Lab3: Windows Forms App

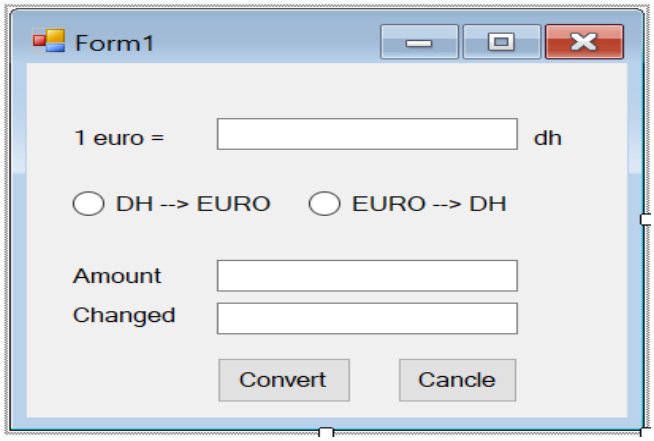
This lab will familiarize you with the different controls and creating the different WinForms applications. You can study some of controls in C# documentation or following link:

http://csharp.net-informations.com/gui/cs_forms.htm

Exercise 1 :

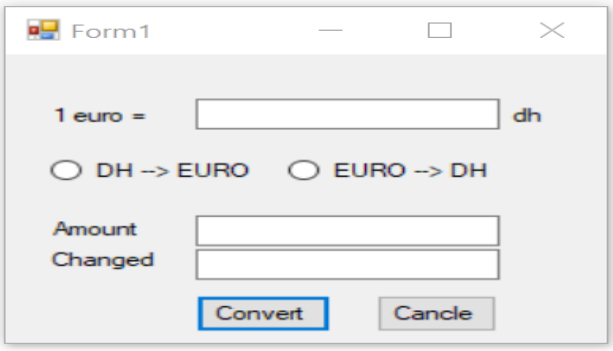
Objective: Manipulate standard controls: **RADIOBUTTON**, **BUTTON**

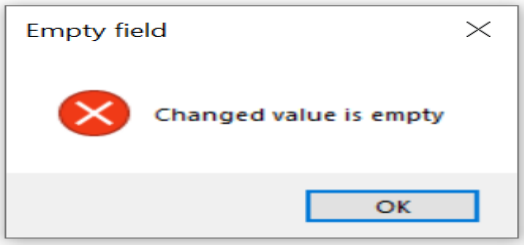
Create an application that allows to realize the conversion EURO / DH and DH / EURO as follows:



- Clicking on the "**Convert**" button converts the amount from DH or EURO to DH or EURO depending on the option chosen.
- Predict error cases (empty field, invalid field, operation not selected, ...)
- Clicking on the "**Cancel**" button clears all the fields.

  Execution example:

