito subito 0
  - ⇨ci sono casi in cui l'esecuzione può andare avanti indefinitamente; es: non digito mai 0
  - ⇨possono esserci situazioni di errore
- Condizioni di ciclo indefinito ("loop")
  - ⇨ciclo scritto in modo tale che l'esecuzione va sempre avanti indefinitamente

16

Istruzione WHILE

- Esempio di loop
  - ⇨la condizione è sempre verificata

```
int x;
x = 1;
while ( x > 0 ) {
    cout << sqrt(x) << endl;
    x = x + 1;
}
cout << "Fine";
```

la condizione non diventa mai falsa perchè il valore di x aumenta

>> loop.cpp

17

Istruzione WHILE

- Con il while
  - ⇨si possono scrivere anche iterazioni chiuse
- Esempio di iterazione chiusa
  - ⇨leggere la lunghezza del lato di 5 quadrati e calcolare e stampare l'area di ciascuno
  - ⇨in questo caso l'operazione da ripetere (calcolo e stampa dell'area del quadrato) deve essere ripetuta esattamente 5 volte (ogni volta su un valore diverso)

18

Conteggio con Contatore

- Problema frequente
  - ⇨conteggio di un gruppo di oggetti
  - ⇨esempio: contare il numero degli esami
- Tecnica algoritmica notevole
  - ⇨utilizzo di una variabile intera (es: i, conta...) all'interno di un ciclo
  - ⇨la variabile svolge le funzioni di "contatore"
  - ⇨valore iniziale pari a 0
  - ⇨incrementata ad ogni nuovo elemento

19

Conteggio con Contatore

- Metafora
  - ⇨la variabile contatore è analoga ad un "contatore meccanico"
- Tre operazioni
  - ⇨reset del conteggio
  - ⇨incremento del conteggio
  - ⇨lettura del valore contato

20

Tipologie di Iterazione

- Iterazione a conteggio
  - ⇨viene usata una variabile contatore per contare il numero di esecuzioni del ciclo
  - ⇨vengono fissati (a priori) un valore iniziale ed il valore finale del contatore
  - ⇨il valore della variabile viene fatto crescere ad ogni ripetizione
  - ⇨finché la variabile non raggiunge il valore massimo

21

Istruzione WHILE

- Tecnica standard per realizzarlo
  - ⇨uso una variabile intera (es: i) come contatore
  - ⇨inizialmente i = 0
  - ⇨ad ogni esecuzione i = i + 1
  - ⇨mi fermo quando i raggiunge il valore finale
- Variabile di controllo del ciclo
  - ⇨il valore di i mi dice in ogni istante quante volte è stato già eseguito il corpo del ciclo

22

Istruzione WHILE

3 elementi fondamentali
-valore iniziale di i
-condizione sul valore finale
-istruzione di incremento

```
#include <iostream>
using namespace std;
int main() {
    float lato, area;
    int i;
    i = 0;
    while (i < 5){
      cout << "Immetti la lung. del lato" << endl;
      cin >> lato;
      area = lato * lato;
      cout << "Area quadrato: " << area << endl;
      i = i + 1;
    }
    cout << "Fine";
    return 0;
}
```

23

Esercizi WHILE

- Iterazione aperta
  - ⇨Calcolo della media dei voti
    - ⇨Soluzioni esercizi 05, 06 (con guardie)
  - ⇨Calcolo peso forma con guardie
    - ⇨Soluzione 07
- Iterazione chiusa
  - ⇨Calcolo della media di 10 valori casuali compresi tra 1 e 100
    - ⇨Soluzione esercizio 08

24

Istruzione FOR

- Questo tipo di ciclo è molto frequente
- Per rendere più facile la struttura
  - ⇨esiste un'istruzione apposita
  - ⇨l'istruzione FOR
  - ⇨consente di scrivere in modo più compatto il ciclo, ed in particolare i 3 elementi fondamentali

