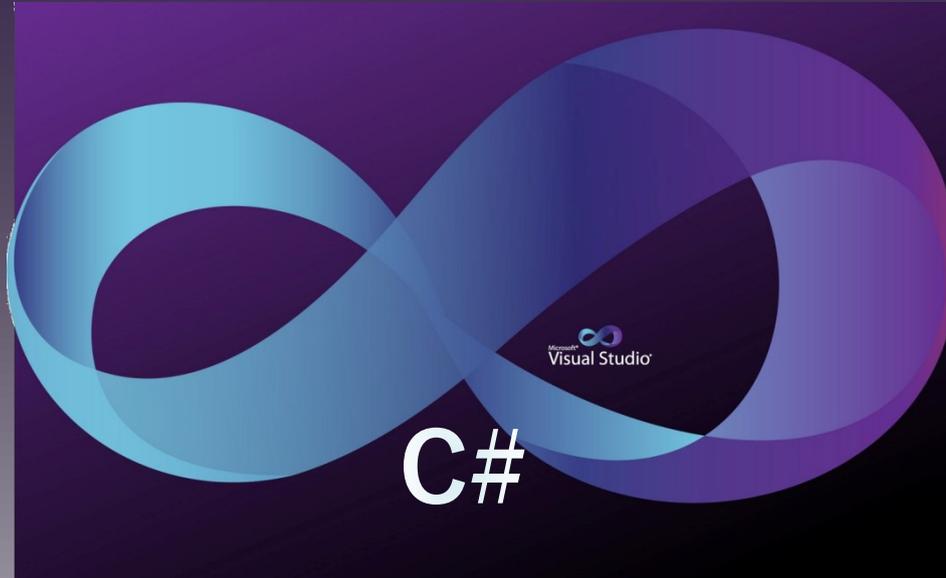


**ISTITUTO TECNICO INDUSTRIALE
G. M. ANGIOY
SASSARI**



CORSO DI PROGRAMMAZIONE

INTRODUZIONE ALLE ISTRUZIONI DI SELEZIONE

DISPENSA 02.01

[02-01_Istruzioni_Decisionali_\[ver_18\]](#)



Questa dispensa è rilasciata sotto la licenza Creative Common CC BY-NC-SA. Chiunque può copiare, distribuire, modificare, creare opere derivate dall'originale, ma non a scopi commerciali, a condizione che venga riconosciuta la paternità dell'opera all'autore e che alla nuova opera vengano attribuite le stesse licenze dell'originale.

Versione del: **07/11/2020**

Revisione numero: **18**

Prof. Andrea Zoccheddu
Dipartimento di Informatica

**DIPARTIMENTO
INFORMATICA E TELECOMUNICAZIONI**





ISTRUZIONI DECISIONALI

ISTRUZIONE DI SELEZIONE CONDIZIONALE

SE ... ALLORA ...

PROGETTO GUIDATO UNO

- prepara un **form1** simile alla figura
- associa al pulsante **button1** il seguente gestore di evento:

```
if (textBox1.Text == textBox2.Text)
    BackColor = Color.Red;
```

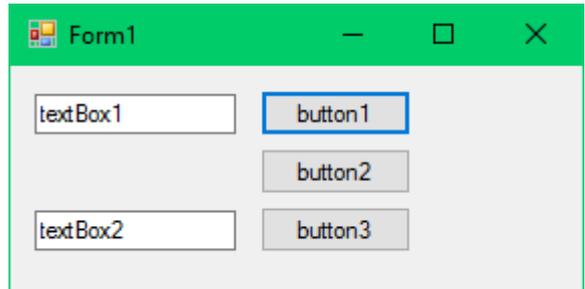
- associa al pulsante **button2** il seguente gestore di evento:

```
if (textBox1.Text != textBox2.Text)
    BackColor = Color.Green;
```

- associa al pulsante **button3** il seguente gestore di evento:

```
if (textBox1.Text == "0")
    BackColor = Color.Blue;
```

- prova a eseguire il progetto



PROGETTO GUIDATO DUE

- prepara un form1 simile alla figura
- associa al pulsante **button1** il seguente gestore di evento:

```
if (textBox1.Text == textBox2.Text)
    BackColor = Color.Red;
else
    BackColor = Color.Blue;
```

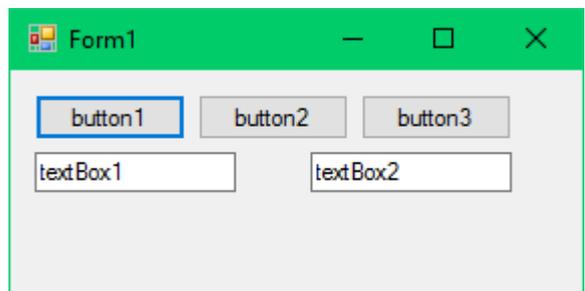
- associa al pulsante **button2** il seguente gestore di evento:

```
if (textBox1.Text != textBox2.Text)
    BackColor = Color.Green;
else
    BackColor = Color.Yellow;
```

- associa al pulsante **button3** il seguente gestore di evento:

```
if (textBox1.Text != "")
    BackColor = Color.White;
else
    BackColor = Color.Black;
```

- prova a eseguire il progetto





ISTRUZIONI DI SELEZIONE

ISTRUZIONE IF

Le istruzioni di selezione consentono di scegliere quali altre istruzioni eseguire tra alcune diverse alternative sezioni di codice, in base a una o più condizioni specifiche.

Il linguaggio **Visual C#** offre due istruzioni diverse:

- **if**
- **switch**

L'istruzione if offre due possibili alternative:

- **if**
- **if ... else**

L'esercizio precedente usa l'istruzione IF. L'istruzione assume due forme distinte, dette anche if_then (che corrisponde alla coppia se_allora dell'italiano) e if_then_else (che corrisponde alla trpla se_allora_altrimenti dell'italiano).

SINTASSI IF

L'istruzione if può assumere la prima sintassi seguente:

```
if (Condizione)  
    Istruzione;
```

con una sola istruzione. Dopo la parola riservata if (minuscola) è necessario scrivere una condizione tra parentesi tonde. La condizione è una espressione booleana, che deve restituire un valore booleano. L'istruzione che si scrive sotto la condizione è una sola e può essere una qualsiasi delle istruzioni Visual C#.

Esempio A

```
int x = 13;  
if (x < 17)  
    x = 23;  
int y = 29;
```

L'esempio precedente valuta la condizione che risulta True; quindi il programma assegna a x il valore 23 e poi prosegue modificando anche la y.

Esempio B

```
int x = 13;  
int y = 17;  
if (x < y)  
    x = y;  
y = 29;
```

L'esempio precedente valuta la condizione che risulta True; quindi il programma assegna a x il valore 17 e poi prosegue modificando anche la y.

funzionamento di IF

L'istruzione lavora nel seguente modo:

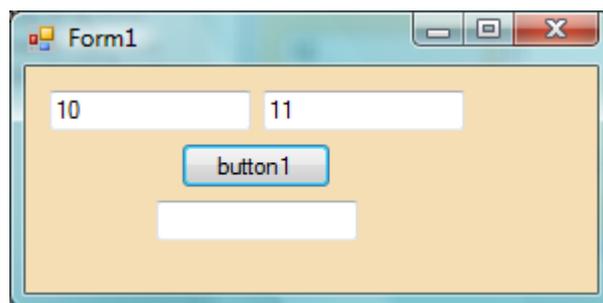
- valuta la condizione tra parentesi tonde che deve rendere true oppure false
- se la condizione ha reso **true**, allora si esegue l'istruzione scritta sotto la condizione
- se la condizione ha reso **false**, allora NON si esegue l'istruzione scritta sotto la condizione
- in ogni caso dopo aver eseguito o meno detta istruzione si prosegue con le istruzioni successive.



