Form applications

How to create

Create new „Windows Forms App“

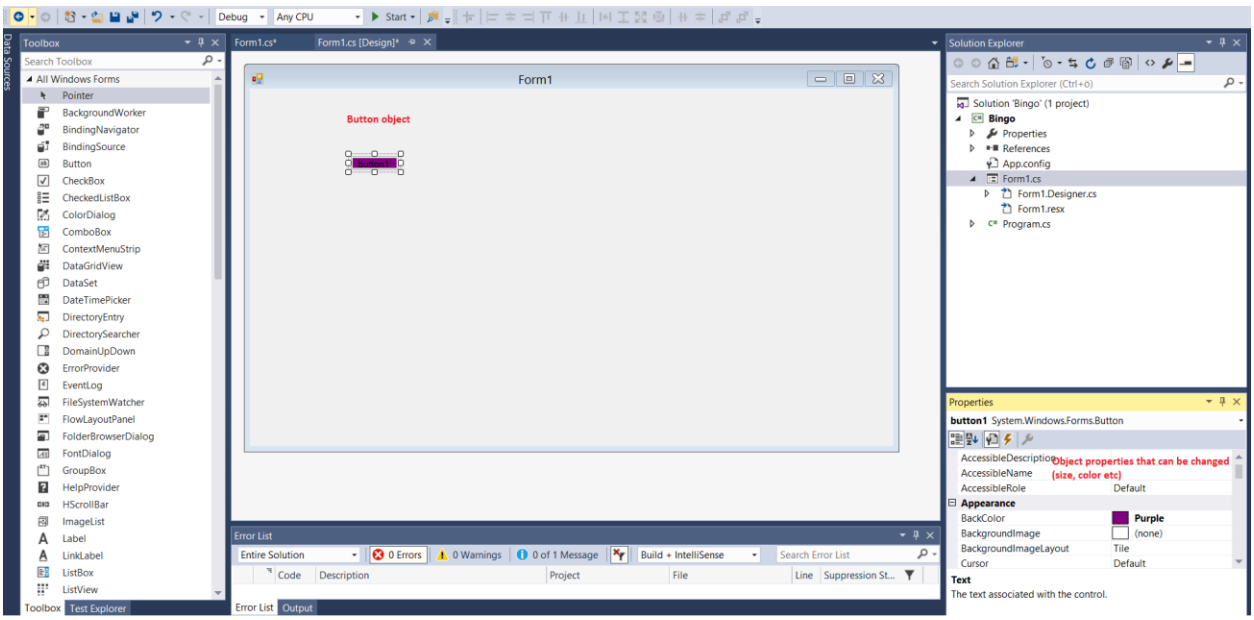
Toolbox: offers some built in controls that can be dragged onto the form (*View->Toolbox*)

Common controls:

- Button -> Creates button
- Label -> creates text label
- TextBox -> creates text box

Properties

If you choose the object on the form (with one mouse left click) then in the bottom right you can choose the color and appearance.



Events:

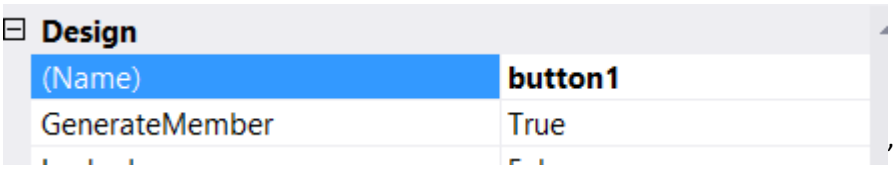
With objects there are connected events (OnChanged, OnClicked) where methods can be written what happen in this case. Events can be triggered with double click: if you double click on a button then „OnButtonChanged“ event is shown.

```
private void button1_Click(object sender, EventArgs e)
 {

 }
```

Form1_OnLoad() method – event of loading the form. All the code that should be executed when loading the form should be there.

Names:

All objects have names and can be called by these names. Names can be viewn in the bottom right properties container under the 'name' field. Names can be changed and objects can be called by names.



Create each exercise in a separate project within the same solution: right click on the solution and „add new project".

