Solutions:

**Shape.java-**

* I found creating this class straight forward and followed the instruction exactly. The constructor needed to take in three parameters in order to initialize the characteristics of the shape. The only three characteristics that made sense were rectangle, solid, and color. At first it did not occur to me to use rectangle in the parameter, but then I realized that Rectangle is what can hold height, width, x and y. Which makes it an object that holds the characteristic of the shapes size. I used setRect after looking through the java docs on Class Rectangle and realized from there that setRect is what I needed to initialize the rectangle object. At first, I tried to create a new rectangle object, but this did not work.

**Oval.java**-

* Here Oval inherits things from the parent class shape, so I have oval taking in all the parameters that shape does along with a super statement. My draw method takes in argument graphics g, sets the color to the color g has and then has an if else statement which relies on the Boolean solid to determine whether to fill or draw oval.

**Rectangular.java-**

* Follows the same process as oval, but with drawRect and fillRect to draw a rectangle instead.

**Drawing.java-**

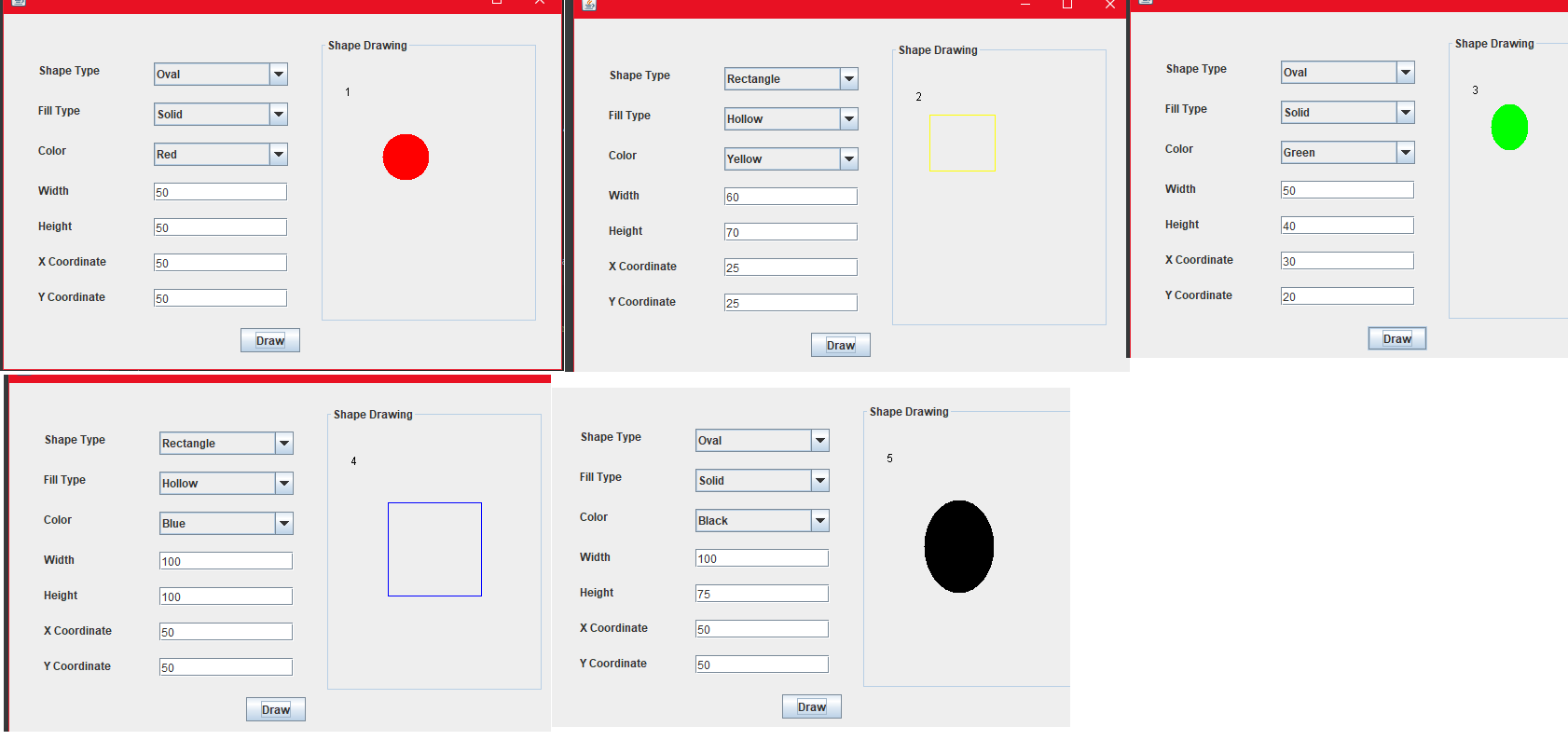
* Here shape is drawn on to the graphics object and checked to ensure it is within bounds of the preferred size that is also set here. The main part of it is the drawShape method which takes the shape to be drawn and checks if it is within the bounds of the preferred size. I achieved this by using the contains class which checks first the upper left point of the shape is inside the bounds and then checks the bottom right corner. If contains is true repaint() is called to paint the shape on the panel. If the shape is out of bounds it throws my exception.

**OutsideBounds.Java-**

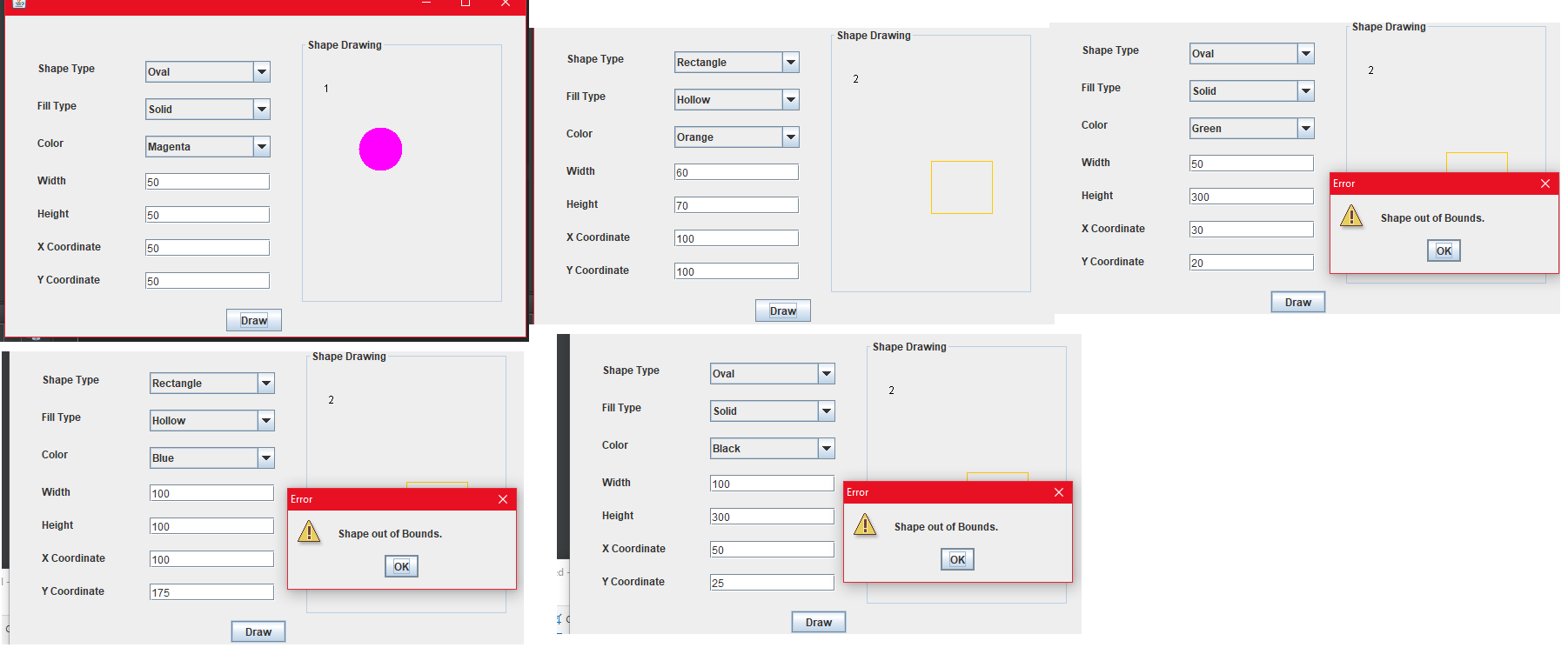
* Here my OustideBounds exception extends Exception and takes String error as an argument.

**Project3.java –**

* Here I created variables to take color height and hollow from the text fields and combo boxes. I have Boolean hollow set to true, then it turns to false if Solid is selected in the combo box to either fill or draw the shape. I also have a switch statement to take in the string name of the color selected and to set the color to the Color that it represents. I also have a try statement that sets the parameters of the shape, and catches a number format exception if user does not enter a number. Then my Rectangle object shapes is created with the parameters from the text fields. After that I have an if else statement to either create a rectangle shape with Rectangle object shapes or create an oval shape with Rectangle object shapes, based of the combo box selection. Whichever is selected a new drawing is created and the shape is added to Drawing panel. Remove all required to remove previous shape, add ads the drawing and repaint actually paints it to the panel.

Test Case 1: Create 5 shapes, in bounds.

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expected output | Output | Pass - Fail |
| 1  (Red ,Oval, Solid, 50, 50, 50, 50)  2  (Yellow, Rectangle, Hollow, 60, 70, 25, 25)  3  (Green, Oval, Solid, 50, 40, 30, 20)  4  (Blue, Rectangle, Hollow, 100, 100, 50, 50)  5  (Black, Oval, Solid, 100, 75, 50, 25) | 5 Shapes exactly as described, with counter reaching 5. | See below | pass |

Test Case 2: Create 5 shapes, 3 in bounds and 2 not

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expected output | Output | Pass - Fail |
| 1  (Magenta, Oval, Solid, 50, 50, 50, 50)  2  (Orange, Rectangle, Hollow, 60, 70, 100, 100)  3  (Green, Oval, Solid, 50, 300, 30, 20)  4  (Blue, Rectangle, Hollow, 100, 100, 100, 175)  5  (Black, Oval, Solid, 100, 300, 50, 25) | 2 Shapes exactly as described, with counter reaching 2. 3 shapes not printing due to out of bounds. | See below | pass |

Test Case 3: Make duplicate shapes and max size shapes.

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expected output | Output | Pass - Fail |
| 1  (Magenta, Oval, Solid, 50, 50, 50, 50)  2  (Magenta, Oval, Solid, 50, 50, 50, 50)  3  (Orange, Rectangle, Hollow, 60, 70, 100, 100)  4  (Orange, Rectangle, Hollow, 60, 70, 100, 100)  5  (Green, Oval, Solid, 50, 35, 30, 20)  6  (Green, Oval, Solid, 50, 35, 30, 20)  7  (Blue, Rectangle, Hollow, 199, 199, 0, 0)  8  (Black, Oval, Solid, 199, 199, 0, 0) | 8 Shapes exactly as described. | See below | pass |



Full screen shot of code running:

