Lezione 3

Istruzioni di controllo
Istruzioni Iterative

I blocchi di istruzione

- Si e' già detto come le istruzioni elementari siano costrutti identificati dal fatto di essere scritti su di una unica riga
- per istruzioni non elementari, come ad esempio quelle che ci serviranno per rappresentare le strutture di iterazione, si deve indicare l'insieme di istruzioni da iterare

I blocchi di istruzione

- per indicare blocchi di istruzioni come appartenenti alla stessa unita' concettuale, i vari costrutti adoperano la convenzione di racchiudere le istruzioni fra una parola chiave PAROLA per specificare il tipo di istruzione e END PAROLA

Istruzione di controllo

- Le istruzioni di controllo servono per influenzare il flusso di esecuzione di un programma
- tramite le istruzioni di controllo è possibile "compiere scelte" all'interno di un programma
- la scelta avviene valutando il valore di verità su di un predicato

If then

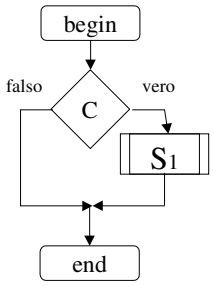
- Il blocco di controllo viene realizzato in VBA tramite il costrutto:

```
If predicato then
'istruzione A ...
'istruzione B ...
End If
```

- Il significato è che viene calcolato il valore di verità del predicato e nel caso in cui questo sia vero viene eseguita la serie di azioni compresa fra Then e End If

Corrispondenza

- Il costrutto corrisponde nel formalismo dei diagrammi a blocchi a:



Esempio

```
Dim eta Integer
eta=50
If eta > 18 Then
' e' maggiorenne
End If
```

Esempio

```
Dim eta Integer
eta=50
If eta > 18 And eta < 50 Then
' può avere la tessera
End If
```

Esempio

```
Dim eta Integer
eta=26
If (eta > 5 And eta < 18) Or eta >= 65 Then
' entra gratis al museo
End If
```

Esempio

```
Dim targa Integer
targa=25635
If (targa Mod 2) = 0 Then
'la targa e' pari
End If
```

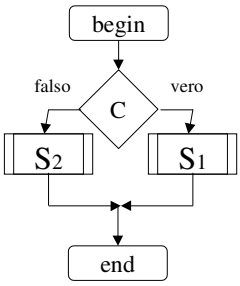
If then else

- Se si vuole specificare una azione anche nel caso in cui il predicato sia falso si usa il costrutto:

```
If predicato then
'azione ...
Else
'altra azione
End If
```

Corrispondenza

- Il costrutto corrisponde nel formalismo dei diagrammi a blocchi a:



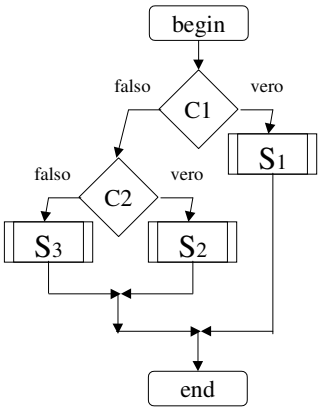
If Then ElseIf Else

- Può accadere di voler testare diverse alternative in successione
- per fare questo si utilizza il costrutto:

```
If predicato C1 Then
'azione S1...
ElseIf predicato C2 Then
'azione S2...
Else
'altra azione S3
End If
```

Corrispondenza

- Il costrutto corrisponde nel formalismo dei diagrammi a blocchi a:



Esercizio

- Scrivere la proposizione per determinare se un anno è bisestile
- Nota: un anno è bisestile se è divisibile per 4 ma non per 100 a meno che non sia divisibile per 400

Soluzione

```
If (anno Mod 4)=0 Then
  If (anno Mod 100)=0 Then
    If (anno Mod 400)=0 Then
    'e' bisestile
    Else
    'non e' bisestile
    End If
  Else
  'e' bisestile
  End If
Else
'non e' bisestile
End If
```

Nota

- Si possono inserire un numero arbitrario di condizioni

```
If predicato Then
'azione ...
ElseIf predicato2 Then
'azione 2...
ElseIf predicato3 Then
'azione 3...
ElseIf predicato4 Then
'azione 4...
Else
'altra azione
End If
```

Select case

- Nel caso in cui la condizione da testare per la validità di un predicato sia il valore assunto da una variabile si può usare un costrutto più leggibile:

```
Select Case variabile
Case val1
'azione 1…
Case val2
'azione2…
Case Else
'azione di default
End Select
```

