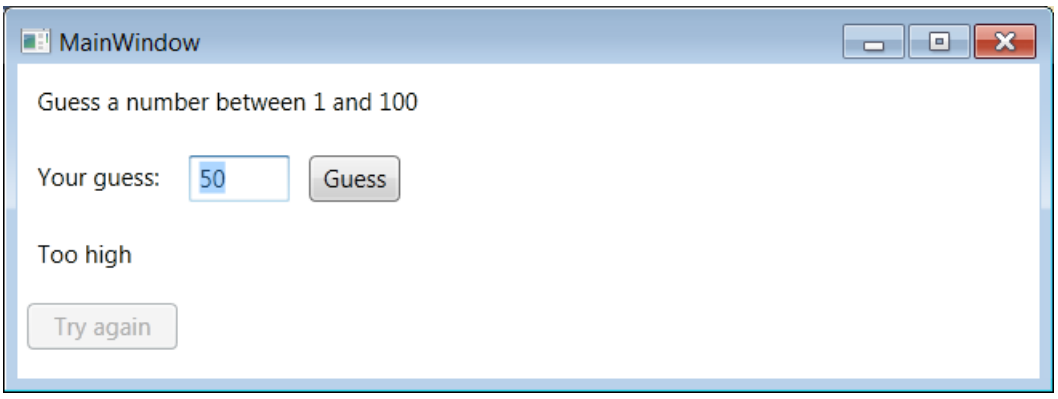
1) Guess a number

You a going to create a WPF application to guess a number. The GUI should look like this.



a)

Create a new WPF Application in Visual Studio.

b)

Add the controls in absolute positions.

- A Label
- Another Label, a TextBox and a Button (Guess) aligned horizontally
- A third Label
- Another button (Try again)

Set the content of the Labels and Buttons to the string shown in the figure. *Also remember to give all control a describing Name* (at least if you want to access the control from your code).

c)

Program a Loaded-event on the Window. This event should instantiates a random number generator (the class Random): Use the property windows for the Window to create a Loaded-event handler.

If you want help on the Random class: Place the keyboard cursor on the "Random" (type or constructor) and press F1.

Program the Click-events for the Buttons:

When the user makes a guess the lower label should show "Too low", "Too high" or "Correct. Congratulation!"

If the guess is correct both buttons should change their isEnabled-property.

If the guess is incorrect the entire text in the TextBox should be selected,such that the user does not have to clear the TextBox before typing a new value. Also give the TextBox the focus.

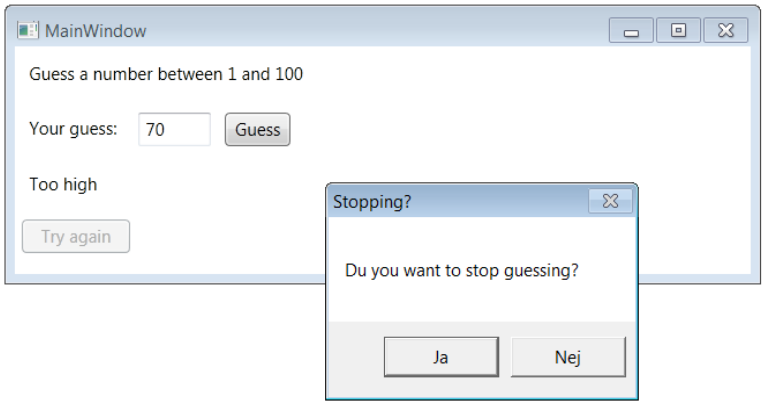
Program the “Try again”-button.

If the TextBox does not have focus when the program starts, then see that is the will be the case.

d)

Add a Closing-eventhandler on the Window.

If the user has not finished a guessing then the user should be asked if he want to close the application:



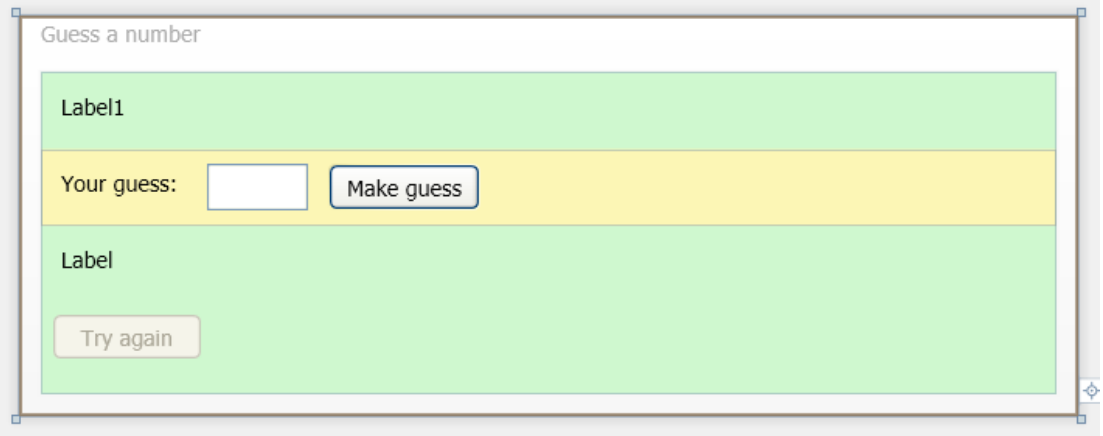
You show a Messagebox and test which button the user clicks this way:

```
MessageBoxResult mr = MessageBox.Show("Do you want to stop guessing?",
                                      "Stopping?", MessageBoxButton.YesNo);

if (mr==MessageBoxResult.Yes) ... ; else ...;
```

