

IL COSTRUTTO SELEZIONE



Il costrutto selezione

Permette di effettuare una scelta tra due possibili opzioni.

Tale scelta verrà effettuata valutando il valore di verità di una condizione espressa da una proposizione decidibile.

Una **proposizione o enunciato** è una espressione del linguaggio, cioè una sequenza di suoni con contenuto linguistico organizzati in parole e frasi, per la quale ha senso domandarsi se essa è vera o falsa.

Le proposizioni si indicano con lettere minuscole: p,q,

...

Ad ogni proposizione può essere associato un valore di verità (vero V o falso F).

Principio di identità: Ogni proposizione ha lo stesso valore di verità di se stessa.

Principio di non contraddizione: Una proposizione non può essere simultaneamente vera e falsa.

Principio del terzo escluso: Una proposizione non può che essere vera o falsa. Non esistono altri valori di verità.

Una proposizione si dice elementare se non può essere espressa dall'unione di proposizioni più semplici, altrimenti si dice composta.

Le proposizioni elementari si uniscono, a formare le proposizioni composte, mediante operatori detti connettivi.

Di un connettivo si deve sapere su quante proposizioni agisce e quale valore di verità restituisce a partire da quelli delle proposizioni di partenza.

Connettivi logici

Un **connettivo** è un operatore che consente di creare proposizioni composte a partire da quelle elementari.

Connettivi unari

Negazione

p	$\neg p$
VERO	FALSO
FALSO	VERO

In linguaggio C/C++
si indica con !

Connettivi binari

Congiunzione, \wedge , et, e, AND

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

In linguaggio C/C++
si indica con `&&`

Connettivi binari

Disgiunzione (non esclusiva), \vee , vel, o, OR

p	q	$p \vee q$
V	V	V
V	F	V
F	V	V
F	F	F

In linguaggio C/ C++
si indica con `||`

Il costrutto selezione

Permette di effettuare una scelta tra due possibili opzioni.

Tale scelta verrà effettuata valutando il valore di verità di una condizione espressa da una proposizione decidibile.

L'istruzione if

Nel linguaggio C/C++ il costrutto selezione si implementa tramite l'istruzione **if**, che ha la seguente sintassi:

if (condizione)

 istruzione1;

Viene eseguita se la condizione è VERA

else

 istruzione2;

Viene eseguita se la condizione è FALSA

ATTENZIONE: Non mettere mai il ; dopo la condizione, altrimenti le istruzioni non saranno eseguite.

La condizione

La condizione può essere:

- ❖ Una variabile singola \rightarrow if aritmetico,
- ❖ Una proposizione elementare, costituita dal confronto, mediante operatori relazionali, tra due variabili o due espressioni,
- ❖ Una proposizione articolata, costituita dalla connessione di proposizioni elementari mediante connettivi logici.

If aritmetico

La condizione è il valore numerico di una variabile o di una espressione.

UGUALE A 0
DIVERSO DA 0



FALSO
VERO

```
if (a)
```

```
    cout<< "Il numero è diverso da 0";
```

```
else
```

```
    cout<< "Il numero è uguale a 0";
```

If aritmetico

Dati due numeri verificare se essi sono uguali e stampare un messaggio.

```
if (a-b)
    cout << "I numeri sono diversi";
else
    cout << "I numeri sono uguali";
```

Gli operatori relazionali

Operatore relazionale	Significato
>	Maggiore di
<	Minore di
>=	Maggiore o uguale
<=	Minore o uguale
==	Uguale
!=	Diverso

ATTENZIONE: L'operatore di uguaglianza == si applica anche ai caratteri.

```
var == 'a';
```

Gli operatori relazionali

ATTENZIONE: L'operatore relazionale di uguaglianza si rappresenta con `==`.

L'=`=` semplice ha invece, il significato di assegnazione.

`if(a==b)` → controlla che il contenuto di a sia uguale al contenuto di b.

`if(a=b)` → assegna ad a il contenuto di b e poi controlla se è 0 o meno.

