Shape Classes

By Spencer Alves, Edna Jones, Mark Morrison

Sources

Horstmann, Sections 3.9 and 2.13; Java Platform, Standard Edition 6 API specification, java.awt.geom.Shape

Summary

The Shape interface defines several methods that are common to all geometric shapes.

- contains(double x, double y) and contains(Point2D p) determine if the shape contains a point.
- contains(double x, double y, double w, double h) and contains(Rectangle2D r) test if the interior of the shape entirely encloses the given rectangle.
- getBounds() and getBounds2D() return the outer boundaries, in the form of a rectangle, of the shape.
- getPathIterator(AffineTransform at) and getPathIterator(AffineTransform at, double flatness) returns an object that allows you to follow the outline of the shape.
- intersects(double x, double y, double w, double h) and intersects(Rectangle2D r) test if the given rectangle intersects the shape.

The shape interface makes heavy use of a separate abstract class called Point2D. Point2D has an x and y location, but no direction or size. Therefore it does not implement Shape. The Point2D class has two concrete subclasses, Point2D.Double and Point2D.Float, which use the double and float types, respectively, to store the coordinates. The Point2D class also defines several useful methods.

There are many "shape classes" that implement Shape, here are some of the more useful ones:

- Rectangle2D is an abstract class with concrete subclasses Rectangle, Rectangle2D.Double, and Rectangle2D.Float. They each have an x and y position of the top-left corner and a dimension with width and height.
- Ellipse2D.Float and Ellipse2D.Double define an ellipse and take the same parameters as Rectangle2D. The position defines the top-left corner of the bounding rectangle.
- Line2D.Float and Line2D.Double define a line with two (x, y) locations. You can use Point2Ds to define the endpoints of the line.
- Arc2D.Float and Arc2D.Double define a part of an ellipse. The constructors take the location of the ellipse, the dimensions of the ellipse, the starting angle, the ending angle, and the shape of the closure as parameters.

Example

```
public static void main(String[] args) {
      Rectangle2D.Double bounds = new Rectangle2D.Double(12, 13, 52, 10);
      Ellipse2D.Double ellipse = new Ellipse2D.Double(16, 52, 10, 10);
      Point2D.Double myPoint = new Point2D(34, 10);
      if (ellipse.intersects(bounds)) {
            System.out.println("Intersection detected!");
      }
      if (ellipse.contains(myPoint)) {
            System.out.println("Point is inside ellipse.");
      }
}
```