1b) Same exercise but with a better layout

Now the layout of the exercise should be changed to be more flexible by using StackPanels



Start a new WPF project.

Delete the Grid-control in the Window:

Go to the XAML code an delete

```
<Grid>

</Grid>
```

Instead insert a StackPanel (the light green area, including the ligth yellow area)

In this StackPanel you insert

- A Label
- A new StackPanel (the light yellow area)
- A Label
- A Button (“Try again”)

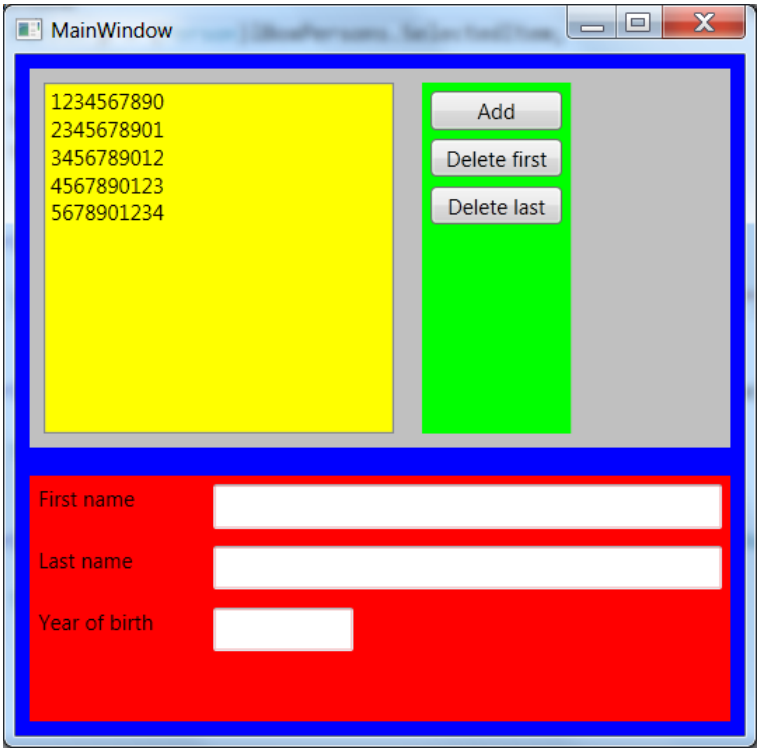
In the new StackPanel:

- Set orientation to “Horizontally”
- Then insert
    - A Label
    - A textbox
    - A Button (“Guess”)

Set Margin for all Control in order to make space between them. A StackPanel can also have a Margin.

2) Databinding and List

You are supposed to create a GUI like shown below. *You don't need to set the background colors, it is only for explanation in this exercise.*



Outermost in the Windows there is a (blue) vertical StackPanel.

This contains another (silver) horizontal StackPanel

Which contains a (yellow) ListBox, a (green) vertical StackPanel

The green StackPanel contains three Buttons

Finally the outermost StackPanel contains a (red) Grid with 2 columns and 3 rows

The Grid contains Label in the left column and TextBoxes in the right column.

a) Now Create the GUI.

b) Add a Class to the project (Right-click the project in the SolutionExplorer) and choose

Add → Class

Copy the following code to the new class-file:

```
class Person
{
  string cpr;
  public string Cpr
  {
    get { return cpr; }
```

```
      set { cpr = value; }
    }

    string firstName;
    public string FirstName
    {
      get { return firstName; }
      set { firstName = value; }
    }

    string lastName;
    public string LastName
    {
      get { return lastName; }
      set { lastName = value; }
    }

    int yearOfBirth;

    public int YearOfBirth
    {
      get { return yearOfBirth; }
      set { yearOfBirth = value; }
    }
  }

  class Service
  {
    public static ObservableCollection<Person> list = new ObservableCollection<Person>();

    static Service()
    {
      list.Add(new Person
            { Cpr = "12345", FirstName = "Bill", LastName="Smith",YearOfBirth=1980});
      list.Add(new Person
            { Cpr = "23456", FirstName = "Chris", LastName = "Larsson", YearOfBirth = 1988 });
      list.Add(new Person
            { Cpr = "34567", FirstName = "Susan", LastName = "Ford", YearOfBirth = 1995 });
      list.Add(new Person
            { Cpr = "45678", FirstName = "Jane", LastName = "Doe", YearOfBirth = 1977 });
      list.Add(new Person
            { Cpr = "56789", FirstName = "Bill", LastName = "Doe", YearOfBirth = 1986 });
    }

  }
```

