CS 121 - Intro to Programming - Lecture 23

Announcements

OWL #10 due tonight (11 PM)
Program 6 due tomorrow, OWL hand-in
Tomorrow's discussions: just 2, program clinics
Ch 12 Embedded Problems due next Tuesday

Final Exam: Friday, Dec 18, 1:30 Mahar
Graphics & Java’s event model

Rely heavily on the concept of an interface

We make up our own

• Scoring

Library interfaces

• Comparable

  built in to String class

we can incorporate in our own classes
public class Infant implements Comparable{
    private String name;
    private int age; // in months
    public Infant(String who, int months){
        name = who;
        age = months;
    }
    ...
    public int compareTo(Object other){ // by age
        int b = ((Infant)other).getAge(); // cast req!
        int a = this.age;
        return(a-b);
    }
}
//now name, alphabetical, is natural ordering:

public int compareTo(Object other){
    String a = this.name;
    String b = ((Infant)other).getName();
    return(a.compareTo(b));
}

In other words: since a, b are Strings, we “hand off” the ordering decision to String class
Graphics - Begin at the end: your next (last?) programming project.

Graphical Opoly

Elements

- program execution advances by clicking
- also by menu selection
- picture updated after every click
- labels updated after every click
A **JFrame** is like a window frame..

It comes with a content pane, a container that holds things (surfaces..)

We’ll place on the surface something we can draw on / write on / add hardware to: a **JPanel**

Then we do just that: we write on it, draw on it, color it.
JFrame (frame)

ContentPane

JPanel (panel - is “affixed” to contentPane)
import java.awt.*;
import javax.swing.*;

public class FirstGraphics{

    public static void main(String[] args){
        JFrame frame = new JFrame("Getting Started");
        Container c = frame.getContentPane();
        BabyGeoPanel p = new BabyGeoPanel(Color.green);
        c.add(p); // add panel to frame's container
        frame.pack(); // prepares frame for display
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

import java.awt.*; import javax.swing.*;

public  class BabyGeoPanel extends JPanel{

    public BabyGeoPanel(Color g){
        setPreferredSize(new Dimension(700,300));
        setBackground(g);
    }

    public void paintComponent(Graphics g){
        super.paintComponent(g);
        g.setColor(Color.red);
        g.fillRect(10,20,100,100);
        g.setColor(Color.blue);
        g.drawOval(200,20,300,100);
        g.setColor(Color.black);
        g.drawRect(200,20,300,100);
        g.drawLine(0,0,200,20);
    }
}
increasing $x$

- $(10, 30)$

increasing $y$

- $(22, 105)$
Two big themes, revisited

Factor out the general framework, use inheritance (good news)

“Write once, run everywhere” principle in Java starts to break down when graphics gets complicated (bad news)
Inheritance and the frame/panel machinery

We’ll split off the frame code and make a general purpose “display” window.

We’ll develop a simple, general mechanism for adding panels to the code.

This will shift the work of GUI development to a panel or panels.
import java.awt.*; import javax.swing.*;

public class DisplayWindow extends JFrame{
    private Container c;
    public DisplayWindow(){
        super("Display");
        c = this.getContentPane();
    }

    public void addPanel(JPanel p){
        p.setPreferredSize(new Dimension(500,400));
        c.add(p); }

    public void showFrame(){
        this.pack();
        this.setVisible(true);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); }
}
}
Java’s event model

What Scanner-based interactions are like..
(a script)

What modern computing is like..

Your program needs to know:

1) What event to listen for
2) Clarify who’s listening
3) Provide a mechanism for event reaction
The basic script for event handling

• Create components for generating events
• Identify listener for events
• Connect listener, event generator
• Enable listener to listen and act
• Describe actions when events are triggered
• Components: buttons, menus, sliders, ...
import java.awt.);

public class BabyButtonDriver{

    public static void main(String[] args){
        DisplayWindow d = new DisplayWindow();
        BabyControlPanel p = new BabyControlPanel();
        d.addPanel(p);
        d.showFrame();
    }
}
}
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class BabyControlPanel0 extends JPanel implements ActionListener{
    JButton quit = new JButton("Quit"); // make button

    public BabyControlPanel0(){
        setBackground(Color.red);
        this.add(quit); // add quit button to panel
        quit.addActionListener(this); // identify panel as listener
    }

    public void actionPerformed(ActionEvent e){ //i-face method
        if (e.getSource() == quit) System.exit(0);
    }
}
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class BabyControlPanel extends JPanel implements ActionListener{

    JButton quit = new JButton("Quit");
    JButton color = new JButton("Toggle Color");
    boolean toggle = false;

    public BabyControlPanel(){
        setBackground(Color.red);
        this.add(quit);
        quit.addActionListener(this);
        this.add(color);
        color.addActionListener(this);
    }
}
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == quit) System.exit(0);
    else if (e.getSource() == color) {
        if (toggle) setBackground(Color.red);
        else setBackground(Color.blue);
        toggle = !toggle;
    }
}
}
public class PhrasePanel extends JPanel implements ActionListener {
    JButton quit = new JButton("Quit");
    JButton place = new JButton("Place");
    JTextField xVal = new JTextField(5);
    JTextField yVal = new JTextField(5);
    JTextField phrase = new JTextField(20);
    int x, y;
public PhrasePanel() {
    setPreferredSize(new Dimension(400,600));
    setBackground(Color.green);
    this.add(quit); // place button in panel
    quit.addActionListener(this);
    this.add(place); // place button in panel
    place.addActionListener(this);
    this.add(xVal);this.add(yVal);
    this.add(phrase);  }

public void paintComponent(Graphics g) {
    super.paintComponent(g);
    g.drawString(phrase.getText(),x,y);
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == quit)
        System.exit(0);
    else if(e.getSource() == place) {
        x = Integer.parseInt(xVal.getText());
        y = Integer.parseInt(yVal.getText());
        repaint();
    }
}