PROGETTO GUIDATO

- prepara un form1 simile alla figura
- associa al pulsante **button1** il seguente gestore di evento:

```
int x = Convert.ToInt32(textBox1.Text);  
int y = Convert.ToInt32(textBox2.Text);  
BackColor = Color.Green;  
if ( x > y )  
    BackColor = Color.Red;
```



- prova a ipotizzare i risultati del programma inserendo delle coppie di numeri nelle caselle opportune e prova a eseguire il progetto per verificare la correttezza del tuo ragionamento

SINTASSI IF... ELSE

L'istruzione **if_else** può assumere la prima sintassi seguente:

```
if (Condizione)  
    IstruzioneThen;  
else  
    IstruzioneElse;
```

con una sola istruzione in ciascuno dei due rami. Dopo la parola riservata **if** (minuscola) è necessario scrivere una condizione tra parentesi tonde. La condizione è una espressione booleana, che deve restituire un valore booleano. L'istruzione che si scrive sotto la condizione è una sola e può essere una qualsiasi delle istruzioni **Visual C#**. Dopo l'istruzione si scrive la parola chiave **else**, cui fa seguito una seconda istruzione (una sola) e può essere una qualsiasi delle istruzioni **Visual C#**.

Esempio C

```
int x = 13;  
int y = 17;  
if (x < y)  
    x = y;  
else  
    y = x;  
int z = 29;
```

FUNZIONAMENTO DI IF ... ELSE

L'istruzione lavora nel seguente modo:

1. valuta la condizione tra parentesi tonde che deve rendere true oppure false
 - 1.1 se la condizione ha reso **true**, allora si esegue l'istruzione **che segue if**
 - 1.2 se la condizione ha reso **false**, allora si esegue l'istruzione **che segue else**
2. in ogni caso dopo aver eseguito una delle due istruzioni, si prosegue con le istruzioni successive.



PROGETTO GUIDATO

- prepara un form1 simile alla figura
- associa al pulsante **button1** il seguente gestore di evento:

```
int x = Convert.ToInt32(textBox1.Text);
int y = Convert.ToInt32(textBox2.Text);
int z ;
if ( x > y )
    z = x;
else
    z = y;
textBox3.Text = Convert.ToString (z);
```



- prova a ipotizzare i risultati del programma inserendo delle coppie di numeri nelle caselle opportune e prova a eseguire il progetto per verificare la correttezza del tuo ragionamento

OSSERVAZIONE: SOLO UN'ISTRUZIONE DOPO L'IF E IDEM DOPO ELSE

È importante osservare che le istruzioni **if** ammettono una sola istruzione dopo la parola chiave.

Esempio C

Consideriamo il seguente frammento di codice:

```
Random dado = new Random();
int x = dado.Next(0,10);
int y = dado.Next(0,10);
if (x < y)
    x = y;
    y = 29;
```

Nell'esempio precedente qualsiasi valore restituisca la condizione, il programma assegna 29 alla variabile y IN OGNI CASO! Persino se la condizione fosse False. Questo accade perché solo un'istruzione è interna all'if.

Esempio D

Consideriamo anche il seguente frammento di codice:

```
Random dado = new Random();
int x = dado.Next(0,10);
int y = dado.Next(0,10);
if (x < y)
    x = 1;
    y = 0;
else
    x = 0;
    y = 1;
```

Nell'esempio precedente si solleva un errore, generato dalla frapposizione di una assegnazione prima della parola chiave else. Il programma NON FUNZIONA!

Se si desidera porre più istruzioni dentro l'if o dentro l'else è necessario usare un blocco costituito da parentesi graffe.



Esempio E

Il seguente esempio:

```
Random dado = new Random();
int x = dado.Next(0,10);
int y = dado.Next(0,10);
if (x < y)
{
    x = y;
    y = 29;
}
```

è radicalmente diverso dall'esempio C.

Nell'esempio E quando la condizione dell'if è vera allora esegue entrambe le assegnazioni, mentre se la condizione è falsa non ne esegue alcuna.