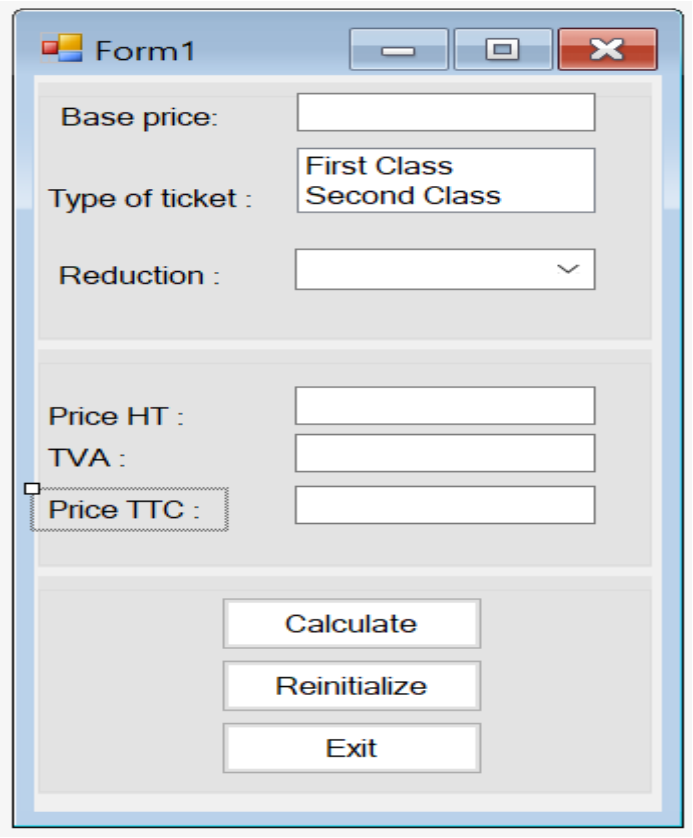




Exercise 2:

Objective: Manipulate the common controls **LISTBOX**, **COMBOBOX**

Create the following C# application:



- The **Calculate button** displays the price excluding tax, and the total price including all taxes. (TVA=20%)

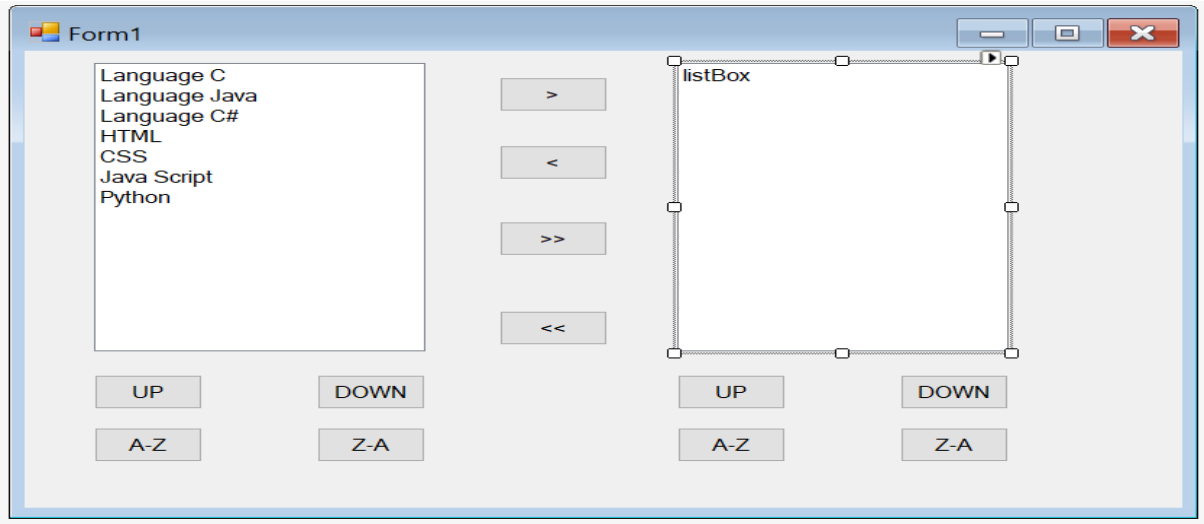
1. The price of the second class equals the basic price.
2. The price of first class equals the base price + 20% of it.
3. The youth card allows you to have a 40% discount on the ticket price.
4. The adult card allows you to have a 30% discount on the ticket price.
5. The family card allows you to have a 50% discount on the ticket price.

- The **Reinitialize button** will empty all the field
- The **Exit button** will exit the program

Exercise 3 :

Objective: Manipulate the common controls **LISTBOX**, **COMBOBOX**

Create C# application



The list on the left feeds when the form loads. The list on the right is initially empty and is powered by the two buttons**'>'**and**'>>'.**

Role of the different buttons:

> Moves the selected item from the list on the left to the list on the right.

< Moves the selected item from the list on the right to the list on the left.

>> Moves the entire list from left to right.

<< Moves the entire list from the right to the left list.