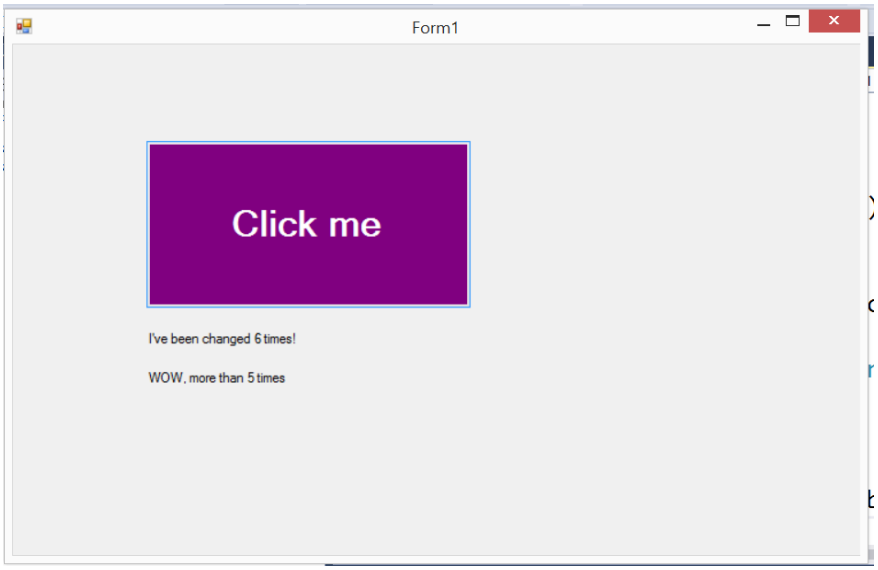
Exercise 1: button and label

- Add a button and 2 labels to the form (you can get them from toolbox). Change the names to „ButtonClick", „LabelClick" and „LabelHidden". Position the labels under the botton.
- Resize the button to big and change the backround color. Change the font.
- Change button text to "Click me" and first label text to empty and second one to „WOW, more than 5 times!". Hide the second label. (field 'text' is for text and 'visible' for hiding)
- Double click on the button and inside the method write:
- LabelClick.Text="Im a changed text!".
- Start your application and click on the button, you should see the text appear.
- Add counter and print out how many times the button has been pressed.

In the form add a new public property of type int with name counter and increase it with each button click.

- If the button has been pressed 5 times, then un-hide the second label.

  LabelHidden.visible=true or *LabelHidden.Show();*



- Add an onclick event to label:

```
private void LabelClick_Click(object sender, EventArgs e)
        {

        }
```

And in that method change the text of the label to : “Cheater!” and reset the counter to 0.

- Now we see that bottom label is visible even if we reset the counter to 0. (It should be visible only If the counter is bigger than 5). Add a condition to fix that.

Exercise 2: timer

New objects: Timer.

Timer allows to do smth repeatedly; example: update a number after every 1 second.

Timer has:

- Start – starts the timer
- Stop –stops the timer
- Tick – event for repeating something after a set time span.
- Interval – setting the time limit in ms. Example: *timer1.Interval = 1000;* is 1 second

Example:

```
public partial class Form1 : Form
    {
        int counter=1;

        public Form1() //loading the form
```

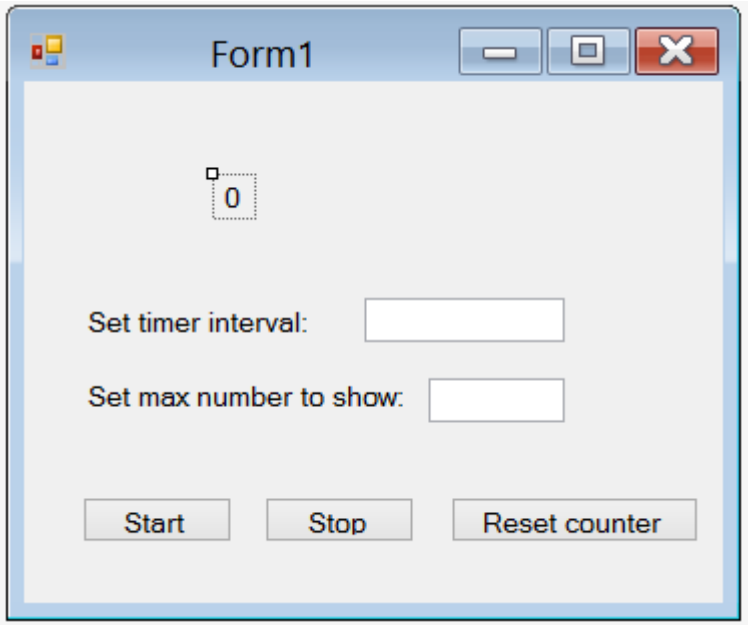
```
        {
            InitializeComponent();
            timer1.Interval = 1000; //timer interval 1 sec
            timer1.Start(); //we start the timer when form is loaded

        }

        //timer tick event; is done after every inteval, ie 1 sec
        private void timer1_Tick(object sender, EventArgs e)
        {
            label1.Text = counter.ToString();//we change the label text
            if (counter > 10)
            {
                //if the counter reaches 10, we stop the timer
                timer1.Stop();
            }
            //we increase the counter
            counter++;
        }
    }
```

Task:

create a form counting numbers from 0 to upwards. User can set the limit for max number and timer interval (the speed for changing the number). It should also have a button for starting and stopping and reseting the counter.

- If start button is pressed, the numbers start changing until the max number. If max number is reached, timer has to stop.
- If stop button is pressed, timer is stopped.
- If reset counter button is pressed, then counter value should be set to 0 and counting continue.

Input box control is ’text box’.



Exercise 3: lottery number generator

Create an application for lotter number generator.

- User should be able to choose how many numbers he or she wants.
- What is the maximum value for lotter numbers. If user chooses 30 then numbers from 1 to 30 are created.
- Application should display:
    - Current number
    - All available numbers
    - Results (what numbers has the machine used).

All numbers can be called out only once so if a number is once chosen as a lottery number, then remove it from availalbe numbers list and add it to used numbers (results) list.

Technical info and possible solution:

- 2 lists for storing the numbers: one for all available numbers and the other one for already used numbers. Used numbers should be removed from available numbers.
- Method for creating initial available numbers. For example; if loto numbers are from 1 to 30 then a list should be created with values from 1 to 30.
- Method for printing out list items. We want to display the numbers in a row with length of for so after printing out 4 items we should have a line break. Line break: „/n“ Example: string multipleRows = „im on the first row /n im on the second row “
- Method for getting next lottery number (current number to display). The number should be removed from available numbers list and added to used numbers list.

- In each timer tick event:
  - new loto number should be generated
  - updated list values (available numbers, used numbers) displayed.