Exercise – Draw Polypoints

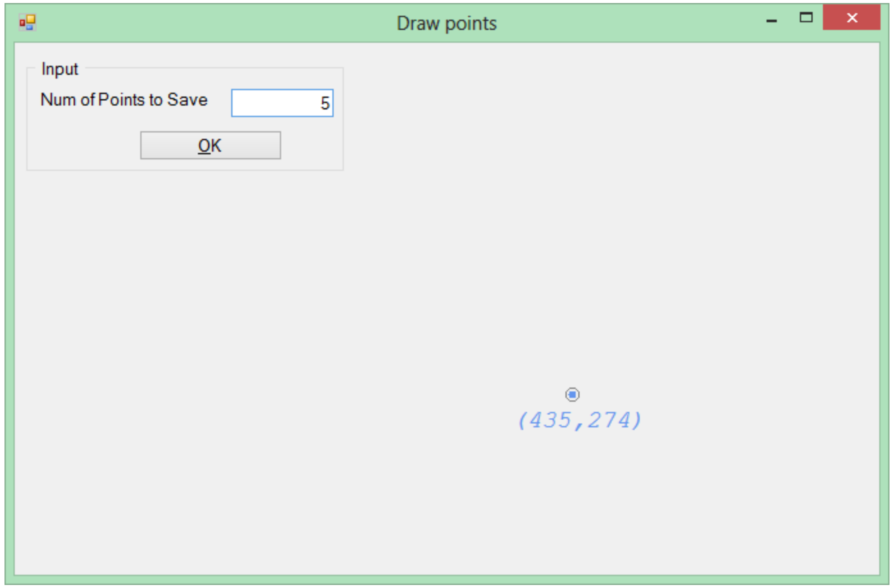
1. Objectives

The objectives of this exercise are to practice with the following:

- One-dimensional arrays
- Structs

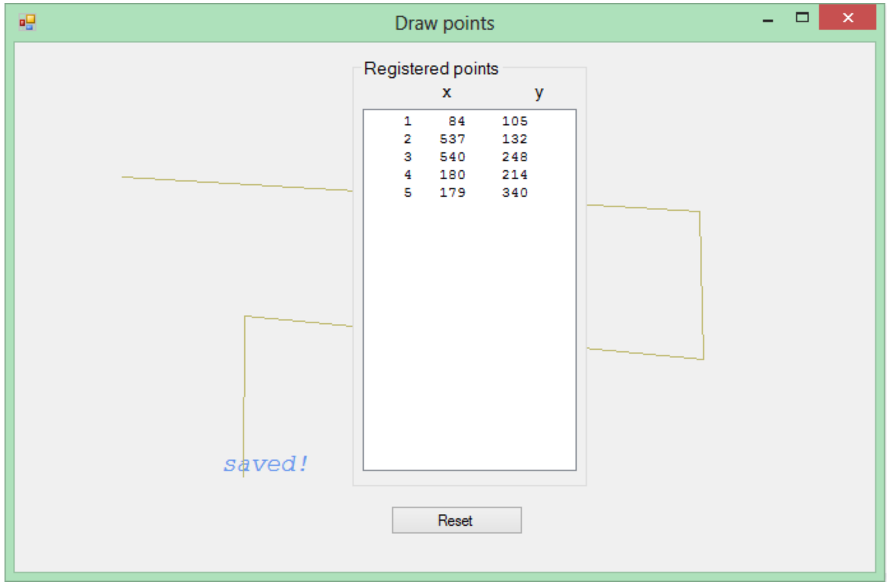
2. Description

Write an application that keeps track of the mouse-clicking on a form. The user gives the number of points to be saved in an array, as in the image at left. After clicking the OK-button, a listbox is displayed and the coordinates (x, y) of the every clicking (up to the size of the array) on the form is recorded in the array. Meanwhile the, the points are draw on the drawing space of the form as in the image at right.



Num of Points to Save: 5

(435,274)



	x	y
1	84	105
2	537	132
3	540	248
4	180	214
5	179	340

saved!

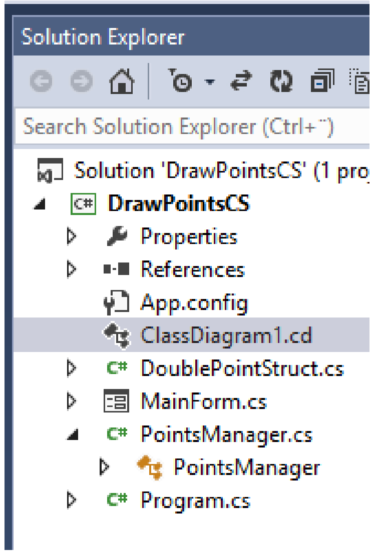
3. Instructions

Follow the class diagram given below.