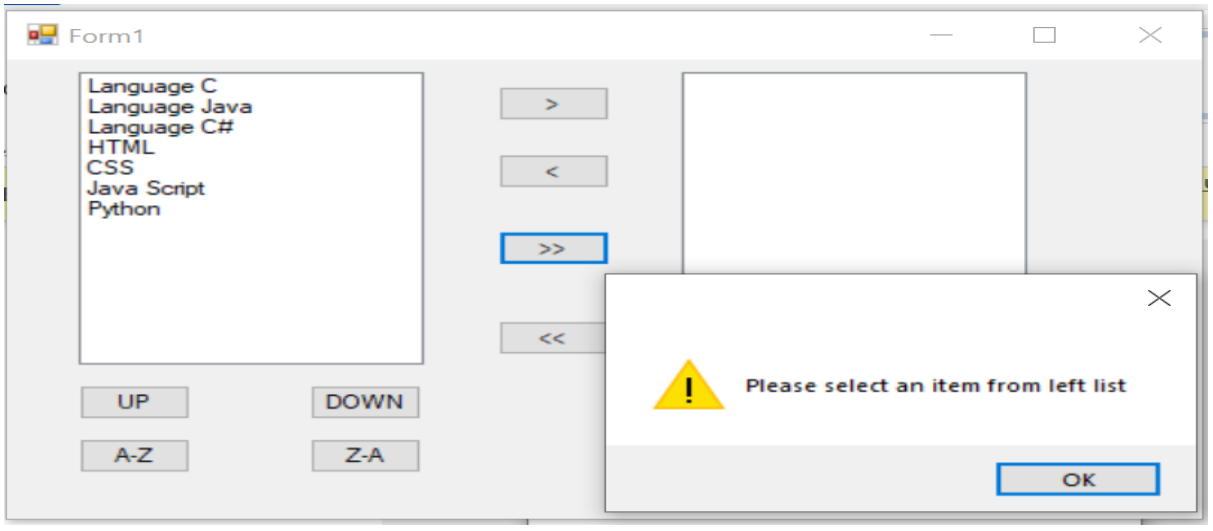
UP Moves the selected item one line up.

DOWN Moves the selected item down one row.

A-Z Sorts the corresponding list in alphabetical order.

Z-A Sorts the corresponding list in reverse order.

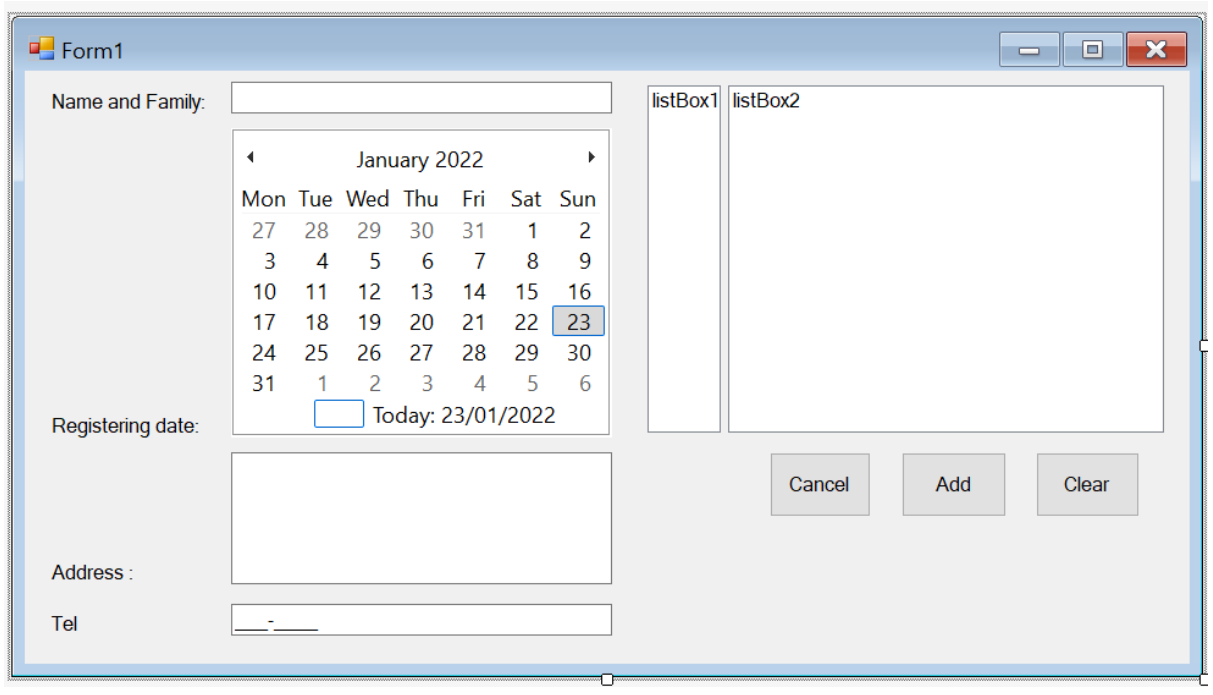
Execution example:



