Programmazione Procedurale

Strutture di Controllo
Tecniche Algoritmiche

versione 3.0

1

Sommario

- Tecniche Algoritmiche Notevoli
- Conteggio con Contatore
- Somma con Accumulatore
- Convalida dei Dati in Ingresso
- Programmazione non Strutturata
  - ⇨Variabili Bandiera (“flag”)

2

Tecniche Algoritmiche Notevoli

- soluzioni standard a problemi ricorrenti di programmazione
- rappresentano un bagaglio di conoscenze riutilizzabili per il programmatore

5

Tecniche Algoritmiche Notevoli

- Tecniche notevoli nei due progetti
  - ⇨somma con variabile “accumulatore” >> somma dei voti
  - ⇨conteggio con variabile “contatore” >> conteggio del numero di esami
  - ⇨cicli di convalida >> verifica dei dati forniti dall’utente
- Inoltre, vedremo anche
  - ⇨utilizzo di una variabile “bandiera” e programmazione non strutturata

6

Conteggio con Contatore

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

7

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

8

Un Esempio: Media con While

```
#include <iostream>
using namespace std;
int main() {
    int voto, sommaVoti, numeroEsami;
    float media;
    bool continua;
    continua = true;
    numeroEsami = 0;
    sommaVoti = 0;
    while (continua == true) {
        cout << "Immetti il voto (0 per fermarti)" << endl;
        cin >> voto;
        if (voto == 0) {
            continua = false;
        } else {
            numeroEsami++;
            sommaVoti += voto;
        }
    }
    if (numeroEsami > 0) {
        media = ((float)sommaVoti) / numeroEsami;
        cout << "La media vale: " << media << endl;
    } else {
        cout << "Non hai fornito nessun voto" << endl;
    }
    return 0;
}
```

9

Conteggio con Contatore

- Attenzione
  - ⇨abbiamo già visto questa tecnica
- Cicli a conteggio
  - ⇨è la stessa tecnica che viene utilizzata per controllare i cicli a conteggio
  - ⇨anche in quel caso viene utilizzato un contatore (per contare il numero di esecuzioni del ciclo)

10

Un Esempio Già Visto: I 5 Quadrati

```
#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++;
    }
    cout << "Fine";
    return 0;
}
```

11

Somma con Accumulatore

- Problema frequente
  - ⇨calcolare la somma dei valori di una collezione
  - ⇨es: somma dei voti
- Tecnica algoritmica
  - ⇨utilizzo una variabile intera all'interno di un ciclo come "accumulatore"
  - ⇨inizialmente la variabile vale 0
  - ⇨sommo ogni nuovo valore al valore dell'accumulatore fino a questo punto

12

Somma con Accumulatore

- Metafora
  - ⇨la brocca graduata
  - ⇨date varie bottiglie di acqua di varia dimensione
  - ⇨vogliamo calcolare il volume complessivo di acqua
  - ⇨verso ogni bottiglia nella brocca graduata
  - ⇨alla fine leggo il volume nella brocca

13

Somma con Accumulatore

```
1.  float voto, sommaVoti;
2.  int numeroEsami, i;
3.  cout << "Quanti esami ? ";
4.  cin >> numeroEsami;
5.  sommaVoti = 0.0;
6.  for (i = 0; i < numeroEsami; i++)
7.      cout << "Voto ? ";
8.      cin >> voto;
9.      sommaVoti = sommaVoti + voto;
10. }
```

Memoria RAM

#2500	numeroEsami	3
#2501	sommaVoti	67.0
#2502	i	3
#2503	voto	24

Schermo

```
Quanti esami ?  3
Voto ?  21
Voto ?  22
Voto ?  24
```

14

Convalida dell'Input

mediaForConvalida.cpp >>
mediaWhileConvalida.cpp >>

- I due programmi acquisiscono dati dall'utente
  - ⇨sarebbe necessario verificare la correttezza dei valori forniti
  - ⇨es: voto tra 18 e 30
  - ⇨es: numero di esami >= 0
- Convalida dei valori in ingresso
  - ⇨è possibile utilizzare un ciclo while

