Venezia, Gregory William Bismayer Edward Goldthorpe

Shapes

Some Methods within the shape interface:

- contains(Point2D p), or (double x, double y, double w, double h), or (double x, double y): Returns a Boolean value based on whether the specified point or shape is within the shape
- getBounds() or getBounds2D(): returns a rectangle that encompasses the shape that calls this method.
- intersects(double x, double y, double h, double w), or(Rectangle r): returns a Boolean based on whether the specified rectangle and the shape that calls this method have overlapping areas.

The following are some of the classes that implement the shape interface.

Polygon class: Polygon's can represent any shape that can be created with the points provided in their fields. Some other methods available:

- add(Point2D p): adds the specified point to the end of the polygon's coordinate arrays(x and y).
- translate(int deltaX, int deltaY): translates the polygon's position by the specified parameters.
- reset(): turns this polygon into a blank polygon with no values in its coordinate arrays.

Rectangle class: Rectangles contain the fields for their height, width, and the coordinates for the upper-left point. Some other methods available:

- add(Point2D p) or(Rectangle r): adds the given point or rectangle to the bounds of this rectangle, subsequently changing the overall shape.
- createUnion(Rectangle r): returns a Rectangle containing the Rectangle that calls this method and the Rectangle parameter.
- createIntersection(Rectangle r): returns the rectangle formed from the overlap of the Rectangle that calls this class and the Rectangle parameter.

Ellipse2D class: Ellipses contain the fields for the ellipse's width, height, and coordinates of the upper-left point of the ellipse's boundary rectangle. Ellipses inherit their methods from the same superclass that Rectangles do.

Line2D class: Lines contain fields for the coordinates of their endpoints. Some other methods available to them are:

- getP1(), getP2(), getX1(), getX2(), getY1(), getY2(): returns the respective values or coordinates.
- Setline(double x1, double y1, double x2, double y2): changes the line so that it matches the specified parameters.

Venezia, Gregory William Bismayer Edward Goldthorpe

Shapes

Here is how to draw shapes in java. You will need at least 3 classes

1. The object class

```
import java.awt.Graphics2D;
import java.awt.Rectangle;

public class Boat {
// your starting coordinates, usually
farthest left, farthest up
private int leftMostX;
private int UpMostY;

public Boat(int x, int y) {
leftMostX = x;
UpMostY = y;
}
// here is where you create the object
public void draw(Graphics2D g2d){
// you can construct as many shapes as
you want, just make sure to draw them all
Rectangle boat = new Rectangle
(leftMostX,UpMostY,80,20);
g2d.draw(boat);
}
}
```

2. The objects component class

```
import java.awt.Graphics;
import java.awt.Graphics2D;
import javax.swing.JComponent;

public class BoatComponent extends
JComponent {
//draws your shape
@Override
public void paintComponent(Graphics g1){
      //typecasts g1 to a 2D
      Graphics2D g2d = (Graphics2D) g1;
      //creates your object at the
specified coordinates
      Boat boat1 = new Boat(5,10);
      //draws your object into g2d
      boat1.draw(g2d);
}
}
```

3. The Viewer Class

```
import javax.swing.JFrame;
public class BoatViewer extends JComponent {

public static void main(String[] args){
// constructs a frame to display, with a size and a title
JFrame frame1 = new Frame();
frame1.setSize(500,500);
frame1.setTitle("boat");
// makes the frame close when you exit
frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
// adds the object to the frame
BoatComponent component1 = new BoatComponent();
frame1.add(component1);
// makes the frame visible
frame1.setVisible(true);
    }
    }
```