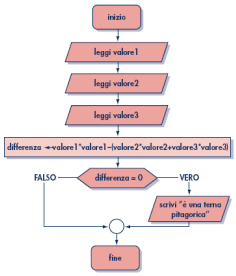
Strutture di controllo

Esempi in SmallBasic

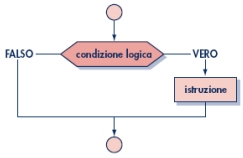
Problema

- **Problema**: *verificare se i tre valori passati in ingresso sono una terna pitagorica*.
- **Nota**: il primo valore immesso deve essere il maggiore dei tre.
- **Input**: tre valori numerici interi, il primo deve essere il maggiore dei tre.
- **Output**: in caso di verifica positiva, viene segnalato che si tratta di una terna pitagorica.

Algoritmo



Commento all'algoritmo

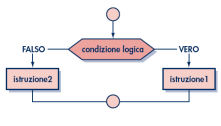


- **Il flusso di esecuzione non è più lineare**.
- Nel blocco decisionale un'istruzione è eseguita solo al verificarsi di una certa condizione
- Nella programmazione strutturata i costrutti di controllo devono avere un solo punto di ingresso e un solo punto di uscita: questo vincolo è rispettato dalla **struttura di controllo decisionale**

Il programma

```
1  TextWindow.WriteLine("Terna pitagorica")
2  TextWindow.Write("Inserire il primo valore (il maggiore)")
3  valore1 = TextWindow.ReadNumber()
4  TextWindow.Write("Inserire il secondo valore")
5  valore2 = TextWindow.ReadNumber()
6  TextWindow.Write("Inserire il terzo valore")
7  valore3 = TextWindow.ReadNumber()
8  differenza = valore1*valore1 - (valore2*valore2+valore3*valore3)
9  If differenza=0 Then
10   TextWindow.WriteLine("e' una terna pitagorica")
11 EndIf
```

… due alternative



- Nell'esempio precedente veniva eseguita una istruzione al verificarsi di una condizione
- In caso di condizione falsa non veniva eseguita alcuna istruzione
- Con questo costrutto "if-then-else" viene eseguita una istruzione o un'altra a seconda del valore della condizione.

Il programma

```
1  TextWindow.WriteLine("Terna pitagorica")
2  TextWindow.Write("Inserire il primo valore (il maggiore)")
3  valore1 = TextWindow.ReadNumber()
4  TextWindow.Write("Inserire il secondo valore")
5  valore2 = TextWindow.ReadNumber()
6  TextWindow.Write("Inserire il terzo valore")
7  valore3 = TextWindow.ReadNumber()
8  differenza = valore1*valore1 - (valore2*valore2+valore3*valore3)
9  If differenza=0 Then
10   TextWindow.WriteLine("e' una terna pitagorica")
11 Else
12   TextWindow.WriteLine("non e' una terna pitagorica")
13 EndIf
```

… una istruzione o … una sequenza di istruzioni …

- Tra Then ed EndIf, ci potrebbe essere più di un'operazione e il computer le eseguirebbe tutte nel caso in cui la condizione fosse valida. Per esempio, potresti scrivere qualcosa del genere:

```
If (ora < 12) Then
  TextWindow.Write("Buongiorno. ")
  TextWindow.WriteLine("Cosa preferisci a colazione?")
EndIf
```

Regole per i nomi delle Variabili

- Il nome deve iniziare con una lettera e non può coincidere con alcuna delle parole chiave come if, for, then, ecc.
- Un nome può contenere qualunque combinazione di lettere, cifre e caratteri di sottolineatura.
- È utile che il nome delle variabili sia significativo – dato che le variabili possono essere lunghe quanto vuoi, utilizza nomi di variabili che descrivono il loro scopo.