Esempio F

Il seguente esempio:

```
Random dado = new Random();
int x = dado.Next(0,10);
int y = dado.Next(0,10);
if (x < y)
{
    x = 1;
    y = 0;
}
else
{
    x = 0;
    y = 1;
}
```

è radicalmente diverso dall'esempio D.

Nell'esempio F quando la condizione dell'if è vera allora esegue entrambe le assegnazioni dove y diventa 0, mentre se la condizione è falsa esegue le altre due istruzioni dove y diventa 1.

CONDIZIONI E CONNETTIVI LOGICI

SE (QUESTO E QUELLO MA NON L'ALTRO)

Per verificare più condizioni insieme è necessario usare i connettivi (operatori) logici.

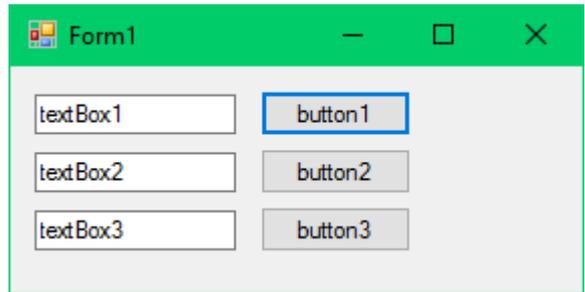
Gli operatori erano NOT AND OR e si scrivevano rispettivamente: **!** **&&** **||** .

Per assicurare che le operazioni siano corrette è consigliato utilizzare le parentesi per imporre le priorità operazionali. Inoltre le condizioni vanno sempre inserite dentro le parentesi tonde della sintassi dell'istruzione IF.

**PROGETTO GUIDATO CONNETTIVI LOGICI**

- prepara un form1 simile alla figura
- associa al pulsante **button1** il seguente gestore di evento:

```
int x = Convert.ToInt32(textBox1.Text);
int y = Convert.ToInt32(textBox2.Text);
int z = Convert.ToInt32(textBox3.Text);
if (( x > y ) && ( x > z ))
    BackColor = Color.Red;
if (( y > x ) && ( y > z ))
    BackColor = Color.Green;
if (( z > x ) && ( z > y ))
    BackColor = Color.Blue;
```



- associa al pulsante **button2** il seguente gestore di evento:

```
int x = Convert.ToInt32(textBox1.Text);
int y = Convert.ToInt32(textBox2.Text);
int z = Convert.ToInt32(textBox3.Text);
if (( x > y ) || ( x > z ))
    BackColor = Color.Yellow;
else
    BackColor = Color.Black;
```

- associa al pulsante **button3** il seguente gestore di evento:

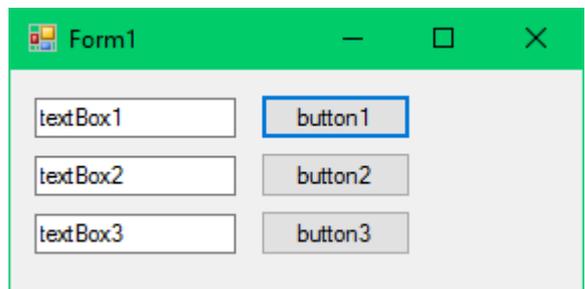
```
int x = Convert.ToInt32(textBox1.Text);
int y = Convert.ToInt32(textBox2.Text);
int z = Convert.ToInt32(textBox3.Text);
if (( x < 0 ) && ( y < 0 ) && ( z < 0 ))
    BackColor = Color.Red;
else
    if (( x < 0 ) || ( y < 0 ) || ( z < 0 ))
        BackColor = Color.Orange;
    else
        BackColor = Color.Green;
```