15

Convalida dell'Input

- In particolare
  - ⇨il ciclo viene dopo una lettura;
    es: `cin >> numeroEsami;`
  - ⇨la condizione del ciclo è vera nel caso in cui il valore letto sia scorretto
    es: `while (numeroEsami < 0) {...}`
  - ⇨se viene letto un valore corretto, il ciclo di convalida viene ignorato
  - ⇨se viene letto un valore scorretto, il ciclo inizia, e va avanti finchè non viene fornito un valore corretto

16

Programmazione Non Strutturata

- Problema frequente
  - ⇨controllare l'esecuzione di un ciclo aperto
  - ⇨ovvero fermare il ciclo quando non deve essere più eseguito
- Vediamo due soluzioni
  - ⇨a. utilizzo di una variabile bandiera ("flag")
  - ⇨b. utilizzo della programmazione non strutturata

17

Variabile Bandiera ("Flag")

- Tecnica algoritmica
  - ⇨viene utilizzata una variabile booleana
  - ⇨inizializzata a true
  - ⇨il ciclo prosegue finchè la var. è vera (la var. è la bandiera che da via libera al ciclo)
  - ⇨ad ogni nuovo voto si controlla se l'utente vuole interrompere; in questo caso la variabile diventa falsa)
- Tecnica notevole per il controllo di cicli

18

Variabile Bandiera ("Flag")

- Metafora
  - ⇨la variabile booleana viene usata come la bandiera di un gran premio
  - ⇨viene sventolata inizialmente per avviare la corsa (valore "true")
  - ⇨viene sventolata alla fine per fermare la corsa (valore "false")

19

Un Esempio: Media con While

```
#include <iostream> using namespace std;
int main() {
    int voto, sommaVoti, numeroEsami;
    float media;
    bool continua;
    continua = true;
    numeroEsami = 0;
    sommaVoti = 0;
    while (continua == true) {
        cout << "Immetti il voto (0 per fermarti)" << endl;
        cin >> voto;
        if (voto == 0) {
            continua = false;
        } else {
            numeroEsami++;
            sommaVoti += voto;
        }
    }
    if (numeroEsami > 0) {
        media = ((float)sommaVoti) / numeroEsami;
        cout << "La media vale: " << media << endl;
    } else {
        cout << "Non hai fornito nessun voto" << endl;
    }
    return 0;
}
```

20

Variabile Bandiera ("Flag")

- Una annotazione
  - ⇨la condizione del ciclo while avrebbe potuto essere scritta anche come
    `while (continua) {...}`
  - ⇨infatti, le due espressioni,
    `(continua)` ← espressione fatta da un unico operando e nessun operatore
    `(continua==true)`
    sono equivalenti (vere se "continua" vale true, false se continua vale false")

21

Variabile Bandiera ("Flag")

- Più in generale
  - ⇨data una variabile b di tipo bool
  - ⇨dovunque sia necessario utilizzare un'espressione del tipo (b==true), è possibile utilizzare solo (b)
  - ⇨dovunque sia necessario utilizzare un'espressione del tipo (b==false) è possibile utilizzare solo `(!b)` ← vera se b vale false, falsa se b vale true

22

Programmazione Non Strutturata

- Fino ad ora
  - ⇨abbiamo visto una forma di programmazione che si chiama strutturata
- Programmazione strutturata
  - ⇨i blocchi (nelle strutture di controllo) vengono eseguiti del tutto o per niente (hanno un unico punto di ingresso ed un unico punto di uscita)
- A volte, però
  - ⇨questo stile non è conveniente

25

Programmazione Non Strutturata

- Per semplificare il codice
  - ⇨è possibile in alcuni casi utilizzare uno stile di "programmazione non strutturata"
- Programmazione non strutturata
  - ⇨i blocchi di istruzioni possono essere interrotti a metà
- Per farlo
  - ⇨due istruzioni nuove: break e continue

26

Istruzione BREAK

- Si usa nelle strutture di controllo
- Sintassi

```
break;
```

- Semantica
  - ⇨interrompe l'esecuzione della struttura di controllo (qualsiasi sia lo stato) per proseguire con l'istruzione successiva
- Esempio
  - ⇨può essere usata per interrompere for/while

