Department of Computer Science
The City College of CUNY

CSc 22100: Software Design Laboratory [Fall 2019]

Exercise 1

A printed report showing the problem, solution methods, codes developed, and outputs produced for the assignment indicated is due during and before the end of the class on Tuesday, 10 October 2019. The deadline is strictly observed.

1- Create a hierarchy of Java classes as follows:

MyLine *is_a* MyShape;
MyRectangle *is_a* MyShape.
MyOval *is_a* MyShape;

class MyShape:

Class MyShape is the hierarchy's superclass and inherits the Java class Object. An implementation of the class defines a point (x, y) and the color of the shape. The class includes appropriate class constructors and methods, including methods that perform the following operations:

a. *getX, getY, getColor* – returns the point (x, y) and color of the MyShape object;
b. *setX, setY, setColor* – sets the point (x, y) and color for the MyShape object;
c. *toString* – returns the object's description as a String. This method must be overridden in each subclass in the hierarchy.
d. *draw* – draws a MyShape object. This method must be overridden in each subclass in the hierarchy. For the MyShape object, it paints the drawing canvas in color.

class MyLine:

Class MyLine inherits class MyShape. The MyLine object is a straight line defined by its two endpoints (x_1, y_1) and (x_2, y_2), and may be of any color. The class includes appropriate class constructors and methods, including methods that perform the following operations:

a. *toString* — returns a string representation of the MyLine object: length and angle with the x-axis;
b. *draw* — draws a MyLine object $[(x_1, y_1), (x_2, y_2)]$.

class MyRectangle:

Class MyRectangle inherits class MyShape. The MyRectangle object is a rectangle of height h and width w, centered at a point (x, y), and may be filled with a color. The class

includes appropriate class constructors and methods, including methods that perform the following operations:

c. *toString* — returns a string representation of the MyRectangle object: width, height, perimeter, and area;
d. *draw* — draws a MyRectangle object of height h and width w, centered at a point (x, y). The center point of the rectangle is defined in class MyShape.

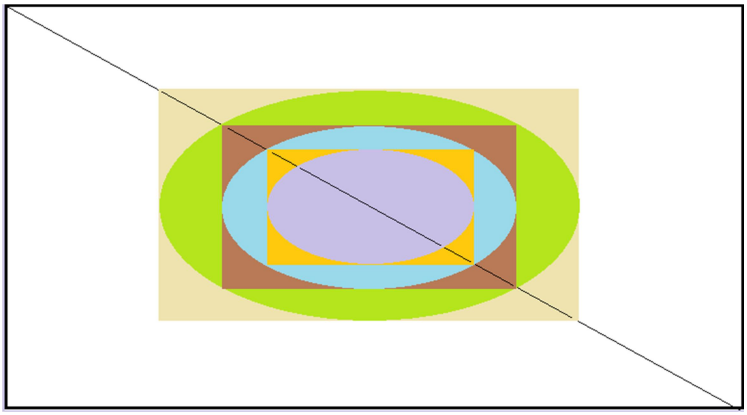
class MyOval:

Class MyOval inherits class MyShape and may use class MyRectangle. The MyOval object is defined by an ellipse inscribed in a rectangle of height h and width w, centered at a point (x, y). The MyOval object may be filled with a color. The class includes appropriate class constructors and methods, including methods that perform the following operations:

a. *getPerimeter* — returns the perimeter of the MyOval object;
b. *getArea* — sets the area of the MyOval object;
c. *toString* — returns a string representation of the MyOval object: axes lengths, perimeter, and area;
d. *draw* — draws a MyOval object inscribed in a rectangle of height h and width w, centered at a point (x, y). The center point of the oval is defined in class MyShape.

2- Use JavaFX graphics and the class hierarchy to draw a geometric configuration comprised of a sequence of alternating concentric ovals and rectangles as illustrated below, subject to the following additional requirements:

   a. The code is applicable to canvases of variable height and width;
   b. The dimensions of the shapes are proportional to the smallest dimension of the canvas;
   c. Only the drawing methods of the GraphicsContext object may be used for drawing the shapes in the class hierarchy;
   d. The rectangles and ovals are filled with different colors of your choice.
   e. The colors used to fill the shapes in the class hierarchy are specified by an enum data type **MyColor**. The enum type includes appropriate constructors and methods, including methods that define, mix, and return colors.



Hesham A. Auda
24 September 2019