`ObservableCollection` is in the namespace `System.Collection.ObjectModel`

Add this in the using list at the top.

Create a Loaded event for the Window:

```
    private void Window_Loaded(object sender, RoutedEventArgs e)
    {
      lBoxPersons.ItemsSource = Service.list;
      lBoxPersons.DisplayMemberPath = "Cpr";
    }
```

Run the project a see that the ListBox in filled with 5 Person Items. Only the Cpr is visible as specified in the DisplayMemberPath property of the ListBox.

c) Add a SelectionChanged eventhandler to the ListBox and add the following lines to it

```
private void lBoxPersons_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
  Person p = (Person)lBoxPersons.SelectedItem;

  tBoxFirstName.Text = p.FirstName;
}
```

Changes the name of the TextBox for the FirstName if is not tBoxFirstName.

Run the project to see that the textbox with the first name follows the selected item.

d) Do the same with the two other TextBox'es.

e) Program the Click event handler for the Add button to Add a new Person to the static Service.list. The values for the new Person might be hardcoded. (Eventually see the static constructor of Service how to Add a new Person.)

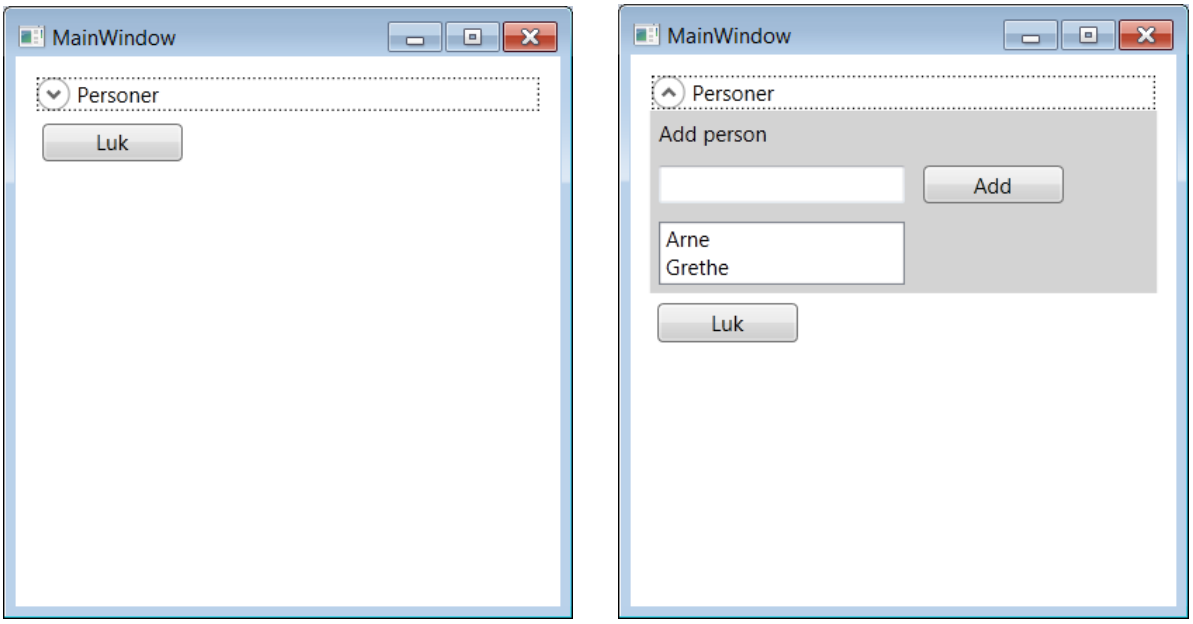
f) Program the two other Buttons.

g) **If you have time left:** Changes the Add Button such that the values for the new Person is read from the TextBox'es. You need to add a textbox for the Cpr value.

Remark: list in Service ought to be a get-property and not a field.

3) Expander

Make a WPF application containing an Expander, which is a control with a Header and a Content. The Content can be collapsed and expanded:



The Content can only contain one Control. The controls in the content is Label, TextBox and ListBox where a few Items are added in the designer (or XAML).

Make a layout as shown above. (In the example the Expanderen has Background set to LightGray)

You must see that the layout can grow like this: