The classes that represent graphical objects form a hierarchy, part of which looks like this:

Graphic courtesy of Eric Roberts
Sending Messages to a GLabel

The following program illustrates sending a message to an object. Note that the label doesn’t appear until it is added to the canvas.

```java
public class HelloProgram extends GraphicsProgram {
    public void run() {
        GLabel label = new GLabel("hello, world", 100, 75);
        label.setFont("SansSerif-36");
        label.setColor(Color.RED);
        add(label);
    }
}
```

Graphic courtesy of Eric Roberts
Graphics Coordinates

- Origin is upper left
- Everything measured in pixels (dots on the screen)
- $x$ coordinates increase to the right
- $y$ coordinates increase going down
- `GLabel` coordinates are baseline of first character

$$(0, 0) \quad \rightarrow \quad \text{hello, world}$$

Graphic courtesy of Eric Roberts
Operations on the GObject Class

The following operations apply to all GObject:

- \texttt{object.setColor(color)}
  
  Sets the color of the object to the specified color constant.

- \texttt{object.setLocation(x, y)}
  
  Changes the location of the object to the point \((x, y)\).

- \texttt{object.move(dx, dy)}
  
  Moves the object on the screen by adding \(dx\) and \(dy\) to its current coordinates.

The standard color names are defined in the \texttt{java.awt} package:

- \texttt{Color.BLACK}
- \texttt{Color.DARK_GRAY}
- \texttt{Color.GRAY}
- \texttt{Color.LIGHT_GRAY}
- \texttt{Color.WHITE}
- \texttt{Color.RED}
- \texttt{Color.YELLOW}
- \texttt{Color.GREEN}
- \texttt{Color.CYAN}
- \texttt{Color.BLUE}
- \texttt{Color.MAGENTA}
- \texttt{Color.ORANGE}
- \texttt{Color.PINK}

Based on slides by Eric Roberts
Operations on the **GLabel** Class

**Constructor**

```
new GLabel(text, x, y)
```

Creates a label containing the specified text that begins at the point \((x, y)\).

**Methods specific to the **GLabel** class**

```
label.setFont(font)
```

Sets the font used to display the label as specified by the font string.

The font is typically specified as a string in the form

```
"family-style-size"
```

- **family** is the name of a font family
- **style** is either `PLAIN`, `BOLD`, `ITALIC`, or `BOLDITALIC`
- **size** is an integer indicating the point size

*Based on slides by Eric Roberts*
# Drawing Geometrical Objects

## Constructors

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>new GRect(x, y, width, height)</code></td>
<td>Creates a rectangle whose upper left corner is at ((x, y)) of the specified size.</td>
</tr>
<tr>
<td><code>new GOval(x, y, width, height)</code></td>
<td>Creates an oval that fits inside the rectangle with the same dimensions.</td>
</tr>
<tr>
<td><code>new GLine(x_0, y_0, x_1, y_1)</code></td>
<td>Creates a line extending from ((x_0, y_0)) to ((x_1, y_1)).</td>
</tr>
</tbody>
</table>

## Methods shared by the GRect and GOval classes

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>object.setFilled(fill)</code></td>
<td>If (fill) is \texttt{true}, fills in the interior of the object; if \texttt{false}, shows only the outline.</td>
</tr>
<tr>
<td><code>object.setFillColor(color)</code></td>
<td>Sets the color used to fill the interior, which can be different from the border.</td>
</tr>
</tbody>
</table>

*Based on slides by Eric Roberts*
Size of Graphics Window

Methods provided by \texttt{GraphicsProgram} class

\begin{tabular}{|l|l|}
\hline
\texttt{getWidth()} & Returns the width of the graphics window. \\
\hline
\texttt{getHeight()} & Returns the height of the graphics window. \\
\hline
\end{tabular}

Note: receiver of these calls is the \texttt{GraphicsProgram} itself, so we don’t specify a separate object as receiver.