25

Istruzione FOR

I 3 elementi fondamentali sono tutti sulla stessa riga
-valore iniziale di i
-condizione sul valore finale
-istruzione di incremento

```
#include <iostream>
using namespace std;
int main() {
     float lato, area;
     int i;
     for ( i=0 ; i<5 ; i=i+1 ) {
       cout << "Immetti la lung. del lato" << endl;
       cin >> lato;
       area=lato*lato;
       cout << "Area quadrato: " << area << endl;
     }
     cout << "Fine";
     return 0;
}
```

26

Istruzione FOR

```
float lato, area;
int i;
for (i=0; i<5; i++) {
   cout << "Lato ? "
        << endl;
   cin >> lato;
   area=lato*lato;
   cout << "Area: "
        << area
        << endl;
}
cout << "Fine";
```

```
float lato, area;
int i;
i=0;
while (i<5) {
   cout << "Lato ? "
        << endl;
   cin >> lato;
   area=lato*lato;
   cout << "Area: "
        << area
        << endl;
   i++;
}
cout << "Fine";
```

27

Istruzione FOR

- Sintassi:

```
for (<e1>; <e2>; <e3>) <corpo>
```

- Dove
  - ⇨<e1> è l'inizializzazione della variabile
  - ⇨<e2> è la condizione del ciclo
  - ⇨<e3> è l'incremento della variabile
  - ⇨<corpo> è un blocco di istruzioni

28

Istruzione FOR

- Semantica: è equivalente ad un while
- In particolare

```
for (<e1>; <e2>; <e3>) <corpo>
```

equivale a

```
<e1>;
while (<e2>) {
  <corpo>
  <e3>
}
```

29

Istruzione FOR

>> Animazione

- Più in dettaglio
  - ⇨viene eseguita l'inizializzazione (e1)
  - ⇨viene verificata la condizione (e2)
  - ⇨se la condizione viene valutata come vera
    - ⇨viene eseguito il corpo
    - ⇨al termine viene effettuato l'incremento (e3)
    - ⇨viene valutata di nuovo la condizione
  - ⇨se la condizione viene valutata come falsa
    - ⇨il ciclo termina e si passa all'istruzione succ.

30

Istruzione FOR

- ATTENZIONE
  - ⇨la variabile del ciclo serve per contare
  - ⇨di conseguenza normalmente è una variabile intera (tipo int)
  - ⇨in alcuni linguaggi questo è obbligatorio (ma in C++ può anche essere float o double)
  - ⇨molto spesso la chiameremo "i" (ma il nome può essere qualsiasi)

32

Nidificazione

- Nel corpo di un ciclo
  - ⇨qualsiasi istruzione
- In particolare
  - ⇨altri cicli, nidificazione arbitraria
- In generale
  - ⇨le strutture di controllo possono essere nidificate liberamente
- Esempio: if in un for in un while in un else

33

Nidificazione

- Vale la stessa semantica
  - ⇨con qualche accortezza per i cicli
  - ⇨i cicli interni ricominciano e vengono eseguiti per ogni esecuzione del ciclo esterno
- Esempio

```
int i, j;
for (i = 0; i < 5; i++) {
    for (j = 0; j < 10; j++) {
        cout << "Eseguito";
    }
}
```

ciclo esterno

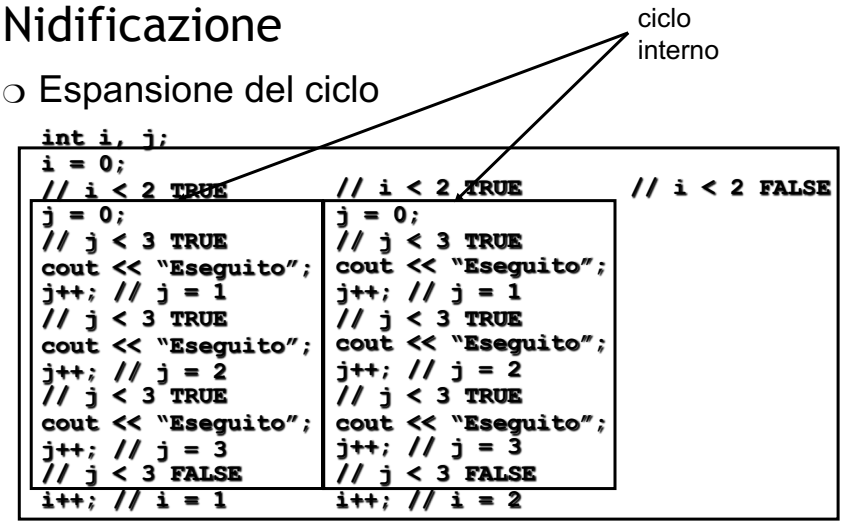
corpo del ciclo esterno (ciclo interno)

