CS102

Introduction to data structures, algorithms, and object-oriented programming

May 2, 2016
Adding Focus

• In order to use KeyListeners, you normally need to add FocusListeners. The container needs to have input focus before the keys will register.

• A component like a JTextField can request the focus at different places in the program:

• JTextField letterInput;

        letterInput.requestFocus();
Rule of Drawing

Graphics objects are often passed into methods as a parameter and the drawing is done using the Graphics object argument in that method.

The original Graphics object is usually passed into the paintComponent method by the system whenever repaint is called.
Sensing Impact between Objects drawn on scene

• If we want the ball to bounce off the paddle, we need to detect:
  1. When lower surface of ball is touching top of paddle;
  2. when ball is moving downward (toward paddle).
• When both these conditions hold, we want to negate the y direction of the ball.
• This manner of collision detection is useful in any application that has objects moving on crossed paths or moving objects with stationary objects in their path.
Layout Managers in Swing

• Control geometric placement in containers.

• Layout manager defines methods for positioning and sizing objects in a container.

• We will look at three different layout managers that are widely used.
Layout Managers in Swing

• In practice, most of the world (except your prof) considers it easier to use one of the existing layout managers to set up a GUI, e.g.:

FlowLayout: Objects flow from left to right as they are added. If frame is resized, flow changes, trying to maintain a set size for components.
import javax.swing.*;
import java.awt.*;

public class ShowFlowLayout{
    public static void main(String[] args){
        JFrame win = new JFrame("FlowLayout Demo");
        win.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        win.getContentPane().setLayout(new FlowLayout());

        for (int i = 0; i < 10; i++) {
            win.getContentPane().add((new JButton(String.valueOf(i))));
        }

        win.pack();
        win.setVisible(true);
    }
}
GridLayout Manager

**GridLayout(int rows, int cols)** Creates a grid layout with the specified number of rows and columns. All components in the layout are given equal size. One, but not both, of rows and cols can be zero, which means that any number of objects can be placed in a row or in a column.

**GridLayout(int rows, int cols, int hgap, int vgap)** Like above except the horizontal and vertical gaps are set to the specified values. Horizontal gaps are places between each of columns. Vertical gaps are placed between each of the rows.
GridLayout Manager

(2,0)

(5,2)

(3,2)
import javax.swing.*;
import java.awt.*;

public class ShowGridLayout{
    public static void main(String[] args){
        JFrame win = new JFrame("GridLayout Demo");
        win.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        win.getContentPane().setLayout(new GridLayout(2, 0));
        win.setSize(200,100);

        for (int i = 0; i < 10; i++) {
            win.getContentPane().add(new JButton(String.valueOf(i)));
        }

        win.pack();
        win.setVisible(true);
    }
}


BorderLayout Manager

BorderLayout: Gives 5 areas to hold components:

North, South, East, West, and Center

When a component is added, you specify one of the 5 areas to put it in. If no object is specified for an area, the area is left blank.
import javax.swing.*;
import java.awt.*;

public class ShowBorderLayout{
    public static void main(String[] args){
        JFrame win = new JFrame("BorderLayout Demo");
        win.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        Container content = win.getContentPane();
        content.setLayout(new BorderLayout());
        content.add(BorderLayout.NORTH, new JButton("North"));
        content.add("South", new JButton("South"));
        content.add("East", new JButton("East"));
        content.add("West", new JButton("West"));
        content.add("Center", new JButton("Center"));

        win.setSize(400,200);
        win.setVisible(true);
    }
}
Layout Managers in Swing

You can use NetBeans to create your GUI layout. (see lecture demo).

[Image: A GUI with a text field labeled "Enter your name" and a count of the number of letters in the entered name.]