

Day 04 .NET and C# Exercises

1) Events uden GUI

Start from DelegateEks3 on WeatherStation.

a) Extend the WeatherStation such that it can activate an alarm for hot temperature:
Add a new delegate type for the heat alarm. Find a suitable name. Declare an event of this delegate type.

If the temperature exceeds 35 degree the event is fired.

To be sure it happens, changes the line

```
temp += rnd.Next(11) - 5;
```

in getNewTemp() to

```
temp += 6;
```

You decide yourself if the new delegate for heat alarm should have parameters. It could be temperature or time (a DateTime object).

Add a new class to handle the heat alarm (it should only display a text f.ex. "Heat Alarm").

b) Extend the WeatherStation such that it in addition to temperature also measures wind force and pressure.

Wind force 0-40 m/s are realistic values.

Pressure 960 – 1040 hPa (hektoPascal) are realistic values.

Add a new delegate type with three parameters for temperature, wind force and pressure and an event variable of this type.

Make a new method in WeatherStation, GetWeatherData(), which sets suitable values for temperature, wind force and pressure and call the new event.

Make a new class for receiving the three values and display them on the Console.

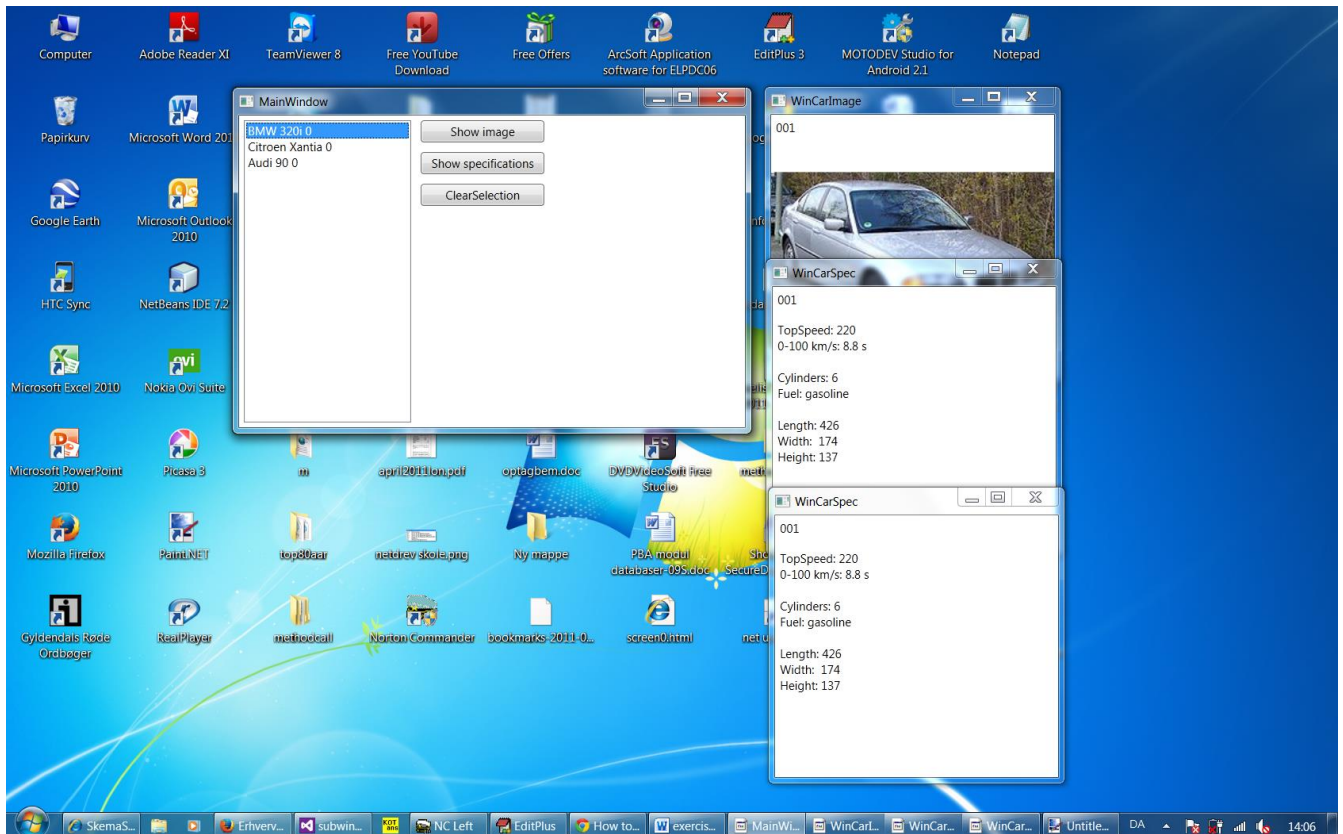
Call GetWeatherData() a number of times in Main.

c) Let MinMaxDispaly use the new event from b) instead of the event, SendTemp.

2) Events

Download `opg2start.zip` from Fronter, unzip, load and run in VisualStudio.

The program looks like this after a couple of click on Show-Buttons:



When the selection in the listbox in the MainWindow is changed we want all subwindows with image or specification to update their content to show image or specification for the new car selected.

Implement this by adding an event to the MainWindow which the subwindows can add themselves to.

Please note that the Property Owner are set in SubWindows. Therefore in a subwindow you can get a reference to MainWindow like this: `(this.Owner as MainWindow)` or `(MainWindow)this.Owner`.

3) Inserting controls at runtime

You are supposed to make a WPF application to show pictures in a WPF application. The main window looks like



a)

Make a new WPF Application and let Window contain a ScrollViewer with a StackPanel.

When the Window is loaded (in the loaded event), you must scan a folder with the pictures:

```
string[] files=System.IO.Directory.GetFiles(@"c:\images","*.jpg");
```

(If you are missing images you can unzip images.zip on Fronter.)

Also *in code* - in the loaded event - insert an Image in the StackPanel for each file. Set the size of the images to 100x100.

If the Image control is called img, the you set the image source like this:

```
Uri uri = new Uri(files[n]);  
ImageSource imgSource = new BitmapImage(uri);  
img.Source = imgSource;
```

Test the Application.

b)

When the user make a MouseUp on a image this image should be shown in a new Window in full Size:



The user made a `MouseButtonUp` on the white BMW

Now add an event handler to the `MouseButtonUp` event of each image.

The event handler for this event must do the following

- Create a new Window
- Insert an Image in the Window's Content
- Set the `ImageSize` to the actual size of the image which was clicked
- Set the `ImageSource` in the Image
- Show the new window as a Dialog

You are faced with one problem: How does the `MouseButtonUp` event handler get the filename for the image to show?

One easy solution is this: All Controls including an Image has a property called `Tag` of type object. Hence you can attach any object to a control with the information needed. Add a `Tag` to all the small images by setting the `Tag` to the filename.

In the event handler of the `MouseButtonUp` event you get the image in the sender parameter from which you get the `Tag` object:

```
Image ctrl = sender as Image;
string filename = ctrl.Tag as string;
```

If you want the new Window to get the size of its Content you must set the `SizeToContent` property for the Window:

```
win.SizeToContent = SizeToContent.WidthAndHeight;
```

If you want the Window to open in a specific position you must set its `Left` and `Top` property.