corpo del ciclo interno eseguito 50 volte

34

Nidificazione

- Espansione del ciclo



Esempio

- Stampare sullo schermo la tabella pitagorica 5x5

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

Esempio

- Analisi del problema
  - ⇨il problema non ha dati di ingresso
- Algoritmo
  - ⇨la tabella è divisa per righe e colonne
  - ⇨ogni elemento è pari al valore dell'indice di riga (i) per l'indice di colonna (j)
  - ⇨si tratta di calcolare e stampare 25 moltiplicazioni

37

Esempio

- Algoritmo (continua)
  - ⇨possiamo utilizzare due cicli for uno nell'altro
  - ⇨il ciclo esterno serve a ripetere per 5 volte le operazioni necessarie per stampare una riga
  - ⇨il ciclo interno serve, per ciascuna riga, a ripetere 5 volte le operazioni necessarie a stampare un elemento
  - ⇨al termine di ciascuna riga è necessario andare a capo

>> pitagoricaFor.cpp

38

Esempio

```
# include <iostream>
using namespace std;
int main (){
    const int N = 5;
    int i, j;
    for(i = 1; i <= N; i++) {
        for(j = 1; j <= N; j++) {
            cout << i * j << " ";
        }
        cout << endl;
    }
    cout << "Fine";
    return 0;
}
```

>>

39

Esempio

- Sono possibili altre soluzioni
  - ⇨es: due while uno nell'altro
  - ⇨semantica equivalente
- Ma anche
  - ⇨un for esterno con un while interno
  - ⇨un while esterno con un for interno
- Il comportamento è sempre lo stesso

>> pitagoricaWhile.cpp

41

Soluzione con While

```
# include <iostream>
using namespace std;
int main (){
    const int n=5;
    int i, j;
    i=1;
    while(i<=n) {
        j=1;
        while(j<=n) {
            cout << i*j << " ";
            j++;
        }
        cout << endl;
        i++;
    }
    cout << "Fine";
    return 0;
}
```

>>

42

Esempio

- Attenzione
  - ⇨è molto più semplice, nel caso del while, commettere errori
- Errore frequente
  - ⇨omettere l'istruzione di incremento della variabile di conteggio
  - ⇨in questo caso si genera un "loop"
- In generale
  - ⇨per i cicli a conteggio meglio il for

43

Riassumendo

- Due tipi di cicli
  - ⇨cicli while per iterazioni “aperte” o “chiuse”
  - ⇨cicli for per iterazioni “chiuse” (“a conteggio”)
- In C++
  - ⇨il for è una forma particolare di while
- Errori frequenti nell’uso dei cicli
  - ⇨cicli nidificati (ATTENZIONE)
  - ⇨loop (ATTENZIONE)

44

Esercizi

- Esercizio n.1
  - ⇨simulare il funzionamento dei due programmi per il calcolo della media
- Esercizio n.2
  - ⇨scrivere un programma che legge dalla tastiera 7 numeri complessi
  - ⇨calcola e stampa il modulo di ciascun numero
- Esercizio n.3
  - ⇨come il precedente, ma il programma deve andare avanti finchè non viene inserito il numero 0

45

Ringraziamenti

- Parte del materiale di questa lezione è stato sviluppato con la collaborazione della Dott.ssa Maria Samela (mariasamela@tiscali.it), a cui va il mio ringraziamento.

46

Termini della Licenza

47

Istruzione WHILE

- Attenzione alla semantica
  - ⇨la condizione viene verificata solo all'inizio e al termine di ogni esecuzione del corpo
  - ⇨non viene controllata continuamente
  - ⇨ci possono essere istanti in cui, durante l'esecuzione del corpo, la condizione non è verificata, ma questo non basta ad interrompere il ciclo