

Introduzione a Visual Basic

Lezione 1

Concetti base e istruzioni condizionali

Mario Alviano

Introduzione all'informatica
Università della Calabria

<http://alviano.net/introinfo>

A.A. 2008/09

Introduzione

- Perché programmiamo?
 - Per risolvere problemi
- In che modo?
 - Con un algoritmo: una sequenza di istruzioni ben determinata
- Come iniziare?
 - Identificando il problema che si vuole risolvere
 - Quali sono i dati in **INPUT**?
 - Quale deve essere l'**OUTPUT**?

Introduzione

- Perché programmiamo?
 - Per risolvere problemi... non con il/la ragazzo/a!!!
- In che modo?
 - Con un algoritmo: una sequenza di istruzioni ben determinata
- Come iniziare?
 - Identificando il problema che si vuole risolvere
 - Quali sono i dati in INPUT?
 - Quale deve essere l'OUTPUT?

Introduzione

- Perché programmiamo?
 - Per risolvere problemi
- In che modo?
 - Con un algoritmo: una sequenza di istruzioni ben determinata
- Come iniziare?
 - Identificando il problema che si vuole risolvere
 - Quali sono i dati in **INPUT**?
 - Quale deve essere l'**OUTPUT**?

Introduzione

- Perché programmiamo?
 - Per risolvere problemi
- In che modo?
 - Con un algoritmo: una sequenza di istruzioni ben determinata
- Come iniziare?
 - Identificando il problema che si vuole risolvere
 - Quali sono i dati in **INPUT**?
 - Quale deve essere l'**OUTPUT**?

Un esempio: area del triangolo

Esempio (Area del triangolo)

- **Problema:** determinare l'area di un triangolo
 - **Input:** base e altezza del triangolo
 - **Output:** l'area del triangolo
-
- Come ottenere l'output a partire dall'input?
 - Ci sono diversi "modi"
 - Ogni "modo" rappresenta un algoritmo

Esempio (Area del triangolo)

- Per ottenere l'output applichiamo la ben nota formula

$$\text{area} = \frac{\text{base} \cdot \text{altezza}}{2}$$

Un esempio: area del triangolo

Esempio (Area del triangolo)

- **Problema:** determinare l'area di un triangolo
- **Input:** base e altezza del triangolo
- **Output:** l'area del triangolo

- Come ottenere l'output a partire dall'input?
- Ci sono diversi "modi"
- Ogni "modo" rappresenta un algoritmo

Esempio (Area del triangolo)

- Per ottenere l'output applichiamo la ben nota formula

$$\text{area} = \frac{\text{base} \cdot \text{altezza}}{2}$$

Un esempio: area del triangolo

Esempio (Area del triangolo)

- **Problema:** determinare l'area di un triangolo
- **Input:** base e altezza del triangolo
- **Output:** l'area del triangolo

- Come ottenere l'output a partire dall'input?
- Ci sono diversi "modi"
- Ogni "modo" rappresenta un algoritmo

Esempio (Area del triangolo)

- Per ottenere l'output applichiamo la ben nota formula

$$\text{area} = \frac{\text{base} \cdot \text{altezza}}{2}$$

Un esempio: area del triangolo

Esempio (Area del triangolo)

- **Problema:** determinare l'area di un triangolo
- **Input:** base e altezza del triangolo
- **Output:** l'area del triangolo

- Come ottenere l'output a partire dall'input?
- Ci sono diversi "modi"
- Ogni "modo" rappresenta un algoritmo

Esempio (Area del triangolo)

- Per ottenere l'output applichiamo la ben nota formula

$$\text{area} = \frac{\text{base} \cdot \text{altezza}}{2}$$

Un esempio: area del triangolo

Esempio (Area del triangolo)

- **Problema:** determinare l'area di un triangolo
 - **Input:** base e altezza del triangolo
 - **Output:** l'area del triangolo
-
- Come ottenere l'output a partire dall'input?
 - Ci sono diversi “modi”
 - Ogni “modo” rappresenta un algoritmo

Esempio (Area del triangolo)

- Per ottenere l'output applichiamo la ben nota formula

$$\text{area} = \frac{\text{base} \cdot \text{altezza}}{2}$$

Un esempio: area del triangolo

Esempio (Area del triangolo)