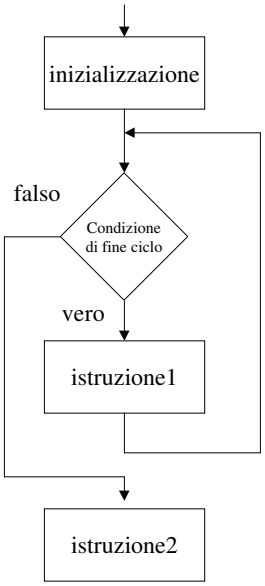
Esempio

```
Dim nome As string
nome="Antonio"
Select Case nome
Case "Mario"
'azione per Mario
Case "Ugo"
'azione per Ugo
Case "Antonio"
'azione per Antonio
Case Else
'azione per sconosciuto
End Case
```

Ciclo

- Il blocco di iterazione strutturata con condizione in testa è rappresentato con il costrutto:

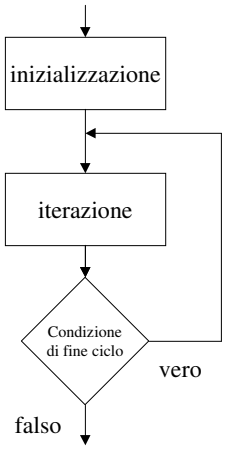
```
inizializzazione

Do While(condizione)
   istruzione1
Loop

istruzione2
```



Ciclo

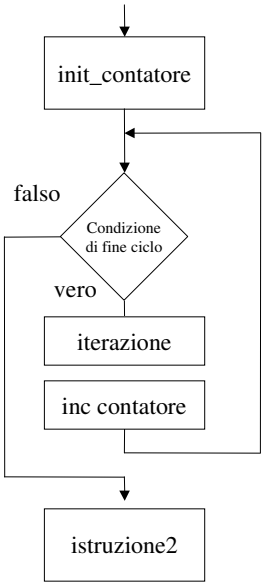
- Il blocco di iterazione strutturata con condizione in coda è rappresentato con l'istruzione:

```
inizializzazione;

Do
   istruzione1
Loop While(condizione)

istruzione2
```



Ciclo enumerativo

- Il ciclo in cui è noto il numero di volte in cui viene ripetuta l'iterazione viene tradotto con l'istruzione *For*



Ciclo enumerativo

- Quando è noto il numero di iterazioni da compiere si utilizza l'istruzione:

```
For counter = start To end
'iterazione
Next counter
'istruzione 2
```

Ciclo enumerativo

- L'utilizzo tipico è:

```
Dim i As Integer
For i = 1 to 10
'iterazione
Next i
```

- ma si può partire/arrivare a qualsiasi valore:

```
Dim i As Integer
For i = 20 to 100
'iterazione
Next i
```

Nota

- Un uso tipico del costrutto for e' per inizializzare i vettori (o array)

```
Dim i As Integer
Dim myArr(10) As Integer
For i = 0 to 10
  myArr(i)=0
Next i
```

Nota

- Si può anche omettere il contatore nell' istruzione Next
- cioè si può scrivere:

```
Dim i As Integer
For i = 1 to 10
'iterazione
Next
```

Ciclo enumerativo

- Nota: l'istruzione *for* è un modo compatto per rappresentare una ciclo di iterazione per vero con condizione in testa
- le seguenti espressioni sono equivalenti:

```
For counter = 1 To 10
  'iterazione
Next

i=1
Do While(i <= 10)
  'iterazione
  i=i+1
Loop
```

Step

- Nel caso in cui non si voglia incrementare la variabile contatore di una unita' alla volta si può specificare l'incremento tramite il costrutto Step come in:

```
Dim i As Integer
For i = 1 to 10 Step 2
'iterazione
Next
```

Step negativo

- Analogamente si può contare all'indietro incrementando con - 1

```
Dim i As Integer
For i = 100 to 1 Step -1
'iterazione
Next
```

Exit For

- Nel caso in cui si voglia terminare in modo anticipato il ciclo si può usare l'istruzione Exit For
- questa istruzione termina immediatamente l'esecuzione del ciclo For
- l'istruzione successiva che viene eseguita e' quella che sarebbe stata eseguita dopo aver terminato normalmente il ciclo

Esempio

```
Dim i As Integer
Dim k As Integer
k=34
For i=1 To 100
  If i > k Then
     Exit For
  End If
Next
```