Principali operatori aritmetici

Operation	Example	Result
Addition	7 + 2	9
Addition	3.4 + 8.1	11.5
Subtraction	6 - 4	2
Subtraction	11.1 – 7.6	3.5
Multiplication	8 * 4	32
Multiplication	2.3 * 12.2	28.06
Division	12 / 2	6
Division	45.26 / 6.2	7.3

Operatori di confronto

> Greater than

< Less than

>= Greater than or equal to

<= Less than or equal to

= Equal to

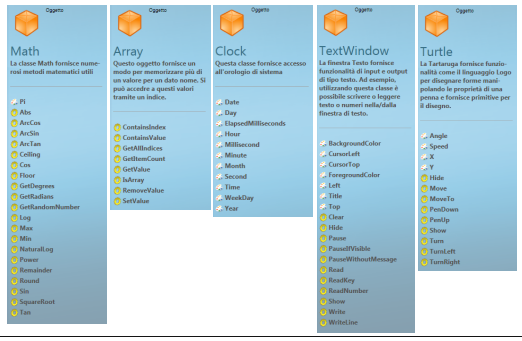
<> Not equal to

Operatori logici

And Logical And

Or Logical Or

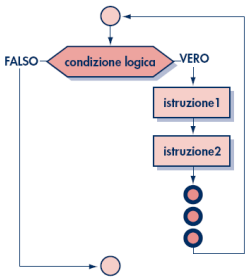
Alcuni oggetti



Le iterazioni (i cicli)

- L'iterazione è una struttura che consente di ripetere più volte l'esecuzione di un insieme di istruzioni.
- Due tipi di iterazione:
  - iterazioni guidate da eventi: non è noto a priori quante volte saranno eseguite.
    Tutto dipende dal fatto che un certo evento si verifichi o meno; se una condizione è vera (o è ancora vera) il blocco viene eseguito, altrimenti si passa all'istruzione successiva al termine del blocco.
  - iterazioni enumerative: consentono di eseguire un blocco di istruzioni un numero predefinito di volte.

Iterazioni non condizionali



- La condizione precede il ciclo.
- Solo se e la condizione è verificata il corpo del ciclo viene eseguito (per corpo del ciclo si intende l'insieme delle istruzioni interne)
- Al termine del corpo si torna a verificare la condizione.
- Il corpo del ciclo potrebbe non essere mai eseguito

Ciclo While

- Il ciclo While è utile specialmente quando il conteggio del ciclo non è noto.
- Il ciclo While è eseguito fin tanto che una data condizione è vera.
- Nell'esempio seguente, dimezziamo un numero fintanto che il risultato è maggiore di 1.

```
numero = 100
While (numero > 1)
  TextWindow.WriteLine(numero)
  numero = numero / 2
EndWhile
```

Ciclo For

- For … EndFor è un ciclo enumerativo.
- Una variabile (indice di ciclo) assume un valore iniziale e viene incrementata ad ogni esecuzione del ciclo fino a raggiungere il valore finale.
-
```
For i = 1 To 24
  TextWindow.WriteLine(i)
EndFor
```
- Se volessi che la variabile fosse incrementata di 2 invece che di 1 – per visualizzare tutti i numeri dispari tra 1 e 24 - sarebbe ancora possibile utilizzare un ciclo For.

```
For i = 1 To 24 Step 2
  TextWindow.WriteLine(i)
EndFor
```

La "tartaruga"

- Il Logo è un linguaggio di programmazione semplice ma potente che dispone di una "Tartaruga" che è visibile sullo schermo e risponde a comandi come Move (spòstati), Forward (avanza), Turn Right (gira a destra), Turn Left (gira a sinistra), ecc.
- Utilizzando la Tartaruga è possibile disegnare forme sullo schermo.
- Small Basic è dotato di un oggetto Tartaruga con molti comandi che possono essere chiamati all'interno dei programmi Small Basic.

Alcune funzioni della tartaruga

- Turtle.Show() – rende visibile la tartaruga
- Turtle.Move(100) – si sposta di 100 pixel
- Turtle.TurnRight() – gira a destra (90 gradi)
- Turtle.Turn(20) - gira di 20 gradi
- Turtle.PenUp() – alza il pennino
- Turtle.PenDown() –abbassa il pennino