27

Istruzione CONTINUE

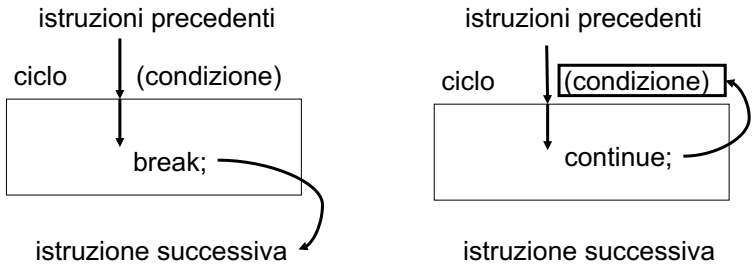
- Simile a break, ma si usa solo nei cicli
- Sintassi

```
continue;
```

- Semantica
  - ⇨interrompe l'esecuzione del corpo del ciclo (qualsiasi sia lo stato) per ritornare in cima e valutare di nuovo la condizione di uscita
- Differenza con break
  - ⇨non necessariamente termina il ciclo

28

Schema di Esecuzione del Ciclo



29

Programmazione Non Strutturata

- Esempio
  - ⇨la media dei voti in versione non strutturata
  - ⇨utilizziamo l'istruzione break per interrompere il ciclo quando l'utente inserisce il voto 0
  - ⇨rende inutile l'utilizzo del flag, e quindi il codice diventa più compatto
- Attenzione
  - ⇨il codice diventa più compatto
  - ⇨la utilizzeremo spesso, ma bisogna fare attenzione

30

Un Esempio: Media con While

>> mediaBreak.cpp

```
#include <iostream>
using namespace std;
int main() {
    int voto, sommaVoti = 0, numeroEsami = 0;
    float media;
    while (true) {
        cout << "Immetti il voto (0 per fermarti)" << endl;
        cin >> voto;
        if (voto == 0) {
            break;
        }
        numeroEsami++;
        sommaVoti += voto;
    }
    if (numeroEsami > 0) {
        media = ((float)sommaVoti) / numeroEsami;
        cout << "La media vale: " << media << endl;
    } else {
        cout << "Non hai fornito nessun voto" << endl;
    }
    return 0;
}
```

31

Programmazione Non Strutturata

- Attenzione alla condizione del ciclo
  - ⇨while (true) {…}
  - ⇨il ciclo andrebbe avanti per sempre
  - ⇨ma si interrompe a causa del break non appena l'utente digita il voto 0
- Errore grave
  - ⇨utilizzare un ciclo di questo tipo senza il break (il ciclo andrebbe avanti indefinitamente)

32

Programmazione Non Strutturata

- Riassumendo
  - ⇨break e continue introducono “salti” fuori dal flusso di esecuzione ordinario
- Avvertenza
  - ⇨possono semplificare il codice
  - ⇨ma è opportuno utilizzare questa funzionalità con attenzione

33

Riassumendo

- Tecniche Algoritmiche Notevoli (ATTENZIONE)
  - ⇨Conteggio con Contatore
  - ⇨Somma con Accumulatore
  - ⇨Convalida dei Dati in Ingresso
- Programmazione Non Strutturata
  - ⇨Istruzione BREAK e CONTINUE

34

Convalida dell'Input

```
cout << "Esami ?" << endl;
cin >> numeroEsami;

while (numeroEsami < 0) {
   cout << "Errore" << endl;
   cout << "Esami ?" << endl;
   cin >> numeroEsami;
}
```

Una soluzione alternativa:

```
bool continua;
continua = true;

while (continua) {
   cout << "Esami ?" << endl;
   cin >> numeroEsami;
   if (numeroEsami < 0) {
      cout << "Errore";
   } else {
      continua = false;
   }
}
```

Svantaggi di questa soluzione:
- il codice è meno compatto
- è molto meno leggibile

Utilizzeremo la prima

35

Termini della Licenza

36