

Programming Project 3

The third programming project involves writing a program that draws two types of shapes, ovals and rectangles. This program consists of six classes. The first class is the `Shape` class, which is an abstract class that extends the predefined Java class `Rectangle`. It should contain two instance variables, the color of the shape and whether the shape is solid or hollow. It should also contain a class (static) variable that keeps track of how many shapes have been created.. It should have three instance methods, one class method and one abstract method:

1. A constructor that accepts three parameters for the purpose of initializing the characteristics of the shape, a `Rectangle` object that defines the dimensions and position of the shape, the color of the shape and whether the shape is solid or hollow. It should also update the number of shapes created so far.
2. An instance method named `setColor` that accepts the `Graphics` object as a parameter and sets the color for the next draw operation to the color of the current shape.
3. An instance method named `getSolid` that returns whether the shape is solid or hollow.
4. A class method named `getNoOfShapes` that returns the number of shapes created so far.
5. An abstract method named `draw` that accepts a `Graphics` object as a parameter.

The `Shape` class has two subclasses. The first is `Oval`. It should have the following two methods:

1. A constructor that accepts three parameters for the purpose of initializing the characteristics of the shape, a `Rectangle` object that defines the dimensions and position of the shape, the color of the shape and whether the shape is solid or hollow.
2. An overridden method `draw` that draws the `Oval` object on the `Graphics` object passed as a parameter.

The second subclass is `Rectangular`. It should have the following two methods:

1. A constructor that accepts three parameters for the purpose of initializing the characteristics of the shape, a `Rectangle` object that defines the dimensions and position of the shape, the color of the shape and whether the shape is solid or hollow.
2. An overridden method `draw` that draws the `Rectangular` object on the `Graphics` object passed as a parameter.

The fourth class is named `Drawing`, which should extend the predefined Java class `JPanel`. It has one instance variable that contains the shape that is currently drawn.. It should have three methods:

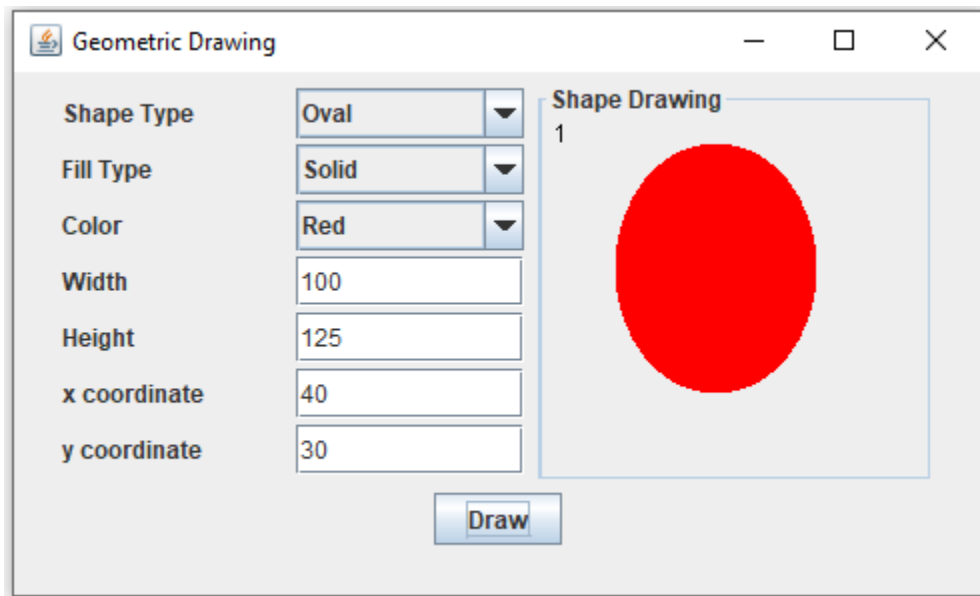
1. An overridden `paintComponent` method that draws the current shape on the `Graphics` object that is passed to it as a parameter. It should also draw the number of shapes that have been created thus far in the upper left corner.
2. An overridden `getPreferredSize` method that specifies the dimensions of the drawing panel as 200 pixels wide and 200 pixels high.

3. An instance method named `drawShape` that is passed the current shape to be drawn. It first checks whether the shape provided will completely fit within the panel. If not, it throws an `OutOfBounds` exception. Otherwise, it saves the shape in the corresponding instance variable. It then calls `repaint` to cause that shape to be drawn.

No additional public methods should be included in any of the above classes.