Exercise 4 :

Objective: Manipulate advanced controls **MONTHCALENDAR**, **MASKEDTEXTBOX**

Create the following C# application:



- Clicking on the "**Add**" button allows you to add a customer to the list by checking the following constraints:

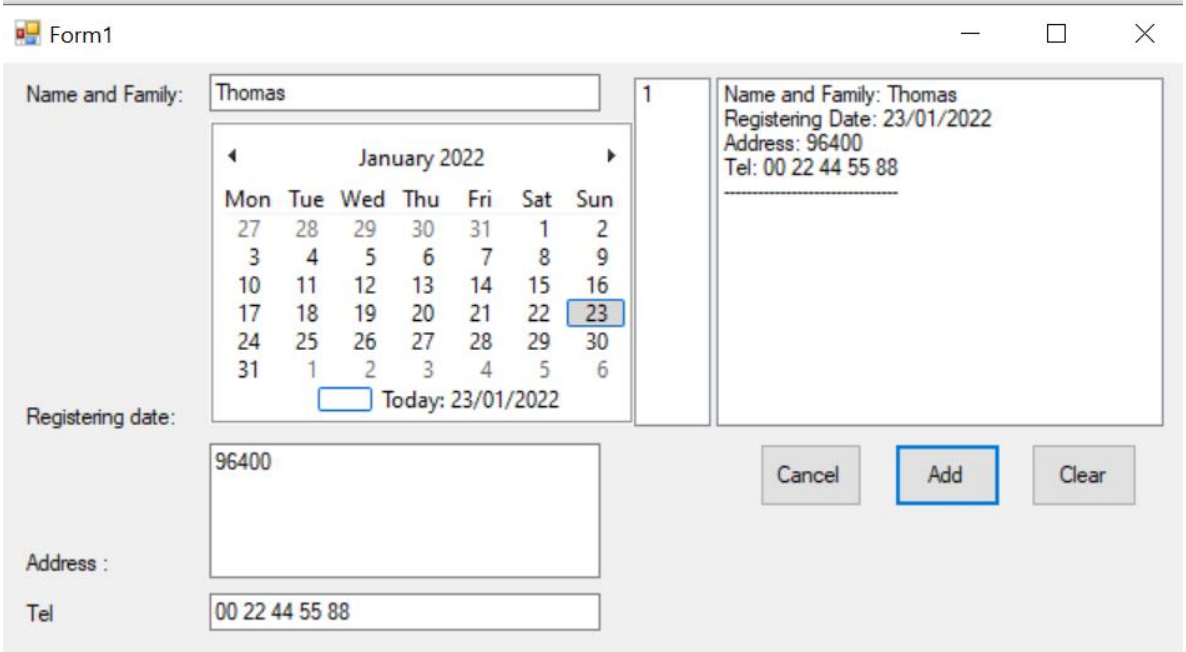
Fields should not be empty. / The registration date must be less than or equal to the current date /The phone number must be valid (10 digits)/ Request confirmation before adding the client.