Operatori relazionali

Dato un numero calcolarne il cubo se il numero è maggiore di zero altrimenti calcolarne il quadrato.

```
if (a>0)
    b=pow(a,3);
else
    b=pow(a,2);

cout << b;
```

SUGGERIMENTO:
eseguire il test con
a=3, -3, 0, -1,5, 2,71

Le istruzioni

Le istruzioni da eseguire in caso di condizione VERA o FALSA, possono essere costituite da una singola istruzione o da un blocco di istruzioni. In questo ultimo caso esse devono essere racchiuse tra { }.

```
if (a>0)
    {b=pow(a,3);
    cout << b;}
else
    {b=pow(a,2);
    cout << b;}
```

SUGGERIMENTO:
E' sempre bene **indentare** i blocchi di istruzioni in modo da rendere il codice più leggibile.

Selezione ad una via

Talvolta il ramo else può anche essere omesso. In questo caso è come se fosse sottointeso che in caso di condizione falsa non viene eseguita nessuna istruzione.

Data l'età di una persona visualizzare se essa è maggiorenne.

```
if (eta >= 18)
    cout << "Maggiorenne";
```

SUGGERIMENTO:
eseguire il test con
eta = 5, 40, 17, 19, 18

Selezione a cascata

Si parla di selezione a cascata quando ci sono due o più selezioni in sequenza, le cui condizioni sono indipendenti.

Viene analizzata la prima condizione e, al termine della prima selezione, si analizza la seconda condizione.

Selezione a cascata

Dato il prezzo di un prodotto calcola e stampa il prezzo da pagare sapendo che se il prezzo è superiore a 3000€ si applica lo sconto del 6% e se il pagamento è in contanti si applica uno sconto del 2% .

```
if (prezzo>3000)
    prezzo = prezzo-prezzo*6/100;
if(pagamento=='c')
    prezzo = prezzo-prezzo*2/100;
```

Selezione a cascata

Suggerimenti per il testing

- Prezzo = 5000 (maggiore di 3000) e contanti
- Prezzo = 5000 (maggiore di 3000) e non contanti
- Prezzo = 2000 (minore di 3000) e contanti
- Prezzo = 2000 (minore di 3000) e non contanti

Selezione annidata

La selezione annidata si presenta quando vi sono due o più selezioni le cui condizioni sono tra di loro dipendenti.

Viene analizzata la prima condizione e poi la seconda è analizzata solo se la prima è vera o falsa (o l'una o l'altra).

ATTENZIONE : La selezione annidata può presentarsi nel ramo vero, nel ramo falso o in entrambe i rami.

Selezione annidata

Determinare e visualizzare il maggiore di 3 numeri.

```
if (a>b)
    {if(a>c)
        cout << “Il numero maggiore è”<<a;
      else
        cout << “Il numero maggiore è”<<c;}
else
    {if(b>c)
        cout <<“Il numero maggiore è”<<b;
      else
        cout <<“Il numero maggiore è”<<c;}
```

Selezione multipla

Si usa quando si vogliono prevedere più di due opzioni, in corrispondenza di diversi valori di una variabile o di una espressione.

Si può implementare mediante una serie di selezioni a cascata oppure con strutture specifiche fornite dal linguaggio.

Selezione multipla

```
switch (variabile o espressione){  
    case valore1: istruzione1;  
                break;  
    case valore2: istruzione 2;  
                break;  
    ...  
    default: istruzione n;  
            break;  
}
```

Selezione multipla

Dato il numero dei lati visualizzare il nome del corrispondente poligono regolare.

```
switch (nlati){  
    case 3: cout<< "Triangolo";  
        break;  
    case 4: cout <<"Quadrato";  
        break;  
    ...  
    default: cout<<"Poligono sconosciuto";  
        break; }  
}
```

Selezione multipla

Costruire un calcolatrice per le 4 operazioni aritmetiche.

Acquisire due numeri e il simbolo dell'operazione.

La variabile operazione può assumere i seguenti valori

‘+’ ‘-’ ‘*’ ‘/’

In caso di immissione di un simbolo diverso visualizzare
“Operazione non prevista”.

Selezione multipla

```
switch (o){  
    case '+': cout <<a+b;  
        break;  
    case '-': cout <<a-b;  
        break;  
    case '*': cout <<a*b;  
        break;  
    case '/': cout << a/b;  
        break;  
    default: cout <<"Operazione non prevista";  
        break; }  
}
```