- prova a eseguire il progetto

PROGETTO GUIDATO CONNETTIVI LOGICI

- prepara un form1 simile alla figura
- associa al pulsante **button1** il seguente gestore di evento:

```
int x = Convert.ToInt32(textBox1.Text);
int y = Convert.ToInt32(textBox2.Text);
int z = Convert.ToInt32(textBox3.Text);
if ((x != y) && (x != z) && (y != z))
    BackColor = Color.Crimson;
else
    if (! (x != y) && (x != z) && (y != z))
        BackColor = Color.Gold;
    else
        BackColor = Color.Azure;
```



- prova a eseguire il progetto



RIEPILOGO: DA RICORDARE PER LA VERIFICA ORALE

DEFINIZIONI 1. DA RICORDARE SULL'ISTRUZIONE IF

- Le **istruzioni decisionali** (dette anche di selezione o condizionali) consentono di scegliere se eseguire altre istruzioni
- L'istruzione if è un'istruzione decisionale
- L'istruzione if offre due possibili alternative. L'istruzione if senza alternative oppure l'istruzione if ... else.
- L'istruzione if richiede una **condizione** sempre tra parentesi tonde.
- La condizione è una **espressione booleana**, ovvero una sequenza di simboli che si può valutare e che deve restituire un valore logico di tipo Vero o Falso

DEFINIZIONI 2. DA RICORDARE SULL'ISTRUZIONE IF A UNA VIA

- L'istruzione if base richiede una **condizione** tra parentesi tonde e **una sola altra istruzione** da eseguire
- Se la condizione rende True, allora il programma esegue l'istruzione interna.
- In ogni caso il programma prosegue ad eseguire le istruzioni successive

DEFINIZIONI 3. DA RICORDARE SULL'ISTRUZIONE IF ELSE A DUE VIE

- L'istruzione if else richiede una **condizione** tra parentesi tonde, poi **una sola altra istruzione** da eseguire se la condizione è vera, poi la parola chiave else e infine **una sola altra istruzione** da eseguire se la condizione è falsa.
- Se la condizione rende True, allora il programma esegue l'istruzione interna all'if.
- Se la condizione rende False, allora il programma esegue l'istruzione interna all'else.
- In ogni caso il programma prosegue ad eseguire le istruzioni successive

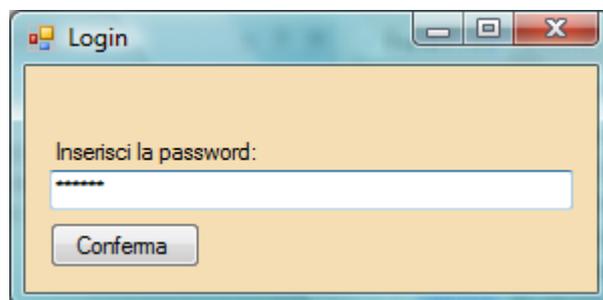


ESERCIZI

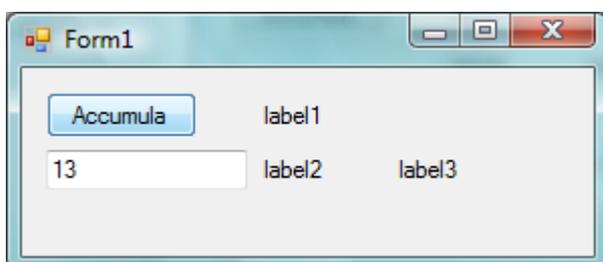
ESERCIZI

ESERCIZIO 1. PASSWORD

- prepara un form1 simile alla figura
- impostare la proprietà **PasswordChar** del controllo textBox1 con il carattere * (asterisco)
- prepara il pulsante Conferma per verificare la password inserita
- la password è scelta di nascosto dal programmatore
- se la parola inserita è corretta, allora colora la finestra di verde, altrimenti la colora di rosso



