Department of Computer Science
The City College of CUNY

CSc 22100 [P 32132]: Software Design Laboratory [Spring 2021]

Assignment 1

A report uploaded on the Bloackboard's course page for the section showing [1] *the problem,* [2] *solution methods,* [3] *codes developed, and* [4] *outputs produced for the assignment indicated is due by* 2:00 *pm on Tuesday,* 9 March 2021. ***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 MyShapeis the hierarchy's superclass and extends the Java class Object. An implementation of the class defines a reference 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* – return the x- and y-coordinates of the reference point of the MyShape object;
b. *area, perimeter* – return the area and perimeter of the object. These methods must be overridden in each subclass in the hierarchy. For the MyShape object, the methods return zero.
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 the color specified.

Class MyLine:

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

a. *length* — returns the length of the MyLine object;
b. *xAngle* — returns the angle (in degrees) of the MyLine object with the x-axis;

c. *toString* — returns a string representation of the MyLine object, including the line's endpoints, length, and angle with the x-axis;

d. *draw* — draws a MyLine object.

Class MyRectangle:

Class MyRectangle extends class MyShape. The MyRectangle object is a rectangle of height *h* and width *w*, and a top left corner point p(*x*, *y*), and may be filled with a color. The class includes appropriate class constructors and methods, including methods that perform the following operations:

e. *getX, getY, getWidth, getHeight* — return the width, height of the MyRectangle object

f. *toString*— returns a string representation of the MyRectangle object: top left corner point, width, height, perimeter, and area;

g. *draw*— draws a MyRectangle object of height *h* and width *w*, anchored at p(*x*, *y*).

Class MyOval:

Class MyOval extends class MyShape. The MyOval object is defined by an ellipse within a rectangle of height *h* and width *w*, and a top left corner point p(*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. *getX, getY, getA, getB* — return the x- and y-coordinates of the center point and abscissa of the MyOval object;

b. *toString*— returns a string representation of the MyOval object: axes lengths, perimeter, and area;

c. *draw*— draws a MyOval object.

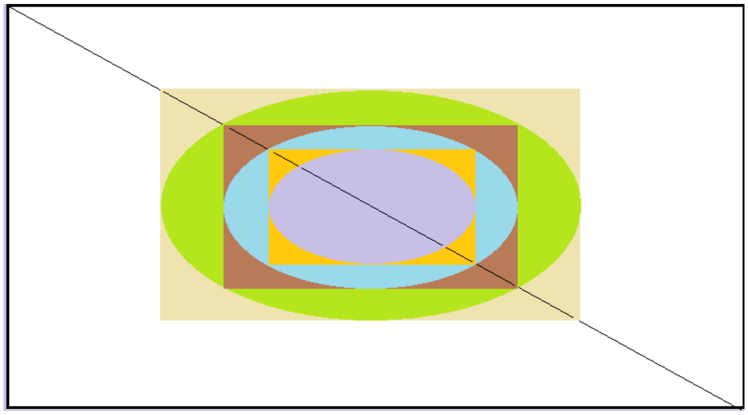
2- Use JavaFX graphics and the class hierarchy to draw a geometric configuration comprised of a sequence of alternating concentric ovals and their inscribed rectangles as illustrated below, subject to the following additional requirements:

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. The ovals and rectangles are filled with different colors of your choice, specified through a MyColor enum reference type.

3- Explicitly specify all the classes imported and used in your Java code.



Best wishes

Hesham A. Auda
02-23-2021