The fifth class named `OutOfBounds` should define a checked exception.

The sixth class named `Project3` should contain the main method. It should generate the GUI shown below:



The combo-box for the shape type should allow two choices, either *Rectangle* or *Oval*. The combo-box for the fill type should also allow two choices, either *Hollow* or *Solid*. The combo-box for the color should allow seven choices, *Black*, *Red*, *Orange*, *Yellow*, *Green*, *Blue* or *Magenta*.

Clicking the *Draw* button should first check whether any non integer values have been entered in any of the fields that require integers. If so, an error message should be displayed in a `JOptionPane` window. Otherwise an appropriate `Shape` object should be created and passed to the `drawShape` method of the `Drawing` class. If that call results in an `OutOfBounds` exception being thrown, an appropriate error message should be displayed in a `JOptionPane` window.

Be sure to follow good programming style, which means making all instance variables private, naming all constants and avoiding the duplication of code. Furthermore you must select enough different kinds of shapes to completely test the program.

Grading Rubric:

The following grading rubric will be used to determine your grade:

Attribute	Meets
Shape class	<p>15 points</p> <p>an abstract class that extends the predefined Java class Rectangle. It should contain two instance variables, the color of the shape and whether the shape is solid or hollow. It should also contain a class (static) variable that keeps track of how many shapes have been created.. It should have three instance methods, one class method and one abstract method:</p> <ol style="list-style-type: none">1. A constructor that accepts three parameters for the purpose of initializing the characteristics of the shape, a Rectangle object that defines the dimensions and position of the shape, the color of the shape and whether the shape is solid or hollow. It should also update the number of shapes created so far.2. An instance method named setColor that accepts the Graphics object as a parameter and sets the color for the next draw operation to the color of the current shape.3. An instance method named getSolid that returns whether the shape is solid or hollow.4. A class method named getNoOfShapes that returns the number of shapes created so far.5. An abstract method named draw that accepts a Graphics object as a parameter.
Oval class	<p>15 points</p> <p>It should have the following two methods:</p> <ol style="list-style-type: none">1. A constructor that accepts three parameters for the purpose of initializing the characteristics of the shape, a Rectangle object that defines the dimensions and position of the shape, the color of the shape and whether the shape is solid or hollow.2. An overridden method draw that draws the Oval object on the Graphics object passed as a parameter.

Rectangular Class	<p>15 points</p> <p>It should have the following two methods:</p> <ol style="list-style-type: none"> 1. A constructor that accepts three parameters for the purpose of initializing the characteristics of the shape, a Rectangle object that defines the dimensions and position of the shape, the color of the shape and whether the shape is solid or hollow. 2. An overridden method draw that draws the Rectangular object on the Graphics object passed as a parameter.
Drawing Class	<p>15 points</p> <p>should extend the predefined Java class JPanel. It has one instance variable that contains the shape that is currently drawn.. It should have three methods:</p> <ol style="list-style-type: none"> 1. An overridden paintComponent method that draws the current shape on the Graphics object that is passed to it as a parameter. It should also draw the number of shapes that have been created thus far in the upper left corner. 2. An overridden getPreferredSize method that specifies the dimensions of the drawing panel as 200 pixels wide and 200 pixels high. 3. An instance method named drawShape that is passed the current shape to be drawn. It first checks whether the shape provided will completely fit within the panel. If not, it throws an OutsideBounds exception. Otherwise, it saves the shape in the corresponding instance variable. It then calls repaint to cause that shape to be drawn.
OutsideBounds class	<p>10 points</p> <p>defines a checked exception</p>
Project 3 class	<p>15 points</p> <p>should contain the main method. It should generate the GU.</p> <p>The combo-box for the shape type should allow two choices, either Rectangle or Oval. The combo-box for the fill type should also allow two choices, either Hollow or Solid. The combo-box for the color should allow seven</p>

	<p>choices, Black, Red, Orange, Yellow, Green, Blue or Magenta.</p> <p>Clicking the Draw button should first check whether any non integer values have been entered in any of the fields that require integers. If so, an error message should be displayed in a JOptionPane window. Otherwise an appropriate Shape object should be created and passed to the drawShape method of the Drawing class. If that call results in an OutOfBounds exception being thrown, an appropriate error message should be displayed in a JOptionPane window.</p>
Documentation and style	<p>15 points</p> <p>Follow good programming style, which means making all instance variables private, naming all constants and avoiding the duplication of code.</p> <p>Furthermore you must select enough different kinds of shapes to completely test the program</p>