- **Problema:** determinare l'area di un triangolo
 - **Input:** base e altezza del triangolo
 - **Output:** l'area del triangolo
-
- Come ottenere l'output a partire dall'input?
 - Ci sono diversi “modi”
 - Ogni “modo” rappresenta un algoritmo

Esempio (Area del triangolo)

- Per ottenere l'output applichiamo la ben nota formula

$$\text{area} = \frac{\text{base} \cdot \text{altezza}}{2}$$

Codificare gli algoritmi

- Come possiamo codificare un algoritmo?
- In molti modi, ad esempio con un programma
 - Visual Basic
 - Pascal
 - C++
 - Java
 - ...
- Ma è più facile ragionare in termini di meta-programmi
 - Istruzioni non eseguibili dalla macchina ma più facilmente comprensibili dagli umani
- Allora dobbiamo imparare un altro linguaggio oltre a Visual Basic?

Codificare gli algoritmi

- Come possiamo codificare un algoritmo?
- In molti modi, ad esempio con un programma
 - Visual Basic
 - Pascal
 - C++
 - Java
 - ...
- Ma è più facile ragionare in termini di meta-programmi
 - Istruzioni non eseguibili dalla macchina ma più facilmente comprensibili dagli umani
- Allora dobbiamo imparare un altro linguaggio oltre a Visual Basic?

Codificare gli algoritmi

- Come possiamo codificare un algoritmo?
- In molti modi, ad esempio con un programma
 - **Visual Basic**
 - Pascal
 - C++
 - Java
 - ...
- Ma è più facile ragionare in termini di meta-programmi
 - Istruzioni non eseguibili dalla macchina ma più facilmente comprensibili dagli umani
- Allora dobbiamo imparare un altro linguaggio oltre a Visual Basic?

Codificare gli algoritmi

- Come possiamo codificare un algoritmo?
- In molti modi, ad esempio con un programma
 - **Visual Basic**
 - Pascal
 - C++
 - Java
 - ...
- Ma è più facile ragionare in termini di meta-programmi
 - Istruzioni non eseguibili dalla macchina ma più facilmente comprensibili dagli umani
- Allora dobbiamo imparare un altro linguaggio oltre a Visual Basic?

Codificare gli algoritmi

- Come possiamo codificare un algoritmo?
- In molti modi, ad esempio con un programma
 - **Visual Basic**
 - Pascal
 - C++
 - Java
 - ...
- Ma è più facile ragionare in termini di meta-programmi
 - Istruzioni non eseguibili dalla macchina ma più facilmente comprensibili dagli umani
- Allora dobbiamo imparare un altro linguaggio oltre a Visual Basic?

Codificare gli algoritmi

- Come possiamo codificare un algoritmo?
- In molti modi, ad esempio con un programma
 - **Visual Basic**
 - Pascal
 - C++
 - Java
 - ...
- Ma è più facile ragionare in termini di meta-programmi
 - Istruzioni non eseguibili dalla macchina ma più facilmente comprensibili dagli umani
- Allora dobbiamo imparare un altro linguaggio oltre a Visual Basic?
 - **Assolutamente no!**

Area del triangolo: meta-programma

Un meta-programma è una rappresentazione di un algoritmo in termini di istruzioni procedurali (anche espresse in linguaggio naturale).

Esempio (Area del triangolo: meta-programma)

```
Programma principale()
  Variabili: base, altezza, area

  base = Leggi("Inserire base")
  altezza = Leggi("Inserire altezza")

  area = base * altezza / 2
  Scrivi("L'area è " & area)
Fine Programma
```

Area del triangolo: meta-programma

Un meta-programma è una rappresentazione di un algoritmo in termini di istruzioni procedurali (anche espresse in linguaggio naturale).

