COSC 123 Computer Creativity

Graphics and Events

Dr. Ramon Lawrence University of British Columbia Okanagan ramon.lawrence@ubc.ca

Key Points

1) Draw shapes, text in various fonts, and colors.

2) Build window applications using ${\tt JFrame}/{\tt JPanel}$ and Swing components.

3) Understand events, event listeners, and event adapters.

4) Write code for handling mouse, keyboard, and window events.

Page 2

COSC 123 - Dr. Ramon Law

Java Programs Overview

To this point, all our Java programs have received input and displayed output in the console (text window).

Types of Java programs:

- ◆1) Console applications text-based applications which perform input and output using the console
- ◆2) Graphical applications stand-alone Java applications which have a graphic user interface with components such as windows, control buttons, menus, and check boxes.

Page 3

OSC 123 - Dr



COSC123-Dr. Ramon Lawrence Creating a Frame Windows import javax.swing.JFrame •create our own class (like MyFrame) which extends JFrame •provide a constructor for our MyFrame class •set the size of our frame using the setSize method ⇒ usually performed in MyFrame constructor To use the MyFrame window: •define a mainline which instantiates a MyFrame instance •use the setTitle method to set the frame title (optional) •use the setVisible method to display the frame on the screen

Page 5

COSC123-Dr. Ramon Lawrence Graphical Applications Creating a Frame Window import javax.swing.JFrame; public class MyFrame extends JFrame { public static void main(String[] args) { MyFrame frame = new MyFrame(); frame.setDitle("Frame Title"); frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); frame.setVisible(true); } public MyFrame() { final int DEFAULT_FRAME_WIDTH = 300; setSize(DEFAULT_FRAME_WIDTH, DEFAULT_FRAME_HEIGHT); } Page 6























COSC 123 - Dr. Remon Lawrence Java Swing Components	
The Java Swing package contains the user interface components that we will use in our graphical applications.	
Component	Import Package
JButton	javax.swing.JButton
ButtonGroup	javax.swing.ButtonGroup
Check box	javax.swing.JCheckBox
Combo box	javax.swing.JComboBox
JFrame	javax.swing.JFrame
JLabel	javax.swing.JLabel
JPanel	javax.swing.JPanel
Radio button	javax.swing.JRadioButton
Text field	javax.swing.JTextField
	Page 18









GUI Components JTextField and JTextArea JTextField allows us to read in a single line of text. JTextArea allows us to handle multiple lines of text. With a JTextField, you may give the # of characters: JTextField txtField = new JTextField(5); // 5 chars. With a JTextArea, you can give the # of rows/cols: JTextArea txtArea = new JTextArea(5,40);//5 rows, 40 cols Page 23

COSC 123 - Dr. Ramon Lawren



COSC 123 - Dr. Ramon Lawrence



ButtonGroup sizeGroup = new ButtonGroup();

which buttons are grouped with each other.

if (smallButton.isSelected()) return "Small";

sizeGroup.add(smallButton); sizeGroup.add(mediumButton); sizeGroup.add(largeButton);

smallButton.setSelected(true);

JRadioButton largeButton = new JRadioButton ("Large");

The ButtonGroup class allows the programmer to specify

You can select buttons or determine if buttons are selected by:

Page 25

COSC 123 - Dr.

COSC 123- Dr. Ramon Lawrence GUI Components JCheckBox Overview The JCheckBox class allows the user to select yes/no valued inputs (i.e. true or false). JCheckBox boldCheckBox = new JCheckBox("Bold"); •Note: Do not place check boxes inside a button group because they are not mutually exclusive.

Page 26

GUI Components JComboBox Overview

of disjoint inputs where radio buttons are too awkward. ◆A JComboBox allows you to select an item from the list.

 If the list is editable, you can type in your own selection that may not already be in the list.

JComboBox itemCombo = new JComboBox(); itemCombo.addItem("Item 1"); itemCombo.addItem("Item 2");

You can get the selected item in the list by:

generate action events that should be detected using an action listener.







Components

Question: What is the best component to use if the user must pick only one item from 50 possible choices?

COSC 123 - Dr. Ramon Lav

- A) JRadioButton
- B) JComboBox
- **C)** JCheckBox
- **D)** JButton

COSC 123 - Dr. Ramon Lawrence **Events and Event Handling Diamon Structure Programming Philosophy** In graphical applications, the programmer must react instead of dictate the events that occur in a program. As a programmer, you design a graphical user interface with windows, buttons, and components that the user can interact with. You do not know the order or the sequence of events the user will generate, but you must be able to react to them. Page 32

Events and Event Handling Overview

An $\ensuremath{\textit{event}}$ is a notification to your program that something has occurred.

◆For graphical events (mouse click, data entry), the Java window manager notifies your program that an event occurred.
⇔ There are different *kinds* of events such as keyboard events, mouse

click events, mouse movement events, etc. An **event handler** or **listener** is part of your program that is

responsible for "listening" for event notifications and handling them properly.

An event listener often only listens for certain types of events. An event source is the user interface component that generated the event.