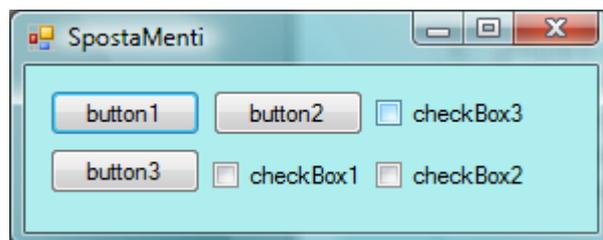
ESERCIZIO 2. NUMERI PARI



- prepara un form1 simile alla figura
- il pulsante Accumula deve accumulare **solo** i numeri interi **pari** digitati nella casella, mostrare nella prima etichetta **quanti** numeri pari sono stati inseriti, nella seconda etichetta la loro **somma** e nella terza la loro **media**
- attenzione a controllare se i numeri inseriti sono zero (nessun numero pari finora inserito)

ESERCIZIO 3. CONTROLLI CHECKBOXES

- prepara un form1 simile alla figura
- il **button1** deve colorare di Rosso la finestra solo se la checkBox1 è spuntata, altrimenti non ha effetto
- il **button2** deve colorare di Verde la finestra solo se la checkBox2 è spuntata, altrimenti non ha effetto
- per controllare se c'è il segno di spunta usare la proprietà **Checked** del controllo **CheckBox**; ad esempio `CheckBox.Checked == true`
- il **button3** deve colorare di Blu la finestra solo se la checkBox1 è spuntata, altrimenti la colora di Giallo
- eseguire il progetto e verificare il funzionamento dei pulsanti**
- modificare il progetto come segue:
- il **button1** deve colorare di Viola la finestra se e solo se entrambe checkBox1 e checkBox2 sono spuntate, altrimenti non ha effetto
- il **button2** deve colorare di Celeste la finestra se e solo se almeno una tra checkBox1 o checkBox2 è spuntata, altrimenti non ha effetto
- il **button3** deve colorare di Rosa la finestra se e solo se checkBox1 e checkBox2 hanno lo stesso segno di spunta, altrimenti la colora di Grigio



**ESERCIZIO 4. MINORE E MAGGIORE**

MinMax

Inserisci un numero:

Inserisci un numero:

Minimo:

Massimo:

- prepara un form1 simile alla figura
- il pulsante **Calcola** deve mostrare nella terza casella il numero minore scritto nelle altre due caselle e inoltre deve mostrare nella quarta casella il numero maggiore scritto nelle altre due caselle.

ESERCIZIO 5. PROGRAMMA COMMERCIALE 1

- Preparare un nuovo progetto per un programma di acquisto di frutta
- Il prezzo delle Arance è 0,55 euro / Kg. Ma se si acquista più di 100 Kg si ha uno sconto del 20% (solo sulle arance).
- Il prezzo delle Mele è 0,35 euro / Kg. Ma se si acquista più di 50 Kg si ha uno sconto del 15% (solo sulle mele).
- Il prezzo delle Pere è 0,65 euro / Kg. Ma se si acquista più di 120 Kg si ha uno sconto del 25% (solo sulle pere).

Acquisto:

Quantità:

Arance:

Mele:

Pere:

- Il pulsante acquisisce le quantità acquistate dalle rispettive textBox e poi calcola l'importo totale da pagare e lo mostra nella etichetta label5.

ESERCIZIO 6. PROGRAMMA COMMERCIALE 2

- Preparare un nuovo progetto per un programma di acquisto di frutta
- Le arance sono vendute a 0,55 euro / Kg.
Le mele sono vendute a 0,45 euro / Kg.
Le pere sono vendute a 0,85 euro / Kg.
- Se almeno un tipo di frutta è acquistato in quantità superiore a 150 Kg, allora si ottiene uno sconto del 50% sull'importo totale.
- Il pulsante acquisisce le quantità acquistate dalle rispettive textBox e poi calcola l'importo totale da pagare e lo mostra nella etichetta label5.