Esempio (Area del triangolo: meta-programma)

```
Programma principale()  
  Variabili: base, altezza, area  
  
  base = Leggi("Inserire base")  
  altezza = Leggi("Inserire altezza")  
  
  area = base * altezza / 2  
  Scrivi("L'area è " & area)  
Fine Programma
```

Da meta-programma a programma Visual Basic

- La traduzione da meta-programma a programma Visual Basic è quasi immediata
- I programmi iniziano con `Sub main()` e finiscono con `End Sub`
- Per dichiarare una variabile si usa l'istruzione `Dim`, specificando il tipo
 - `As Integer`: la variabile è di tipo "numero intero"
 - `As Double`: la variabile è di tipo "numero decimale"
 - `As String`: la variabile è di tipo "testo"
 - `As Boolean`: la variabile è di tipo "booleano" (vero o falso)
 - `As ...`
- Per leggere un dato si usa l'istruzione `InputBox`
- Per scrivere qualcosa si usa l'istruzione `MsgBox`

Da meta-programma a programma Visual Basic

- La traduzione da meta-programma a programma Visual Basic è quasi immediata
- I programmi iniziano con **Sub main()** e finiscono con **End Sub**
- Per dichiarare una variabile si usa l'istruzione **Dim**, specificando il tipo
 - As Integer: la variabile è di tipo "numero intero"
 - As Double: la variabile è di tipo "numero decimale"
 - As String: la variabile è di tipo "testo"
 - As Boolean: la variabile è di tipo "booleano" (vero o falso)
 - As ...
- Per leggere un dato si usa l'istruzione **InputBox**
- Per scrivere qualcosa si usa l'istruzione **MsgBox**

Da meta-programma a programma Visual Basic

- La traduzione da meta-programma a programma Visual Basic è quasi immediata
- I programmi iniziano con **Sub main()** e finiscono con **End Sub**
- Per dichiarare una variabile si usa l'istruzione **Dim**, specificando il tipo
 - As Integer: la variabile è di tipo "numero intero"
 - As Double: la variabile è di tipo "numero decimale"
 - As String: la variabile è di tipo "testo"
 - As Boolean: la variabile è di tipo "booleano" (vero o falso)
 - As ...
- Per leggere un dato si usa l'istruzione **InputBox**
- Per scrivere qualcosa si usa l'istruzione **MsgBox**

Da meta-programma a programma Visual Basic

- La traduzione da meta-programma a programma Visual Basic è quasi immediata
- I programmi iniziano con **Sub main()** e finiscono con **End Sub**
- Per dichiarare una variabile si usa l'istruzione **Dim**, specificando il tipo
 - As Integer: la variabile è di tipo "numero intero"
 - As Double: la variabile è di tipo "numero decimale"
 - As String: la variabile è di tipo "testo"
 - As Boolean: la variabile è di tipo "booleano" (vero o falso)
 - As ...
- Per leggere un dato si usa l'istruzione **InputBox**
- Per scrivere qualcosa si usa l'istruzione **MsgBox**

Da meta-programma a programma Visual Basic

- La traduzione da meta-programma a programma Visual Basic è quasi immediata
- I programmi iniziano con **Sub main()** e finiscono con **End Sub**
- Per dichiarare una variabile si usa l'istruzione **Dim**, specificando il tipo
 - As Integer: la variabile è di tipo "numero intero"
 - As Double: la variabile è di tipo "numero decimale"
 - As String: la variabile è di tipo "testo"
 - As Boolean: la variabile è di tipo "booleano" (vero o falso)
 - As ...
- Per leggere un dato si usa l'istruzione **InputBox**
- Per scrivere qualcosa si usa l'istruzione **MsgBox**

Area del triangolo: programma Visual Basic

Esempio (Area del triangolo: programma Visual Basic)

```
Sub main()  
    Dim base As Integer  
    Dim altezza As Integer  
    Dim area As Integer  
  
    base = InputBox("Inserire base")  
    altezza = InputBox("Inserire altezza")  
  
    area = base * altezza / 2  
    MsgBox("L'area è " & area)  
End Sub
```

Scrittura di un programma Visual Basic

- Avviare Microsoft Visual Basic 6 (dal menù start)
- Selezionare dal menù “File” la voce “Nuovo Progetto”
- Se c'è un progetto aperto ci viene chiesto se vogliamo salvarlo (scegliete cosa fare)
- Selezionare “EXE standard” e premere il pulsante “OK”
- Dal menù “Progetto” selezionare “Rimuovi Form1”
- Dal menù “Progetto” selezionare “Inserisci modulo”, quindi cliccare su “Apri”
- Una volta scritto il codice, selezionare dal menù “Esegui” la voce “Avvia”

Istruzioni condizionali semplici

(1)

- Spesso si vuole eseguire un'istruzione o un blocco di istruzioni solo se si verifica una certa condizione.
- In questi casi si usa un'istruzione condizionale
- Un'istruzione condizionale determina se un blocco di istruzioni deve essere eseguito sulla base di una condizione booleana
- Come possiamo rappresentarla nel nostro meta-linguaggio?

```
Se CONDIZIONE Allora  
    BLOCCO ISTRUZIONI  
Fine Se
```

Istruzioni condizionali semplici

(1)

- Spesso si vuole eseguire un'istruzione o un blocco di istruzioni solo se si verifica una certa condizione.
- In questi casi si usa un'istruzione condizionale
- Un'istruzione condizionale determina se un blocco di istruzioni deve essere eseguito sulla base di una condizione booleana
- Come possiamo rappresentarla nel nostro meta-linguaggio?

```
Se CONDIZIONE Allora  
    BLOCCO ISTRUZIONI  
Fine Se
```

Istruzioni condizionali semplici

(1)

- Spesso si vuole eseguire un'istruzione o un blocco di istruzioni solo se si verifica una certa condizione.
- In questi casi si usa un'istruzione condizionale
- Un'istruzione condizionale determina se un blocco di istruzioni deve essere eseguito sulla base di una condizione booleana
- Come possiamo rappresentarla nel nostro meta-linguaggio?

```
Se CONDIZIONE Allora  
    BLOCCO ISTRUZIONI  
Fine Se
```

Istruzioni condizionali semplici

(1)

- Spesso si vuole eseguire un'istruzione o un blocco di istruzioni solo se si verifica una certa condizione.
- In questi casi si usa un'istruzione condizionale
- Un'istruzione condizionale determina se un blocco di istruzioni deve essere eseguito sulla base di una condizione booleana
- Come possiamo rappresentarla nel nostro meta-linguaggio?

```
Se CONDIZIONE Allora  
    BLOCCO ISTRUZIONI  
Fine Se
```

Istruzioni condizionali semplici

(2)

Esempio (Numero pari: meta-programma)

Dato un numero, determinare se è pari.

- **Input:** un numero intero
- **Output:** stampare "Pari" se il numero è pari

```
Programma principale()  
  Variabili: numero  
  numero = Leggi("Inserisci un numero")  
  
  Se numero è divisibile per 2 Allora  
    Scrivi("Pari")  
  Fine Se  
Fine Programma
```

Istruzioni condizionali semplici

(3)

- L'istruzione condizionale in VB è

```
If CONDIZIONE Then
    BLOCCO ISTRUZIONI
End If
```
- CONDIZIONE è un'espressione booleana
 - $a > 4$: a è maggiore di 4
 - $a \leq b$: a è minore o uguale a b
 - $a * b = 100$: a moltiplicato b è uguale a 100
 - $a \text{ Mod } 4 = 3$: a diviso 4 dà resto 3
 - $(4 < a) \text{ And } (a < 9)$: a è maggiore di 4 e a è minore di 9
 - $(4 < a) \text{ Or } (a < 9)$: a è maggiore di 4 oppure a è minore di 9

Istruzioni condizionali semplici

(4)

Esempio (Numero pari: programma VB)

Dato un numero, determinare se è pari.

- **Input:** un numero intero
- **Output:** stampare "Pari" se il numero è pari

```
Sub main()  
    Dim numero As Integer  
  
    numero = InputBox("Inserisci un numero")  
  
    If numero Mod 2 = 0 Then  
        MsgBox("Pari")  
    End If  
End Sub
```

Istruzioni condizionali con alternativa

(1)

- E se volessimo stampare “pari” se il numero è pari e “dispari” altrimenti?
- Possiamo specificare un blocco di istruzioni da eseguire se la condizione è falsa

```
Se CONDIZIONE Allora
    BLOCCO ISTRUZIONI 1
Altrimenti
    BLOCCO ISTRUZIONI 2
Fine Se
```

- Se CONDIZIONE è vera, allora verrà eseguito BLOCCO ISTRUZIONI 1
- Altrimenti, se CONDIZIONE è falsa, allora verrà eseguito BLOCCO ISTRUZIONI 2

Istruzioni condizionali con alternativa

(1)

- E se volessimo stampare “pari” se il numero è pari e “dispari” altrimenti?
- Possiamo specificare un blocco di istruzioni da eseguire se la condizione è falsa

```
Se CONDIZIONE Allora
    BLOCCO ISTRUZIONI 1
Altrimenti
    BLOCCO ISTRUZIONI 2
Fine Se
```

- Se CONDIZIONE è vera, allora verrà eseguito BLOCCO ISTRUZIONI 1
- Altrimenti, se CONDIZIONE è falsa, allora verrà eseguito BLOCCO ISTRUZIONI 2

Istruzioni condizionali con alternativa

(1)

- E se volessimo stampare “pari” se il numero è pari e “dispari” altrimenti?
- Possiamo specificare un blocco di istruzioni da eseguire se la condizione è falsa

```
Se CONDIZIONE Allora
    BLOCCO ISTRUZIONI 1
Altrimenti
    BLOCCO ISTRUZIONI 2
Fine Se
```

- Se CONDIZIONE è vera, allora verrà eseguito BLOCCO ISTRUZIONI 1
- Altrimenti, se CONDIZIONE è falsa, allora verrà eseguito BLOCCO ISTRUZIONI 2

Istruzioni condizionali con alternativa

(2)

Esempio (Pari o dispari: meta-programma)

Dato un numero, determinare se è pari o dispari.

- **Input:** un numero intero
- **Output:** stampare "Pari" se il numero è pari, altrimenti stampare "Dispari"

```
Programma principale()
```

```
  Variabili: numero
```

```
  numero = leggi("Inserisci un numero")
```

```
  Se numero è divisibile per 2 Allora
```

```
    Scrivi("Pari")
```

```
  Altrimenti
```

```
    Scrivi("Dispari")
```

```
  Fine Se
```

```
Fine Programma
```

Istruzioni condizionali con alternativa

(3)

- L'equivalente in VB è

```
If CONDIZIONE Then
    BLOCCO ISTRUZIONI 1
Else
    BLOCCO ISTRUZIONI 2
End If
```
- **NOTA:** else, in inglese, significa altrimenti

Istruzioni condizionali con alternativa

(3)

- L'equivalente in VB è

```
If CONDIZIONE Then
    BLOCCO ISTRUZIONI 1
Else
    BLOCCO ISTRUZIONI 2
End If
```
- **NOTA:** else, in inglese, significa altrimenti

Istruzioni condizionali con alternativa

(4)

Esempio (Pari o dispari: programma VB)

Dato un numero, determinare se è pari o dispari.

- **Input:** un numero intero
- **Output:** stampare "Pari" se il numero è pari, altrimenti stampare "Dispari"

```
Sub main()  
    Dim numero As Integer  
    numero = InputBox("Inserisci un numero")  
  
    If numero Mod 2 = 0 Then  
        MsgBox("Pari")  
    Else  
        MsgBox("Dispari")  
    End If  
End Sub
```

Istruzioni condizionali combinate

(1)

- Possiamo combinare “altrimenti” con un ulteriore “se”

Se CONDIZIONE 1 Allora

BLOCCO ISTRUZIONI 1

AltrimentiSe CONDIZIONE 2 **Allora**

BLOCCO ISTRUZIONI 2

Altrimenti

BLOCCO ISTRUZIONI 3

Fine Se

- BLOCCO ISTRUZIONI 1 verrà eseguito se CONDIZIONE 1 è vera
- BLOCCO ISTRUZIONI 2 verrà eseguito se CONDIZIONE 1 è falsa e CONDIZIONE 2 è vera
- BLOCCO ISTRUZIONI 3 verrà eseguito se CONDIZIONE 1 e CONDIZIONE 2 sono entrambe false
- Naturalmente, è possibile inserire ulteriori “AltrimentiSe”, omettere “Altrimenti” e annidare le istruzioni condizionali

Istruzioni condizionali combinate

(1)

- Possiamo combinare “altrimenti” con un ulteriore “se”
Se CONDIZIONE 1 Allora
 BLOCCO ISTRUZIONI 1
AltrimentiSe CONDIZIONE 2 Allora
 BLOCCO ISTRUZIONI 2
Altrimenti
 BLOCCO ISTRUZIONI 3
Fine Se
- BLOCCO ISTRUZIONI 1 verrà eseguito se CONDIZIONE 1 è vera
- BLOCCO ISTRUZIONI 2 verrà eseguito se CONDIZIONE 1 è falsa e CONDIZIONE 2 è vera
- BLOCCO ISTRUZIONI 3 verrà eseguito se CONDIZIONE 1 e CONDIZIONE 2 sono entrambe false
- Naturalmente, è possibile inserire ulteriori “AltrimentiSe”, omettere “Altrimenti” e annidare le istruzioni condizionali