- Clicking on the "**Cancel**" button clears the fields.
- Clicking on the "**Clear**" button allows you to empty the list of customers after confirmation.

Note:

➢ Use the MASK property of the **MASKEDTEXTBOX** control to adjust the phone number input mask.

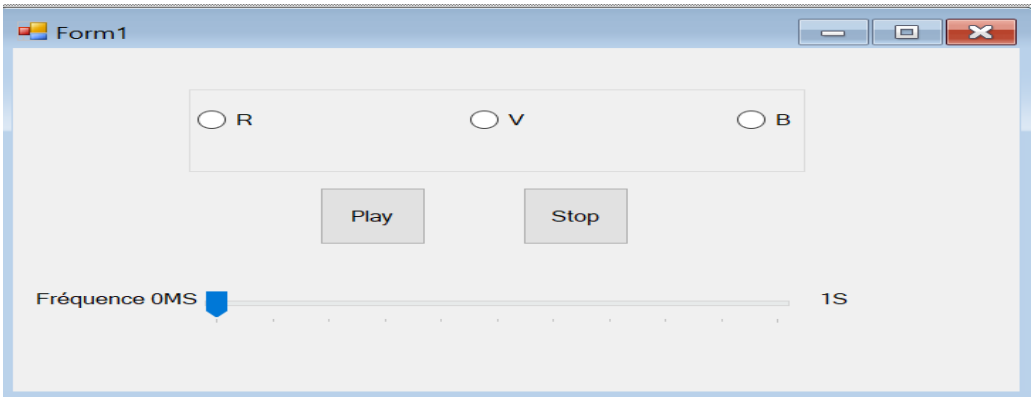
Execution example:



Exercise 5 :

Objective: Manipulate advanced **TIMER**, **TRACKBAR** controls

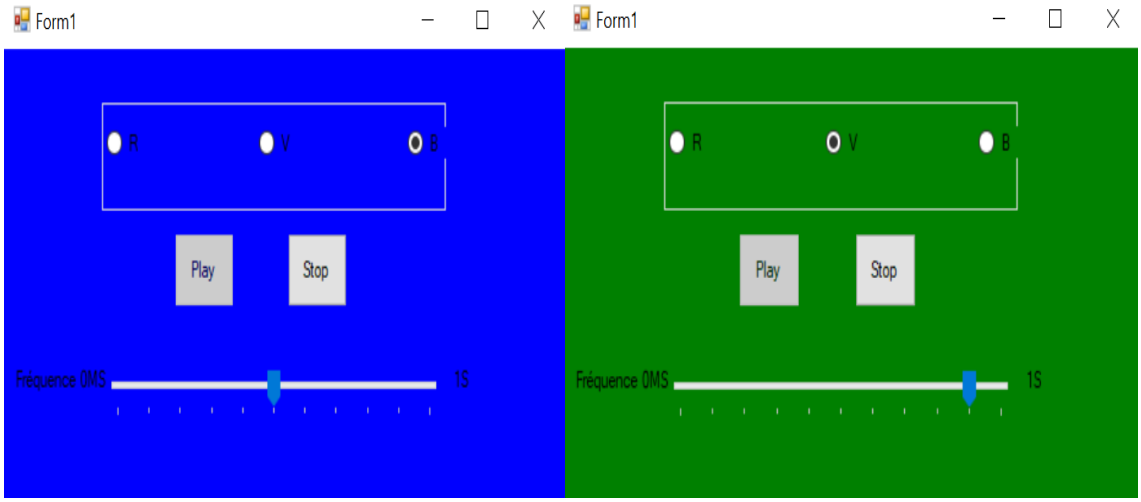
Create the following application:



The purpose of the application is to turn on colored lamps (check the **RADIOBUTTON**) at the rate of the frequency set in the **TRACKBAR** control.

- When the "**Play**" button is clicked, the lamps start to turn on and off automatically and in order (Red, Green, Blue) according to the frequency of the setting bar.
- When clicking on the "**Stop**" button, stop the action and reset the form.
- Enable and disable the various controls in the application so that they become accessible only when they can be used.