3.1 Define a struct as shown in the class diagram below.

3.2 Write a class PointManager in which you create an array of the struct.

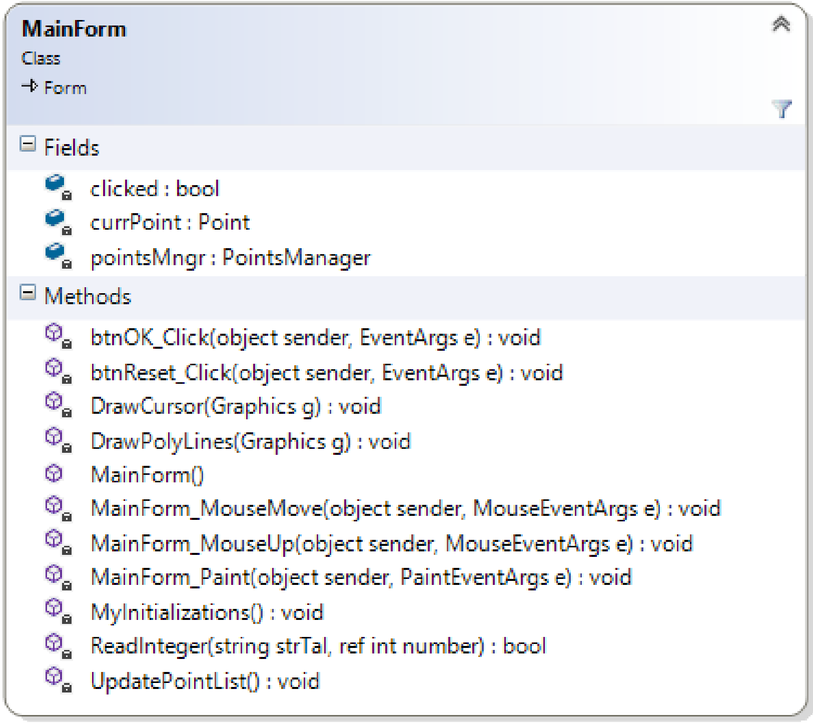
3.3 Handle the Mouse-up and Paint events of the form to perform the drawing.



Solution Explorer

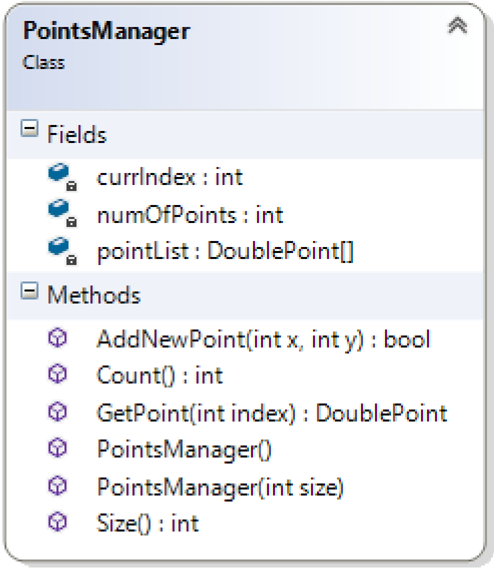
Search Solution Explorer (Ctrl+¨)

```
Solution 'DrawPointsCS' (1 pro
◢ DrawPointsCS
    ▷ Properties
    ▷ References
      App.config
      ClassDiagram1.cd
    ▷ DoublePointStruct.cs
    ▷ MainForm.cs
    ◢ PointsManager.cs
        ▷ PointsManager
    ▷ Program.cs
```



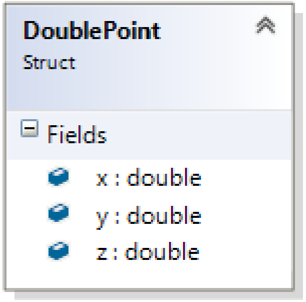
```
MainForm
Class
→ Form

⊟ Fields
    clicked : bool
    currPoint : Point
    pointsMngr : PointsManager
⊟ Methods
    btnOK_Click(object sender, EventArgs e) : void
    btnReset_Click(object sender, EventArgs e) : void
    DrawCursor(Graphics g) : void
    DrawPolyLines(Graphics g) : void
    MainForm()
    MainForm_MouseMove(object sender, MouseEventArgs e) : void
    MainForm_MouseUp(object sender, MouseEventArgs e) : void
    MainForm_Paint(object sender, PaintEventArgs e) : void
    MyInitializations() : void
    ReadInteger(string strTal, ref int number) : bool
    UpdatePointList() : void
```



```
PointsManager
Class

⊟ Fields
    currIndex : int
    numOfPoints : int
    pointList : DoublePoint[]
⊟ Methods
    AddNewPoint(int x, int y) : bool
    Count() : int
    GetPoint(int index) : DoublePoint
    PointsManager()
    PointsManager(int size)
    Size() : int
```



```
DoublePoint
Struct

⊟ Fields
    x : double
    y : double
    z : double
```

4. Help and solution

4.1 Usually, it is not the form on which you would want to make your drawings. A Panel controls very appropriate to draw on. It will then be the Panel's Paint-event that you should use to draw.

4.2 Note also that you should not call the Paint-handler method to redraw. Controls have a method called Invalidate() that causes a redraw. So whenever you want to refresh your drawing, call the Invalidate() method (using a panel, call the panel's Invalidate).

4.3 The Paint-handler method has a parameter **e** that contains valuable information. It has an object of Graphics that you can use for drawing lines, and other figures (and even strings).

```
private void MainForm_Paint(object sender, PaintEventArgs e)
```

4.4 The MouseUp-handler method also has such a method parameter **e** that maintains the coordinate of the mouse position.

```
private void MainForm_MouseUp(object sender, MouseEventArgs e)
```

4.5 A suggested solution is given via the course's Its L. Download the project and run the application without looking at the source code to see how it works. Try then to create a similar program.

Good Luck!

Programming is fun. Never give up. Ask for help!

Farid Naisan,

Course Responsible and Instructor