Istruzioni condizionali combinate

(1)

- Possiamo combinare “altrimenti” con un ulteriore “se”
 - Se CONDIZIONE 1 Allora
BLOCCO ISTRUZIONI 1
 - AltrimentiSe** CONDIZIONE 2 **Allora**
BLOCCO ISTRUZIONI 2
 - Altrimenti
BLOCCO ISTRUZIONI 3
 - Fine Se
- BLOCCO ISTRUZIONI 1 verrà eseguito se CONDIZIONE 1 è vera
- BLOCCO ISTRUZIONI 2 verrà eseguito se CONDIZIONE 1 è falsa e CONDIZIONE 2 è vera
- BLOCCO ISTRUZIONI 3 verrà eseguito se CONDIZIONE 1 e CONDIZIONE 2 sono entrambe false
- Naturalmente, è possibile inserire ulteriori “AltrimentiSe”, omettere “Altrimenti” e annidare le istruzioni condizionali

Istruzioni condizionali combinate

(2)

Esempio (Esito esame: meta-programma)

Dato il voto di un esame, diciamo che se è minore di 18 si è bocciati, altrimenti se è maggiore di 30 si è promossi con lode, altrimenti si è promossi (perché $18 \leq \text{voto} \leq 30$).

- **Input:** voto (un numero intero)
- **Output:** stampare l'esito dell'esame

```
Programma principale()
  Variabili: voto
  voto = leggi("Inserisci il voto")

  Se voto < 18 Allora
    Scrivi("Bocciato :(")
  AltrimentiSe voto > 30 Allora
    Scrivi("Promosso con lode :D")
  Altrimenti
    Scrivi("Promosso :)")
  Fine Se
Fine Programma
```

Istruzioni condizionali combinate

(3)

- In Visual Basic: **Elseif** CONDIZIONE **Then**
- **Nota:** Elseif va scritto unito

Esempio (Esito esame: programma VB)

Dato il voto di un esame, diciamo che se è minore di 18 si è bocciati, altrimenti se è maggiore di 30 si è promossi con lode, altrimenti si è promossi (perché $18 \leq \text{voto} \leq 30$).

- **Input:** voto (un numero intero)
- **Output:** stampare l'esito dell'esame

```
Sub main()  
    Dim voto As Integer  
    voto = InputBox("Inserisci il voto")  
  
    If voto < 18 Then  
        MsgBox("Bocciato :(")  
    ElseIf voto > 30 Then  
        MsgBox("Promosso con lode :D")  
    Else  
        MsgBox("Promosso :)")  
    End If  
End Sub
```

Istruzioni condizionali combinate

(3)

- In Visual Basic: **Elseif** CONDIZIONE **Then**
- **Nota:** Elseif va scritto unito

Esempio (Esito esame: programma VB)

Dato il voto di un esame, diciamo che se è minore di 18 si è bocciati, altrimenti se è maggiore di 30 si è promossi con lode, altrimenti si è promossi (perché $18 \leq \text{voto} \leq 30$).

- **Input:** voto (un numero intero)
- **Output:** stampare l'esito dell'esame

```
Sub main()  
    Dim voto As Integer  
    voto = InputBox("Inserisci il voto")  
  
    If voto < 18 Then  
        MsgBox("Bocciato :(")  
    ElseIf voto > 30 Then  
        MsgBox("Promosso con lode :D")  
    Else  
        MsgBox("Promosso :)")  
    End If  
End Sub
```

Istruzioni condizionali combinate

(3)

- In Visual Basic: **Elseif** CONDIZIONE **Then**
- **Nota:** Elseif va scritto unito

Esempio (Esito esame: programma VB)

Dato il voto di un esame, diciamo che se è minore di 18 si è bocciati, altrimenti se è maggiore di 30 si è promossi con lode, altrimenti si è promossi (perché $18 \leq \text{voto} \leq 30$).

