

## GUI design using Swing

Sources:

<http://stackoverflow.com/questions/408820/what-is-the-difference-between-swing-and-awt>

Big Java 4<sup>th</sup> Edition

<http://docs.oracle.com/javase/7/docs/technotes/guides/swing/>

Swing is the nickname for a graphical user interface (GUI) library in Java; unlike AWT it creates its graphics purely through Java, not through the operating system. Both Swing and AWT are GUI libraries.

There are several components to creating a simple Swing program:

- a class that paints the component onto the content pane (import java.awt.Graphics and javax.swing.JPanel, extend JPanel)
- a class that acts as a "frame" for the holding the content pane (import javax.swing.JFrame)
- any additional classes that perform whatever other features you'd like to include in your program (for example, a class that implements ActionListener defines what should be done when a user performs certain operations, such as clicking a button or pressing Enter in a text field)

How to implement the GUI Components:

- First off, be aware that Java's coordinate system starts at the upper left hand corner at (0, 0). The x-coordinate is the horizontal location moving from left to right, and the y-coordinate is the vertical location moving from top to bottom. Coordinate units are measured in pixels.
- Override the paintComponent(Graphics g) method to whatever graphics you need to paint.

How to implement the frame class:

- Instantiate an object of the JFrame class and objects of your components.
- Set the size of the frame
- Set the title of the frame (optional)
- Set the default close operation
- Make the frame visible
- Add your components to the frame
- If you have extra classes, instantiate them here (for example, an ActionListener object).
  - If you have a button, instantiate it too.

How to create a class that implements ActionListener (more specifically, creating a JButton):

- Create a method whose parameter is ActionEvent. Within this method define what you want your button to do

Using Layout Managers:

- Layout Managers do two things (based on container's properties and on the children's min/preferred/max sizes):
  - Calculate the min/preferred/max sizes for a container
  - Lay out the container's children
- Several Layout Managers include BorderLayout, BoxLayout, GridLayout, and FlowLayout.
- Every content pane is initialized to use the BorderLayout.
- Every JPanel is initialized to use FlowLayout

Code example:

```
import java.awt.Graphics;
import javax.swing.JPanel;

public class Circle extends JPanel
{
    public void paintComponent(Graphics g)
    {
        int width = getWidth();
        int height = getHeight();

        super.paintComponent(g);

        g.drawOval(0, 0, width, height);
    }
}

import javax.swing.JFrame;

public class CircleTest
{
    public static void main(String[] args)
    {
        Circle circle1 = new Circle();
        JFrame application = new JFrame();

        application.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        application.add(circle1);
        application.setSize(250, 250);
        application.setVisible(true);
    }
}
```

Layout Manager examples:

