Department of Computer Science
The City College of CUNY

CSc 22100 (27461): Software Design Laboratory [Fall 2018]

Exercise 2

A printout showing the problem, solution method, codes developed, and outputs produced for the tests indicated is due during and before the end of the class on Wednesday, 24 October 2018. The deadline is strictly observed.

Consider the class hierarchy in Exercise 1.

1- Amend the hierarchy of Java classes in Exercise 1 as follows:

MyPolygon *is_a* MyShape;
MyRectangle *is_a* MyShape
MyOval *is_a* MyShape;
MyCircle *is_a* MyOval;

2- *Interface* MyShapeInterface, *interface* MyPositionInterface, and *interface* MyShapePositionInterface are specified in connection with the class hierarchy.

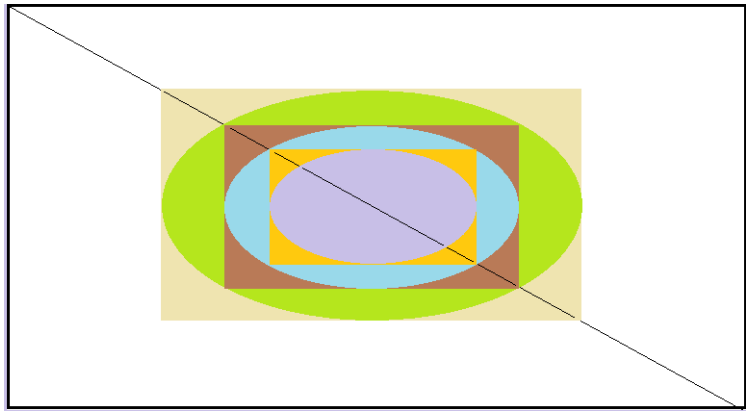
3- *Interface* MyShapeInterface includes appropriate abstract, static, and/or default methods that describe the intrinsic functions and behaviors of the specific object types of the class hierarchy, including:
   a. *getArea* — describes the area of an object in the class hierarchy;
   b. *getPerimeter* — describes the perimeter of an object in the class hierarchy.

4- *Interface* MyPositionInterface includes appropriate abstract, static, and/or default methods that describe the positional functions and behaviors of the specific object types of the class hierarchy, including:
   a. *getPoint* – returns the point (x, y);
   b. *moveTo* – moves point (x, y) to point $(x + \Delta x, y + \Delta y)$;
   c. *distanceTo* – returns distance from point (x, y) to a point;

5- The *abstract* class MyShape implements *interface* MyShapePositionInterface which extends *interface* MyShapeInterface and *interface* MyPositionInterface. *Interface* MyShapePositionInterface includes appropriate abstract, static, and/or default methods that describe the functions and behaviors of the specific object types of the class hierarchy, including:
   a. *getBoundingBox* — returns the bounding rectangle of an object in the class hierarchy;
   b. *doOverlap* — returns true if two objects in the class hierarchy overlap.

6- Use JavaFX graphics and the class hierarchy to build a class Application that processes *polymorphically* the subclasses in the hierarchy to draw the geometric object shown, 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 rectangles and ovals are filled with different colors of your choice.



Hesham A Auda
10 October 2018