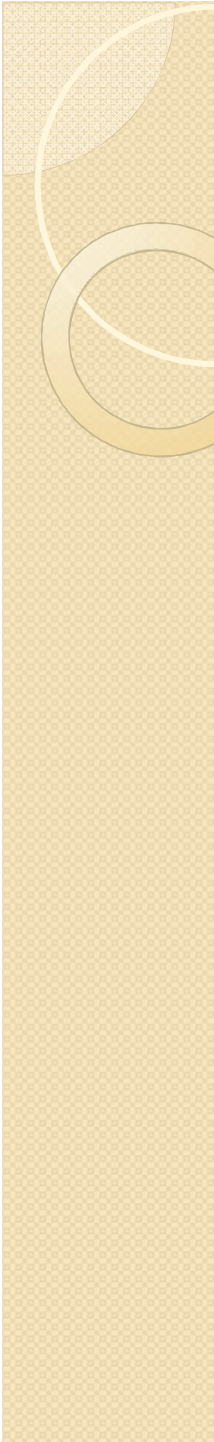




Course Name:
Advanced Java



Lecture 20

Topics to be covered

- AWT
- The Rendering Pipeline
- Shapes
- Areas
- Strokes
- Paint

Rendering Pipeline

- In the rendering pipeline, the following steps take place to render a shape:
-
- The path of the shape is stroked.
- The shape is transformed.
- The shape is clipped. If there is no intersection b/w the shape & the clipping area, then the process stops.
- The remainder of the shape after clipping is filled.
- The pixels of the filled shape are composed with the existing pixels.

Shapes

- Here are some of the methods in the Graphics class to draw shapes:
-
- drawLine
- drawRectangle
- drawRoundRect
- draw3DRect
- drawPolygon
- drawPolyline
- drawOval
- drawArc

Using the Shape Classes

- For the RoundRectangle2D shape, you specify the top left corner, width & height & the x & y dimensions of the corner area that should be rounded.
-
- RoundRectangle2D r = new RoundRectangle2D.Double(150,200,100,50,20,20);
-
- here w=150, h=200, x=100, y=50, r=20.
-
- The code above, produces a rounded rectangle with circles of radius 20 at each of the corners.
-

Areas

- The Java 2D API supports four *constructive area geometry* operations that combine two areas to a new area:
-
- **add**— The combined area contains all points that are in the first or the second area.
- **subtract**— The combined area contains all points that are in the first but not the second area.
- **intersect**— The combined area contains all points that are in the first and the second area.
- **exclusiveOr**— The combined area contains all points that are in either the first or the second area, but not in both.

Strokes

- You can construct strokes of arbitrary thickness. For example, here is how you draw lines that are 10 pixels wide.
- `g2.setStroke(new BasicStroke(10.0F));`
- `g2.draw(new Line2D.Double(. . .));`
- When a stroke is more than a pixel thick, then the end of the stroke can have different styles. These are called **end cap styles**. There are three choices

Paint

The java 2D API provides three such classes:

Color class – To fill shapes with a solid color, simply call setPaint with a color object, such as:

```
g2.setPaint(Color.red);
```

GradientPaint class – It varies colors by interpolating b/w two given colors values. u construct a GradientPaint object by specifying two points & the colors u want at these two points, such as:

```
g2.setPaint(new GradientPaint(p1,color.red, p2, color.pink));
```

TexturePaint class – It fills an area with repetitions of an image. To construct a TexturePaint object, you specify a BufferedImage & an anchor rectangle. The anchor rectangle is extended indefinitely in x & y directions to tile the entire coordinate plane. The image is scaled to fit into the anchor & then replicated into each tile