- **Input:** voto (un numero intero)
- **Output:** stampare l'esito dell'esame

```
Sub main()  
    Dim voto As Integer  
    voto = InputBox("Inserisci il voto")  
  
    If voto < 18 Then  
        MsgBox("Bocciato :(")  
    ElseIf voto > 30 Then  
        MsgBox("Promosso con lode :D")  
    Else  
        MsgBox("Promosso :)")  
    End If  
End Sub
```

L'ambiente VB e l'oggetto Form

- Cos'altro si può fare in Visual Basic?
- Vediamo qualcosa dell'ambiente
- L'oggetto Form
 - Una finestra della nostra applicazione
 - Possiamo ridimensionarla a nostro piacere

L'ambiente VB e l'oggetto Form

- Cos'altro si può fare in Visual Basic? **Molto!**
- Vediamo qualcosa dell'ambiente
- L'oggetto Form
 - Una finestra della nostra applicazione
 - Possiamo ridimensionarla a nostro piacere

L'ambiente VB e l'oggetto Form

- Cos'altro si può fare in Visual Basic? **Molto!**
- Vediamo qualcosa dell'ambiente
- L'oggetto Form
 - Una finestra della nostra applicazione
 - Possiamo ridimensionarla a nostro piacere

L'ambiente VB e l'oggetto Form

- Cos'altro si può fare in Visual Basic? **Molto!**
- Vediamo qualcosa dell'ambiente
- L'oggetto Form
 - Una finestra della nostra applicazione
 - Possiamo ridimensionarla a nostro piacere

L'ambiente VB e l'oggetto Form

- Cos'altro si può fare in Visual Basic? **Molto!**
- Vediamo qualcosa dell'ambiente
- L'oggetto Form
 - Una finestra della nostra applicazione
 - Possiamo ridimensionarla a nostro piacere

L'oggetto CommandButton

- Creare un nuovo progetto per ogni programma non è il massimo
- Impariamo a usare i bottoni
 - Inserimento e ridimensionamento
 - Proprietà Nome
 - Proprietà Caption
 - Evento Click

L'oggetto CommandButton

- Creare un nuovo progetto per ogni programma non è il massimo
- Impariamo a usare i bottoni
 - Inserimento e ridimensionamento
 - Proprietà Nome
 - Proprietà Caption
 - Evento Click

L'oggetto CommandButton

- Creare un nuovo progetto per ogni programma non è il massimo
- Impariamo a usare i bottoni
 - Inserimento e ridimensionamento
 - Proprietà Nome
 - Proprietà Caption
 - Evento Click

L'oggetto CommandButton

- Creare un nuovo progetto per ogni programma non è il massimo
- Impariamo a usare i bottoni
 - Inserimento e ridimensionamento
 - Proprietà Nome
 - Proprietà Caption
 - Evento Click

L'oggetto CommandButton

- Creare un nuovo progetto per ogni programma non è il massimo
- Impariamo a usare i bottoni
 - Inserimento e ridimensionamento
 - Proprietà Nome
 - Proprietà Caption
 - Evento Click

L'oggetto CommandButton

- Creare un nuovo progetto per ogni programma non è il massimo
- Impariamo a usare i bottoni
 - Inserimento e ridimensionamento
 - Proprietà Nome
 - Proprietà Caption
 - Evento Click

Esercizi

- Creiamo un bottone per ognuno dei seguenti programmi
 - Anno bisestile
 - Area quadrato
 - Perimetro rettangolo
 - Conversione Lire - Euro
 - Conversione Euro - Lire
 - Media di 5 interi
- L'implementazione? Spetta a voi!
 - Identificate le variabili di input
 - Identificate l'output da restituire
 - Identificate un algoritmo per ottenere l'output
- Il primo lo facciamo insieme

Esercizi

- Creiamo un bottone per ognuno dei seguenti programmi
 - Anno bisestile
 - Area quadrato
 - Perimetro rettangolo
 - Conversione Lire - Euro
 - Conversione Euro - Lire
 - Media di 5 interi
- L'implementazione? Spetta a voi!
 - Identificate le variabili di input
 - Identificate l'output da restituire
 - Identificate un algoritmo per ottenere l'output
- Il primo lo facciamo insieme