Execution example:



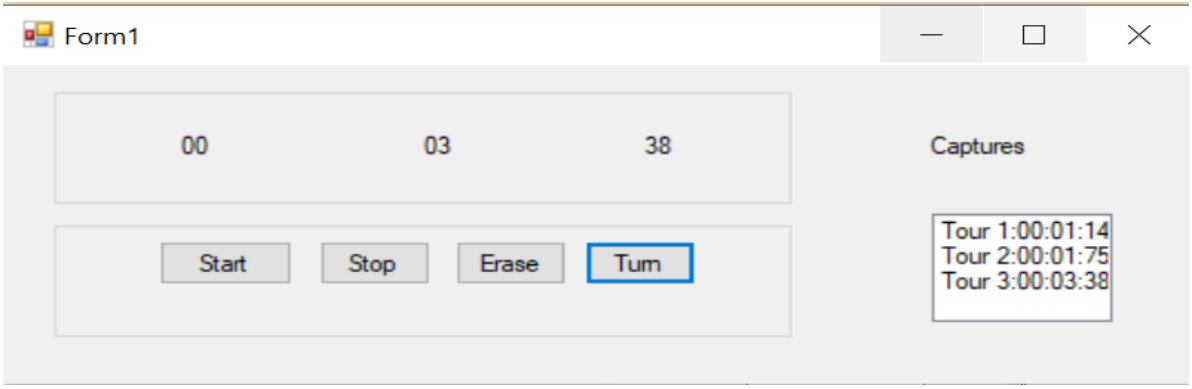
Exercise 6 :

Objective: Manipulate **TIMER** controls

Create the following C# "Stopwatch" application:

- The "Start" button is used to start the stopwatch.
- The "Stop" button allows you to pause the stopwatch.
- The "Erase" button resets the stopwatch and clears the list.
- The "Turn" button is used to capture the displayed value of the stopwatch.

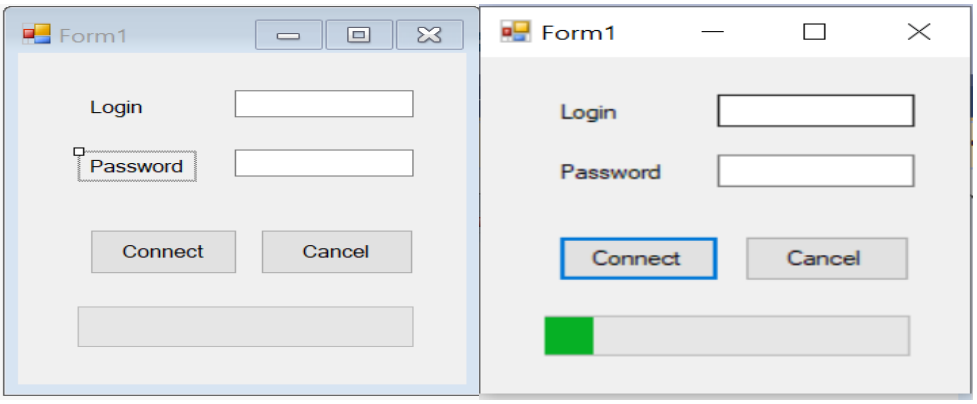
- **Execution example:**



Exercise 7 :

Objective: Manipulate advanced **TIMER**, **PROGRESSBAR** controls

Create the following C# application:



The user must enter their login and password in 30 seconds. If the user enters an incorrect code, a message box shows him "Wrong code", if he does not enter the information correctly in 30 seconds, a message box displays "Time is over" and the application stops. If he enters correct information a message box displays him "Welcome!".

The progress bar tells the user the level of time remaining.

Exercise 8 :

Objective: Work on different type of events, sender,…

Create a calculator with all the possible functionalities such as backspace, on/off, etc.

- Try to control your application by keyboard events. Your program should work by both keyboard and clicking on controls.

- Use sender in the possible parts of the code for example operators and numbers.