

Swing and GUI Programming

Lecture 16
CGS 3416 Spring 2016

April 12, 2016

Graphics classes

- ▶ In the original version of Java, graphics components were in the AWT library (Abstract Windows Toolkit)
 - ▶ Was okay for developing simple GUI applications
 - ▶ For different platforms, AWT components mapped to platform-specific components.
 - ▶ Prone to platform specific bugs.
 - ▶ Primary package: `java.awt`. See other awt package APIs on docs.oracle.com/javase/8/docs/api/
- ▶ When Java 2 was released, a library known as the **Swing components** were introduced with the idea of replacing the older AWT user-interface components (like `Button`, `TextField`, `TextArea`, etc).
 - ▶ less dependent on target platform
 - ▶ a more robust and flexible library
 - ▶ Primary package: `javax.swing`
- ▶ Since release of Java 2, the Swing components are recommended for building graphic user interfaces for later browsers

Java Graphics API (Using Swing Components)

- ▶ **Component:** A base class for all non-menu graphic user interface classes.
- ▶ **Container:** a base class for container classes. A container is used to group smaller components. The most important containers are:
 - ▶ **JApplet** - for holding Applets
 - ▶ **JFrame** - for holding GUI components in applications. A window that is on the outer level (not inside another window)
 - ▶ **JPanel** - invisible container holding user-interface components. Can be nested, and can be used as canvases for drawing graphics.
 - ▶ **JDialog** - for creating dialog boxes (usually temporary popup messages or dialogs for receiving additional info).

JComponent

Base class for all of the lightweight Swing components, which are graphical items placed on the canvases or containers. Its subclasses are the basic elements for constructing GUIs. Here are just a few of the more common elements:

- ▶ JButton - for creating push buttons
- ▶ JCheckBox - for creating toggle checkboxes
- ▶ JMenu - for pop-up menus
- ▶ JRadioButton - for radio buttons (made into a group, only one can be selected)
- ▶ JLabel - a display area for a short string or image
- ▶ JList - a component allowing the user to select from a list
- ▶ JOptionPane - a component allowing the user to pop up an easy dialog box as an information message or for user input
- ▶ JTextField - component allowing an editable line of text
- ▶ JTextArea - multi-line area for displaying text

Helper classes

Helper classes - used by components and containers to control drawing and placing of objects. Some important helper classes (from package `java.awt`):

- ▶ `Graphics` - abstract class. Provides graphical context for drawing
- ▶ `Color` - used for specifying colors in components and drawings
- ▶ `Font` - specify fonts used in `Graphics` drawings
- ▶ `FontMetrics` - abstract class. Encapsulates information and properties about the rendering of a font on screen
- ▶ `Dimension` - encapsulates width and height of a component in an object

Events

- ▶ Event: A signal that something has happened in a program. Examples: Button clicks, mouse movements, menu selections
- ▶ GUI programs generally driven by events, rather than a specific procedural order
- ▶ Events are handled with event objects. These are triggered by actions on source objects (components or objects on which the event is generated), and they must implement corresponding event listener interfaces. The listener listens for the event, and invokes an event handler when event occurs.
- ▶ `java.util.EventObject`: Base class for event classes in Java

Some examples of event types:

These are just a few examples, not a comprehensive list.

- ▶ `ActionEvent` - clicking a button, pressing return on a text field
- ▶ `ItemEvent` - clicking a check box, selecting an item
- ▶ `WindowEvent` - Closing a window, opening a window
- ▶ `ContainerEvent` - component added to a container
- ▶ `ComponentEvent` - resizing a component, hiding a component
- ▶ `TextEvent` - changing a text value
- ▶ `MouseEvent` - clicking the mouse, dragging the mouse
- ▶ `KeyEvent` - pressing a key on the keyboard