Esercizi

- Creiamo un bottone per ognuno dei seguenti programmi
 - Anno bisestile
 - Area quadrato
 - Perimetro rettangolo
 - Conversione Lire - Euro
 - Conversione Euro - Lire
 - Media di 5 interi
- L'implementazione? Spetta a voi!
 - Identificate le variabili di input
 - Identificate l'output da restituire
 - Identificate un algoritmo per ottenere l'output
- Il primo lo facciamo insieme

Svolgimento esercizio Anno Bisestile

(1)

Esempio

- In genere, un anno è bisestile se è divisibile per 4
- Tuttavia, per i secoli questo non vale
- Un secolo è bisestile se è divisibile per 400
(ovvero solo un secolo su 4 è bisestile)

- **Input:** anno (un intero)
- **Output:** stampa "L'anno è bisestile" o "L'anno non è bisestile"

Svolgimento esercizio Anno Bisestile

(1)

Esempio

- In genere, un anno è bisestile se è divisibile per 4
- Tuttavia, per i secoli questo non vale
- Un secolo è bisestile se è divisibile per 400
(ovvero solo un secolo su 4 è bisestile)

- **Input:** anno (un intero)
- **Output:** stampa “L’anno è bisestile” o “L’anno non è bisestile”

Svolgimento esercizio Anno Bisestile

(2)

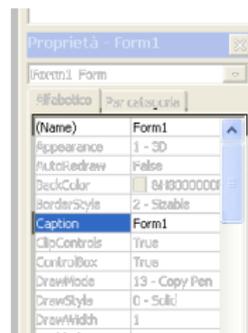
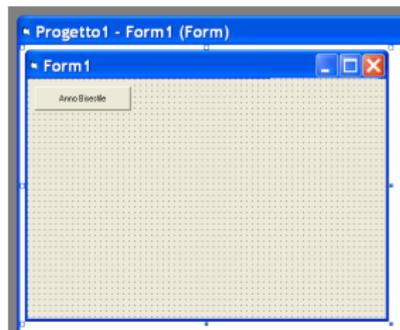
Esempio (Anno bisestile: meta-programma)

```
Programma principale()  
  Variabili: anno  
  
  anno = Leggi("Inserisci un anno")  
  
  Se (anno è divisibile per 4 E  
      anno non è divisibile per 100) Oppure  
      (anno è divisibile per 400) Allora  
      Scrivi("L'anno è bisestile")  
  Altrimenti  
      Scrivi("L'anno non è bisestile")  
  Fine Se  
Fine Programma
```

Svolgimento esercizio Anno Bisestile

(3)

- Creiamo un nuovo progetto, specificando “EXE standard”
- Allarghiamo la form
- Inseriamo un CommandButton
- Settiamo la proprietà (Name) del bottone a “CommandBisestile”
- Settiamo la proprietà Caption del bottone a “Anno Bisestile”
- Facciamo doppio-click sul bottone per scrivere il codice da eseguire quando il bottone verrà premuto
 - CommandBisestile_Click()



Svolgimento esercizio Anno Bisestile

(4)

Esempio (Anno bisestile: programma VB)

```
Private Sub CommandBisestile_Click()  
    Dim anno As Integer  
    anno = InputBox("Inserisci un anno")  
  
    If (anno Mod 4 = 0 And [...]  
        [...] anno Mod 100 <> 0) [...]  
        [...] Or (anno Mod 400 = 0) Then  
        MsgBox("L'anno è bisestile")  
    Else  
        MsgBox("L'anno non è bisestile")  
    End If  
End Sub
```

- **ATTENZIONE:** In Visual Basic la condizione DEVE essere messa su un'UNICA linea!

Svolgimento esercizio Anno Bisestile

(4)

Esempio (Anno bisestile: programma VB)

```
Private Sub CommandBisestile_Click()  
    Dim anno As Integer  
    anno = InputBox("Inserisci un anno")  
  
    If (anno Mod 4 = 0 And [...]  
        [...] anno Mod 100 <> 0) [...]  
        [...] Or (anno Mod 400 = 0) Then  
        MsgBox("L'anno è bisestile")  
    Else  
        MsgBox("L'anno non è bisestile")  
    End If  
End Sub
```

- **ATTENZIONE:** In Visual Basic la condizione DEVE essere messa su un'UNICA linea!