A button, a window, and scrollbars are all event sources.

Page 33

COSC 123 - Dr. Ramon Law

Page 31



Mouse Event Example Handling mouse click events requires three classes: 1) The event class - that stores information about the event. ⇒ For mouse clicks, this class is MouseEvent. ⇒ The MouseEvent class has methods getX() and getY() that indicate the position of the mouse at the time the event was generated. ⇒ Each event class has the method Object getSource() that returns the source of the event. 2) The listener class - allows your program to detect events. Building your own listener class requires implementing a predefined interface. ⇒ For mouse clicks, the interface is MouseListener. MouseAdapter is a class that implements the MouseListener interface. 3) The event source - is the component in your GUI that generated the event.

Page 35



COSC 123 - Dr. Ramon Lawrence Mouse Event Example Code provide state vaid main(tring() gyp (provide state stat













Page 43











Multiple Button Action Listener

public class MyFrame
{
 public MyFrame()
 {
 ...
 upButton = new JButton("Up");
 downButton = new JButton("Down");
 leftButton = new JButton("Left");
 rightButton = new JButton("Right");
 ActionListener listener = new DirectionListener();
 upButton.addActionListener(listener);
 downButton.addActionListener(listener);
 rightButton.addActionListener(listener);
 rightButton.addActionListener(listener);
 rightButton.addActionListener(listener);
 rightButton.addActionListener(listener);
 rightButton.addActionListener(listener);
 rightButton.addActionListener(listener);
 rightButton.addActionListener(listener);
 rightButton.addActionListener(listener);
 rightButton addActionListener(listener);
 rightButton addActionListener(listener);
 rightButton addActionListener(listener);
 rightButton addActionListener(listener);
 Pictener(listener);
 Pictener(listener);
 rightButton addActionListener(listener);
 rightButton addActionListener(listener);
 Pictener(listener);
 rightButton addActionListener(listener);
 ActionListener(listener);
 Pictener(listener);
 Pictener(listener);
 Pictener(listener);
 ActionListener(listener);
 Pictener(listener);
 ActionListener(listener);
 Pictener(listener);
 ActionListener(listener);
 Pictener(listener);
 ActionListener(listener(listener);
 ActionListener(listener);
 ActionListener(listener);
 ActionListener(listener);
 ActionListener(listener);
 ActionListener(listener);
 Pictener(listener);
 ActionListener(listener);
 ActionListener(listener);
 ActionListener(listener);
 ActionListener(listener);
 ActionListener(listener);
 ActionListener(listener);
 ActionListener(listener);

COSC 123 - Dr. Ramon Lay







Menus Overview Menus allow the user to select options without using buttons and fields. A menu is located at the top of the frame in a menu bar. A menu is a collection of menu items and more menus. You add menu items and submenus with the add method. When a menu item is selected, it generates an action event. Thus, each menu item should have a listener defined.

Page 53

COSC 123 - Dr. Ramon Law













Exercise

Create an application that has a File menu and an edit menu.

- The file menu should have an exit item that closes the application.
- ◆The edit menu should have two subitems:
 ⇔ shape has submenu of rectangle, square, and circle
 ⇔ color has submenu of red, green, blue, yellow
- ♦When the use selects a shape and color, remember the shape and color. Default is rectangle and red.
- When the user clicks on a place on the screen, draw that shape in that color.

Page 60

COSC 123 - Dr. Ramon Lawren

Timer

A *timer* can be used to create events at set times. A timer generates ActionEvents.

Creating a timer:

Timer timer = new Timer(1000, listener);

// The timer fires every 1000 ms (1 second).
// The listener class is called every time.

Starting and stopping a timer:

timer.start();
timer.stop();

Page 61

COSC 123 - Dr. Ramon Law





Page 63











COSC 123 - Dr. Ramon La COSC 123 - Dr. Ramon La Graphical User Interfaces Conclusion Objectives Buttons, text fields, check boxes, combo boxes, and menus are Definitions: event, event handler/listener, event source all components in the Java Swing package that can be used to Java skills: developed a GUI for your application. Create applets and place on web pages. ◆Use the Java coordinate system. Components generate events (usually action events) to indicate Draw basic shapes, change colors and fonts. when they have been clicked on or accessed by the user. ♦ Window applications using JFrame and JPanel. •We handle the events using listeners and adapters. ◆Java Swing components: JButton, JCheckBox, JComboBox, JLabel, JPanel, JRadioButton, JTextField, JTextArea The important thing about Swing is not memorizing the components and their methods, but understanding how the Event listeners versus event adapters ◆Mouse events: MouseListener. MouseAdapter components work and generate events. ♦ Window events: WindowListener, WindowAdapter +Focus on event handling and the concept of using components, not on the definition of the components! ◆ActionListener and use with JButton Page 69 Page 70

Objectives (2)

Java skills (cont.):

- ♦Using inner classes.
- ♦Menus: JMenu, JMenuItem, JMenuBar
- ◆Timer and timer events
- ♦Keyboard events: KeyListener

Page 71

COSC 123 - Dr. Ramon Lawr