

# Window Interfaces Using Swing

Chapter 13

### Objectives

- Describe the basics of event-driven programming
- Design and code a simple GUI that includes buttons and text
- Use the standard classes

#### **Graphical User Interfaces**

- Referred to as GUI (pronounced "gooey")
- Elements of a GUI
  - Window portion of the screen
  - Menu list of choices
  - Button click to choose an item

## **Event Driven Programming**

- An event is an object that represents an action
  - An object "fires" an event
- Figure 13.1
   Event Firing
   And An
   Event
   Listener



- Note <u>example program</u>, listing 13.1 class FirstSwingDemo
- Window appears on screen with message
- Uses JFrame object



- Note use of
  - JLabel object
  - Content pane, the inside of the window
  - Registering a listener
  - Setting the window to be visible

 Figure 13.2 Clicking the Close-Window button



• View <u>listener class</u>, listing 13.2 class WindowDestroyer

#### More About Window Listeners

- Derived from class WindowAdapter
- Figure 13.3 Methods in the Class
   WindowAdapter

public void windowOpened(WindowEvent e) Invoked when a window has been opened.

public void windowClosing(WindowEvent e)

Invoked when a window is in the process of being closed. Clicking the close-window button causes an invocation of this method.

public void windowClosed(WindowEvent e)
 Invoked when a window has been closed.

public void windowIconified(WindowEvent e)

Invoked when a window is iconified. When you click the minimize button in a JFrame object, the window is iconified. See Listing 13.1 for the location of the minimize (iconifying) button.

### More About Window Listeners

#### • Figure 13.3 ctd.

public void windowDeiconified(WindowEvent e) Invoked when a window is deiconified. When you activate a minimized window, it is deiconified.

public void windowActivated(WindowEvent e)
 Invoked when a window is activated. When you click in a window, it becomes the activated window. Other actions can also activate a window.

public void windowDeactivated(WindowEvent e) Invoked when a window is deactivated. When any window is activated, all other windows are deactivated. Other actions can also deactivate a window.

public void windowGainedFocus(WindowEvent e)
 Invoked when a window gains focus. (Focus is not discussed in this text.)

public void windowLostFocus(WindowEvent e)
 Invoked when a window loses focus. (Focus is not discussed in this text.)

public void windowStateChanged(WindowEvent e)
 Invoked when a window changes state.

## Size Units for Screen Objects

- Smallest screen area displayed is a pixel
- With Swing,
  - Both size and position of objects on screen measured in pixels.
- A screen's resolution is a measure of the number of pixels it can display

#### The setVisible Method

- Takes one argument of type boolean
- Other objects besides windows can be made visible or invisible
  - The object calls the method

Syntax:

Object\_For\_Screen.setVisible(Boolean\_Expression);

- A <u>better version</u> of first Swing program, listing 13.3 class FirstWindow
- A program that uses class FirstWindow listing 13.4 class FirstWindowDemo



#### Adding Items to a **JFrame** Window

Requires the following syntax:

Syntax (within a constructor):

getContentPane().add(JLabel\_Object);

Example (within a constructor):

JLabel myLabel = new JLabel("Please don't click that button!"); getContentPane().add(myLabel);

- View <u>new sample</u> window, listing 13.5 class SecondWindow
- Note new elements
  - A title, Second Window
  - A local variable named contentPane to reference the content pane of the window
  - A background color, blue
  - A new way to add the window listener

#### A Window with Color

- Note color constants
- Figure 13.4, the color constants

Color.BLACK	Color.MAGENTA
Color.BLUE	Color.ORANGE
Color.CYAN	Color.PINK
Color.DARK_GRAY	Color.RED
Color.GRAY	Color.WHITE
Color.GREEN	Color.YELLOW
Color.LIGHT_GRAY	

 View <u>demo program</u>, listing 13.6 class SecondWindowDemo

#### A Window with Color

Run of demo program



#### Methods in Class JFrame

• Figure 13.5

public JFrame()

Creates a new JFrame window.

public JFrame(String title)
Creates a new JFrame window with the given title.

```
public void add()
```

This method is inherited from an ancestor class and is basically useless. Adding something to a JFrame's content pane—using getContentPane().add(*Item\_Added*) —involves a different add method.

public void addWindowListener(WindowListener ear) Registers ear as a listener for events fired by the JFrame window.

public Container getContentPane()

Returns the content pane of the JFrame window. Note that the content pane returned is of type Container.

#### Methods in Class JFrame

• Figure 13.5 ctd.

public void setBackground(Color c)
 Sets the background color to c.

public void setForeground(Color c)
 Sets the foreground color to c.

public void setSize(int width, int height)
 Resizes the window to the specified width and height.

public void setTitle(String title)
Displays the given title on the title bar of the window.

public void setVisible(boolean isVisible) Makes the window visible if the argument is true, or invisible if the argument is false.

