Escuela Universitaria de Ingeniería Vitoria-Gasteiz
Ingeniaritzako Unibertsitate Eskola Vitoria-Gasteiz

Fundamentals of Computer Science
2010-2011
Laboratory 2
Sequential programs

erran ta zabal zazu

Objetivos:
- VB Controls: **command button** (cmd), **form** (frm, label (lbl), **text box** (txt) and **picture box** (pct)
- VB controls properties: : **`Name (Nombre)`**, **`Caption`** , **`Text`**, **`Enabled`**
- Changing dynamically the properties of the control
- Boolean constants **`True`**, **`False`**
- **`Print`** y **`Cls`** instructions on picture boxes
- Design of flowcharts for sequential programs

Greeting program 2 (solved)

Let's create a program similar to the one in previous laboratory using different controls (graphic objects). Thus, we read the user's name from a text box and we write the greeting message on a picture box, as shown in Figure 2.1.

Interface

Figure 2.1 Greeting form.

Controls used

- Command Button (**`cmd`**): for example, "*Greet me*"
- Label (**`lbl`**): for example, "*Name*"
- Text Box (**`txt`**)
- Picture Box (**`pct`**)

Operation

1. By clicking on the button labelled "*Greet me*" we collect the value previously introduced by the user in the text box and display the greeting message on the picture box.
2. By clicking on the button labelled "*Erase*" we must remove the contents of the text box (by changing its Text property) and clear the picture box (by executing the Cls function on it).
3. By clicking on the button labelled "*Quit*" the program will finish using the **`End`** instruction, described in detail in the previous laboratory.

Steps to follow

1. Create the objects shown in Figure 2.1 with the properties described in Table 2.1.

Control type/object	Properties	Value
Form (`frm`)	**Name**	**`frmGreet`**
	Caption	Greeting program
Label (`lbl`)	**Name**	**`lblName`**
	Caption	Name
Text Box (`txt`)	**Name**	**`txtName`**
	Text	
Picture Box (`pct`)	**Name**	**`pctResult`**
	AutoRedraw	**`True`**
Command Button (`cmd`)	**Name**	**`cmdGreet`**
	Caption	Greet me
	Name	**`cmdErase`**
	Caption	Erase
	Name	**`cmdQuit`**
	Caption	Quit

Table 2.1 Objects and their properties.

2. Add the code to the buttons to fulfil their mission.

   1. **Code of the "*Greet me*" button**: Collect the value of the **`Text`** property of the text box called `txtName` (holding the name of the user) and form the greeting string to write it on the `pctResult` picture box using the **`Print`** function on it. If its **`AutoRedraw`** property is set to **`False`**, each time we cover the picture box the contents will disappear.

```
Sub cmdGreet_Click()
  Dim name As String, greeting As String ' Input name and greeting string formed
  name = txtName.Text   ' Read the value of the Text property of the text box
  greeting = "Hi " & name & "!" ' Calculate the greeting string
  pctResult.Print (greeting) ' Display the result
End Sub
```

   2. **Code for the "*Erase*" button**: remove the contents of the text box by assigning an empty string to the **`Text`** property of the text box and executing the **`Cls`** instruction on the `pctResult` picture box.

```
Sub CmdErase_Click()
  pctResult.Cls
  txtName.Text = ""
End Sub
```

   3. **Code for the "*Quit*" button**: Same as equivalent in the solved exercise.

Exercise 2.1: Adder program

In order to practice with these new input/output methods our first exercise is similar to the adding program in our previous laboratory. Instead of using **`InputBox`** you must read the information from the text box and instead of showing the result using **`MsgBox`** you must show it in another text box.

Interface

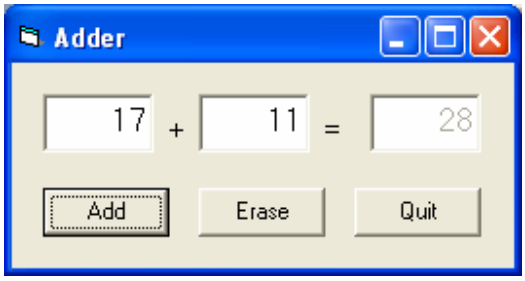

Figure 2.2 Adder form.

Objects utilised

- Three text boxes: operand 1 (`txtOp1`), operand 2 (`txtOp2`) and result (`txtRes`). The user cannot write on the latter and we prevent this by setting the **`Enabled`** property to **`False`** (you can do it straight from the VB integrated development environment, IDE).
- Three command buttons: Sum (**`cmdSum`**), Erase (**`cmdErase`**) and Quit (**`cmdQuit`**).
- Two labels ("+" and "="). The name given to these is not significant as we need not refer to them within the program code.

Steps to follow

1. Design the interface adding the controls to the form and modifying the described properties. It is convenient to give significant names to the objects because this strategy helps identifying them within the program.
2. Add the code to the command buttons:

    1. **Erase** and **Quit** are similar to their corresponding in the previous example.

    2. Command button **Sum** will collect and sum the values introduced in the`txtOp1` and `txtOp2` text boxes and will display the result in `txtRes` text box.

Exercise 2.2: Temperature conversion program

Design the flowchart, the **interface** and **implement** the VB program to **convert** from Celsius degrees to Fahrenheit y vice versa given the interface in Figure 2.3. Note that within the program we specify real numbers separated by a decimal point, but as we are using here a Spanish interface we must enter the numbers separated by commas.

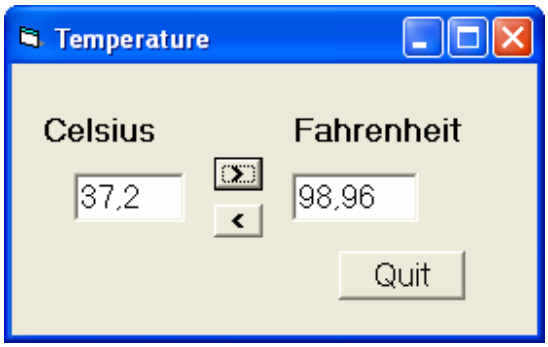

Figure 2.3 Form for the temperature converter.

The conversion formula is: **ºF = ºC · (9/5) + 32**

Exercise 2.3: Calculator program

We propose a simple integer calculator to add, subtract, multiply, divide and obtain the remainder.

Interface

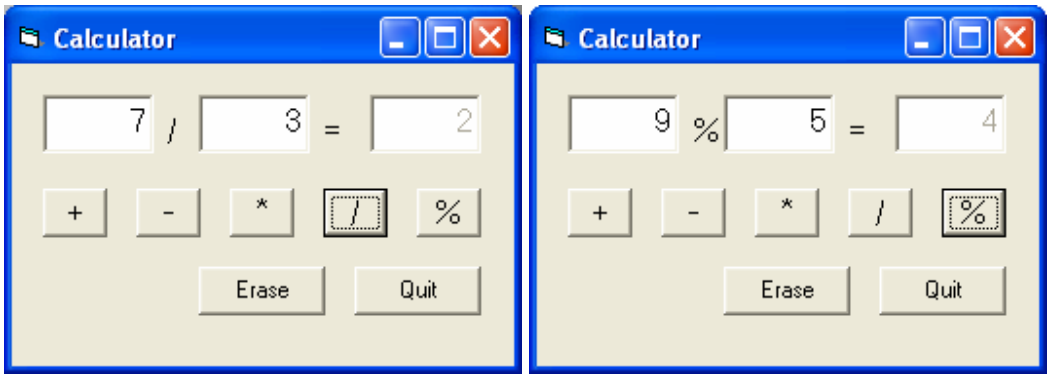

Figure 2.4 Calculator form after two different operations.

Steps to follow

The operation will be similar to that of the corresponding program seen in the previous exercise, but this time after clicking on the operator (+, –, *, /, %) apart from realising the operation between the operators the operation label between the operators will be changed to show the operation that has been carried out.

On the propose interface we have represented with a slash (‘/’) the button corresponding to the integer division but in VB (within the program) this operator is represented by the

backslash (‘\’). Additionally, we have represented with a percentage sign (‘%’) the reminder operation but in VB this operator is expressed by the **Mod** reserved keyword.

When we click on any of the operator button we need to:

1. Read the operand numbers on the text boxes.
2. Carry out the operation on a separate variable.
3. Write the result in the corresponding text box.
4. Change the Caption property of the operator label in between the two operands.

Exercise 2.4: Program to calculate euro notes

Design the flowchart, the **interface** and **implement** the VB program to obtain how to split a given total amount of euro notes of 500, 200, 100, 50, 20, 10, 5 y spare with an interface like the on in Figure 2.5.

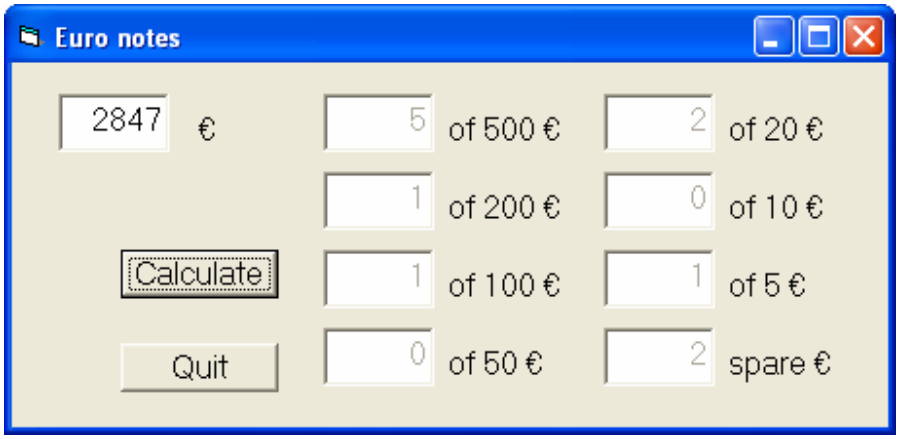

Figure 2.5 Form of the program to calculate the amount of euro notes.

Quick reference table

Operations with controls		
Operation	**Syntax**	**Example**
Read txt	***Variable* = *txt.Text***	`name = txtName.Text`
Write txt	***Txt.Text* = *expression***	`txtGreet.Text = "Hi " & name`
Clear txt	***Txt.Text* = ""**	`txtGreet.Text = ""`
Read pct	***Cannot be done***	
Write pct	***pct.Print expression***	`pctGreet.Print "Hi " & name`
Clear pct	***pct.Cls***	`pctGreet.Cls`