

Strutture di controllo - Parte I

La selezione e l'iterazione

Corso AMAT C++

Liceo Scientifico Statale "A. Volta"

A.S. 2018/19



Sequenza

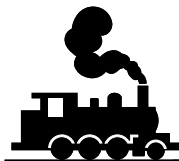
Fino a questo momento, un programma è stato per noi una sequenza di istruzioni, eseguite una in seguito all'altra da computer:

```
1 int main(){  
2     istruzione1;  
3     istruzione2;  
4     istruzione3;  
5     return 0;  
6 }
```

Sequenza

Fino a questo momento, un programma è stato per noi una sequenze di istruzioni, eseguite una in seguito all'altra da computer:

```
1 int main(){  
2     istruzione1;  
3     istruzione2;  
4     istruzione3;  
5     return 0;  
6 }
```



Interruzione

Se, dentro al *main*, diamo l'istruzione `return 0`, l'esecuzione del programma viene immediatamente terminata.

```
1 int main(){  
2     istruzione1;  
3     istruzione2;  
4     return 0;  
5     istruzione 3;  
6 }
```

Nell'esempio, le uniche istruzioni che vengono eseguite sono la prima e la seconda, dopodiché l'esecuzione si interrompe.

Interruzione

Se, dentro al *main*, diamo l'istruzione `return 0`, l'esecuzione del programma viene immediatamente terminata.

```
1 int main(){  
2     istruzione1;  
3     istruzione2;  
4     return 0;  
5     istruzione 3;  
6 }
```

Nell'esempio, le uniche istruzioni che vengono eseguite sono la prima e la seconda, dopodiché l'esecuzione si interrompe.



Selezione

```
1  int main(){
2      istr1;
3      istr2;
4      if(espressione_bool){
5          istr_ramo_vero_A;
6          istr_ramo_vero_B;
7      }else{
8          istr_ramo_falso_A;
9          istr_ramo_falso_B;
10         istr_ramo_falso_C;
11     }
12     istr3;
13     istr4;
14     return 0;
15 }
```

Semantica: il calcolatore esegue istr1 ed istr2. Poi, quando raggiunge la riga 4, calcola quanto vale espressione_bool:

- ▶ se ottiene vero, esegue le *istruzioni del ramo vero*; esaurito questo, l'esecuzione riprende con istruzione3, istruzione4, ...
- ▶ se ottiene falso, allora esegue le *istruzioni del ramo falso*; esaurito questo, l'esecuzione riprende con istr3, istr4, ...

Selezione

La *selezione* introduce una *biforcazione* nel flusso di esecuzione del codice. I due rami sono mutuamente esclusivi: uno ed uno solo di essi viene eseguito, a seconda del valore di verità che si ottiene valutando l'espressione `espressione_bool` (che viene anche chiamata *condizione*). A conclusione della selezione, la biforcazione si ricompone, e l'esecuzione riprende linearmente.



Selezione

```
1 int main(){
2     istr1;
3     istr2;
4     if(espressione_bool){
5         istr_ramo_vero_A;
6         istr_ramo_vero_B;
7     }else{
8     }
9     istr3;
10    istr4;
11    return 0;
12 }
```

```
1 int main(){
2     istr1;
3     istr2;
4     if(espressione_bool){
5         istr_ramo_vero_A;
6         istr_ramo_vero_B;
7     }
8     istr3;
9     istr4;
10    return 0;
11 }
```

Il ramo vero, così come il ramo falso, possono anche essere vuoti (cioè non contenere istruzioni). Quando il ramo falso è vuoto, si può omettere la clausola *else*: i due programmi scritti sopra sono equivalenti.

Iterazione

```
1  int main(){
2      istr1;
3      istr2;
4      while(espressione_bool){
5          istr_ciclo_A;
6          istr_ciclo_B;
7          istr_ciclo_C;
8          istr_ciclo_D;
9      }
10     istr3;
11     istr4;
12     return 0;
13 }
```

Semantica: il calcolatore esegue istr1 ed istr2. Poi, quando raggiunge la riga 4, calcola quanto vale espressione_bool:

- ▶ se ottiene falso, scavalca le istruzioni del ciclo e salta alla riga 10
- ▶ se ottiene vero, esegue le *istruzioni del ciclo*; ciò fatto, ritorna alla riga 4 (e siamo quindi daccapo: il computer calcola quanto vale espressione_bool, ecc. ecc.)

Iterazione

Le istruzioni del ciclo vengono rieseguite più e più volte, *finché*, valutando `espressione_bool` (che viene anche detta *condizione*), si ottiene vero. Quando, valutando `espressione_bool`, ottiene falso, il computer *esce dal ciclo*, e passa oltre. Ogni esecuzione delle istruzioni del ciclo si dice *iterazione*; ogni iterazione è preceduta da una valutazione di `espressione_bool` che ha restituito come risultato vero.