#### Layout Managers

- A layout manager arranges objects
- View <u>example</u> using BorderLayout listing 13.7 class BorderLayoutDemo

Layout Demonstration				
First label here.				
		Sa	ample	
Third label anywhere.		S	screen	
		0	utput	
Second label there.				

#### Layout Managers

• Figure 13.6 Five regions of a **BorderLayout** manager

	BorderLayout.NORTH	
BorderLayout.WEST	BorderLayout.CENTER	BorderLayout.EAST
	BorderLayout.SOUTH	

Syntax

Syntax:

Container\_Object.setLayout(new Layout\_Manager\_Class());

Example (within a constructor):

```
Container contentPane = getContentPane();
contentPane.setLayout(new FlowLayout());
```

```
JLabel label1 = new JLabel("Labels are good.");
contentPane.add(label1);
```

### Layout Managers

• Figure 13.7, some layout managers

Layout Manager	Description
BorderLayout	Arranges the components in five areas: north, south, east, west, and center. You specify an area as a second argument of the add method.
FlowLayout	Arranges components from left to right in the same fashion that you normally write things on a piece of paper.
GridLayout	Arranges components in a grid of rows and columns, with each component stretched to fill its box in the grid.

#### **Button and Action Listeners**

Note <u>sample program</u>, listing 13.8
 class ButtonDemo



#### **Buttons**

- Adding buttons to a GUI
  - Create the instance of the button
  - Use the .add method
- Remember that the Close-Window button is not an object of the class JButton

### Action Listeners, Action Events

- Swing, GUI class definition should:
  - Register a listener for each button
  - Define a method to be invoked when the event is sent to the listener
- A button "fires" action events
- Action events are handled by action listeners

#### Action Listeners, Action Events

- To make a class into an action-listener class you must:
  - Add the phrase implements
     ActionListener to the heading of the class definition.
  - Define a method named actionPerformed.



#### Action Listeners, Action Events

Figure 13.8 buttons and an action listener



#### Model–View–Controller Pattern

 Figure 13.9 the model—view—controller pattern



#### **Container Classes**

- Building a Swing GUI uses inheritance and containment
- Inheritance
  - Use the Swing class JFrame and make your window a derived class of JFrame
- Containment
  - Use one of the Swing classes as a container and to place elements in the container.

#### Class JPanel

- A panel is a container
- Panels can have a color
- To add a button to a panel
  - Use add to place the button in the panel buttonPanel.add(stopButton);
  - Use add to place the panel in the frame's content pane contentPane.add(buttonPanel, BorderLayout.SOUTH);

### Putting Buttons into a Panel

View <u>sample program</u>, listing 13.9
 class PanelDemo



#### Class Container

- A container class is a descendant of the class Container
- A component class is a descendant of the class JComponent



#### Class Container

 Figure 13.10 hierarchy of swing classes, ctd.



#### **Three Kinds of Objects**

- When using a Swing container class, you have three kinds of objects to deal with
  - The container class itself
  - The components you add to the container
  - A layout manager, which positions components inside container

## Text Areas, Text Fields

- A text area contains user input or program output
- View <u>example program</u>, listing 13.10
   class MemoSaver

Call home Sample screen
screen

- Labeling a text field
- Note <u>sample program</u>, listing 13.11 class LabelDemo

	🕾 Name Tester	
A label	A very good name!	Sample
	Test Clear	output

## Numbers as Input, Output

- With Swing
  - All input from, output to the screen is of type string
  - If input is meant to be numeric, programmer must convert from string
  - Similarly numeric values for output must be converted to string

- A GUI adding machine
- View <u>program</u>, listing 13.12 class Adder



## Catching

#### NumberFormatException

- Problem with previous program
  - What happens if user enters non-numeric value in the text field?
- Program should catch that situation and throw an exception
- View <u>improved program</u>, listing 13.13 class ImprovedAdder

#### Catching NumberFormatException

Program gives warning for invalid input

🕾 Adding Machine		
Error. Reenter Number.		
Add Reset	Sa sc ou	mple reen itput

## Summary

- GUIs require event driven programming
- Use Swing with inheritance or add components to a container class
- Define windowing GUI as a derived class of the Swing class **JFrame**.
- Add string of text to GUI as a JLabel object

## Summary

- Click an object of type JButton to fire an event
- Event handled by actionlistener
- Use .add method to place elements in a container object
- For an object of the class JFrame use the method getContentPane

## Summary

- A panel is a container object that is used to group elements inside of a larger container.
- TextField and TextArea objects are used for text input, output