Summer Training 2009

Second Year

OOD: design and develop better

Lab3

# Part0: Review class diagram until now!!

1. Two new members are added to the **Shape**: **PenColor** & **PenWidth**
2. Default constructor is added to the **Shape** to initialize the new members by default values

# Part1: Separate the Shapes Data Structure from the GUI

1. Explain what the benefits of such separation?!

## Hands-on: Make a standalone library (dll) for the shapes

1. File🡪Add New Project
2. Visual C# 🡪Class Library 🡪Name: **ShapesLib**
3. Move the .cs files (Shape, Rect, Ellipse, Circle) from **Section3** project to **ShapesLib** project and change their namespace to **ShapesLib**
4. Makesure that all classes in the library are **public** to be able to access them outside the assembly file of the library
5. In **ShapesLib** project: Project🡪Add Reference🡪.NET🡪 System.Drawing
6. Right click on **ShapesLib** project and choose **Build**

**Now:** you get **ShapesLib.dll** file in the **Bin** folder of the **ShapesLib** project. You can use this **dll** in any project you want by simply adding it as a Reference to your project

1. In **Section3** project: Project🡪Add Reference🡪Browse🡪"Bin\ShapesLib.dll"
2. **using ShapesLib** as **namespace** wherever you need to define a shape

# Part2: Draw the Shape by Any Size (using mouse drag and drop)

* 1. Add **DrawForCreation** method to **Shape** which will be used to temporary draw the shape during the creation (using certain color and pen)
  2. Implement this method in each shape and rebuild the **ShapesLib**
  3. Add **tmpShape** member to **Form1** which will temporary hold the currently created shape and initialize it to **null**
  4. Draw this **tmpShape** (if it's exist!!) in the **Paint** event using **DrawForCreation**
  5. Handle the **mouseDown** event to:
     1. store the first point
     2. create a new **tmpShape** with default minimum size
  6. Handle the **mouseMove** event to update the **tmpShape** with the new dimensions
  7. Handle the **mouseUp** event to finish drawing the shape simply by adding the **tmpShape** to the **allShapes** list which make it a permanent shape
  8. DON'T FORGET to remove the creation logic from the **mouseClick**

## Hands-on: Translate any shape using mouse drag and drop

You should be able to translate any shape by first choosing the **Select** button on the toolbar and then use the mouse to drag and drop the shape you want to translate.

Focus on the following:

1. Add **Translate** method to **Shape** as a **virtual** function and implement it since some shapes need to write its own translation logic (e.g. polygon which translate each point in it)
2. Move the logic of shape selection from **mouseClick** to **mouseDown** since the shape should be selected first before you can translate it

# Part3: Add Polygon Shape and Draw it using Mouse Clicks

## Hands-on:

Given the **Polygon** class that is inherited from **Shape** class and contains the implementation of all **abstract** methods, DO:

* 1. Add this shape to your shapes library
  2. Add a GUI button for it
  3. Handle the drawing of this shape using **mouseClick** to specify the points of the polygon (each **mouseClick** represents a new point in the polygon)

(Hint: use the **DrawForCreation** method to draw the polygon lines during the creation of it)

* 1. To finish the drawing, use **mouseDoubleClick**

**SHOW THAT:** After this, you will be able to select, translate and change the polygon color without writing any extra code.

# TODO Task for students:

Add to your project a free-Hand polygon shape which is same as the polygon but can contains a free-hand drawing **in addition** to the normal polygon lines, see the following figure for an example of such shape.

**To start the free-hand drawing**, the user should push the mouse button and drag it over the form… all points during the dragging should be added to the polygon… this will make the effect of free-hand drawing.

On the other hand, **to draw the normal polygon line**, the user should click the mouse at the points of the lines… only clicked points should be added to the polygon (same as normal polygon).