Acquisto:

Quantità:

Arance:

Mele:

Pere:

**ESERCIZIO 7. PESO CORPOREO**

- Preparare un nuovo progetto per un programma di controllo del peso corporeo. Il pulsante acquisisce altezza e peso di una persona e calcola il valore della seguente formula $PC = \text{Peso} / (\text{Altezza} * \text{Altezza})$;
- Il peso è in Kg, mentre l'altezza è in metri. L'indice così calcolato va confrontato con le seguenti fasce:

Situazione peso	Min	Max
Obesità di III classe (gravissima)	$\geq 40,00$	∞
Obesità di II classe (grave)	35,00	39,99
Obesità di I classe (moderata)	30,00	34,99
Visibilmente sovrappeso	25,00	29,99
Leggermente sovrappeso (falso magro)	22,00	24,99
Regolare (peso ideale)	19,00	21,99
Leggermente sottopeso	17,50	18,99
Visibilmente sottopeso (anoressia moderata)	15,50	17,49
Grave magrezza (anoressia grave)	0	$<15,50$

- Nella label3 andranno mostrati sia il valore dell'indice che la situazione di peso.

L'INDICE DI MASSA CORPOREA (ABBREVIATO IMC O BMI, DALL'INGLESE BODY MASS INDEX) È UN DATO BIOMETRICO, ESPRESSO COME RAPPORTO TRA PESO E ALTEZZA DI UN INDIVIDUO ED È UTILIZZATO COME UN INDICATORE DELLO STATO DI PESO FORMA.

ESERCIZIO 8. EQUAZIONE DI SECONDO GRADO

- Preparare un nuovo progetto per un programma di ricerca delle soluzioni di una equazione di secondo grado; solitamente l'equazione appare nella forma

$$ax^2 + bx + c = 0$$

coi coefficienti denotati dalle lettere a b e c

- L'equazione di secondo grado ha soluzioni che si ricavano dalla seguente formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- Tuttavia la porzione sotto radice quadrata (che si chiama discriminante) può trovarsi in uno dei seguenti modi:

- $b^2 - 4ac < 0 \rightarrow$ nessuna soluzione reale
- $b^2 - 4ac = 0 \rightarrow$ una sola soluzione reale
- $b^2 - 4ac > 0 \rightarrow$ due soluzioni reali distinte

- Preparare il progetto in modo che acquisisca i coefficienti a b e c;
- controlli il discriminante e mostri se la equazione ha soluzioni; se ha almeno una soluzione la mostra

- la radice quadrata si ottiene con la funzione `Math.Sqrt (valore)`

equazione completa $ax^2 + bx + c = 0$

$$\Delta > 0 \rightarrow x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$\Delta = 0 \rightarrow x_1 = x_2 = -\frac{b}{2a}$$

$\Delta < 0 \rightarrow$ *equazione impossibile*



SOMMARIO

ISTRUZIONE DI SELEZIONE CONDIZIONALE	2
SE . . . ALLORA	2
Progetto guidato uno	2
Progetto guidato due	2
ISTRUZIONI DI SELEZIONE	3
Istruzione IF	3
Sintassi IF	3
Progetto guidato	4
Sintassi IF ... ELSE	4
Funzionamento di IF ... ELSE	4
Progetto guidato	5
OSSERVAZIONE: SOLO UN'ISTRUZIONE DOPO L'IF E IDEM DOPO ELSE	5
CONDIZIONI E CONNETTIVI LOGICI	6
SE (QUESTO E QUELLO MA NON L'ALTRO)	6
Progetto guidato connettivi logici	7
Progetto guidato connettivi logici	7
RIEPILOGO: DA RICORDARE PER LA VERIFICA ORALE	8
Definizioni 1. Da ricordare sull'istruzione if	8
Definizioni 2. Da ricordare sull'istruzione if a una via	8
Definizioni 3. Da ricordare sull'istruzione if else a due vie	8
ESERCIZI	9
Esercizio 1. Password	9
Esercizio 2. Numeri pari	9
Esercizio 3. Controlli checkBoxes	9
Esercizio 4. Minore e maggiore	10
Esercizio 5. Programma commerciale 1	10
Esercizio 6. Programma commerciale 2	10
Esercizio 7. Peso corporeo	11
Esercizio 8. Equazione di secondo grado	11