Introduction to Graphical User Interfaces

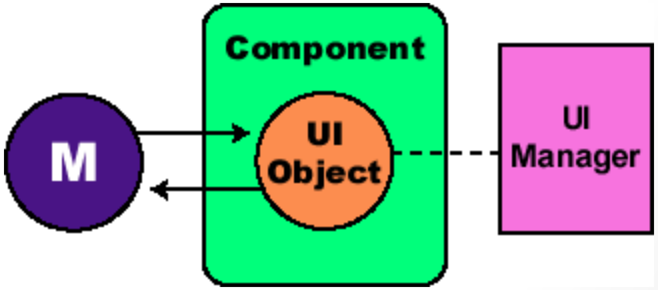
Terminology, common components, AWT and Swing, events, listeners

GUI terminology

- The **user interface** of a program is the part that the user views and interacts with.
- Programs that interact with users via windows and other visual components have **graphical user interfaces**. They are sometimes called **GUIs**.
- GUIs have two primary features:
  - **Widgets**: user controls (buttons, menus, etc)
  - **Event-driven actions**: responses to users.

Model, View, Controller & GUIs

- MVC is a common design pattern for developing programs
  - Model: backend for an application.
  - View: visualization of the backend.
  - Controller: translates between view and model
- GUIs tightly couple the View and Controller
  - View (how the object is presented) and controller (how events on the object are processed) belong to one UI object.
  - GUI components (such as JButton) delegate display to the underlying "look and feel" of the operating system.



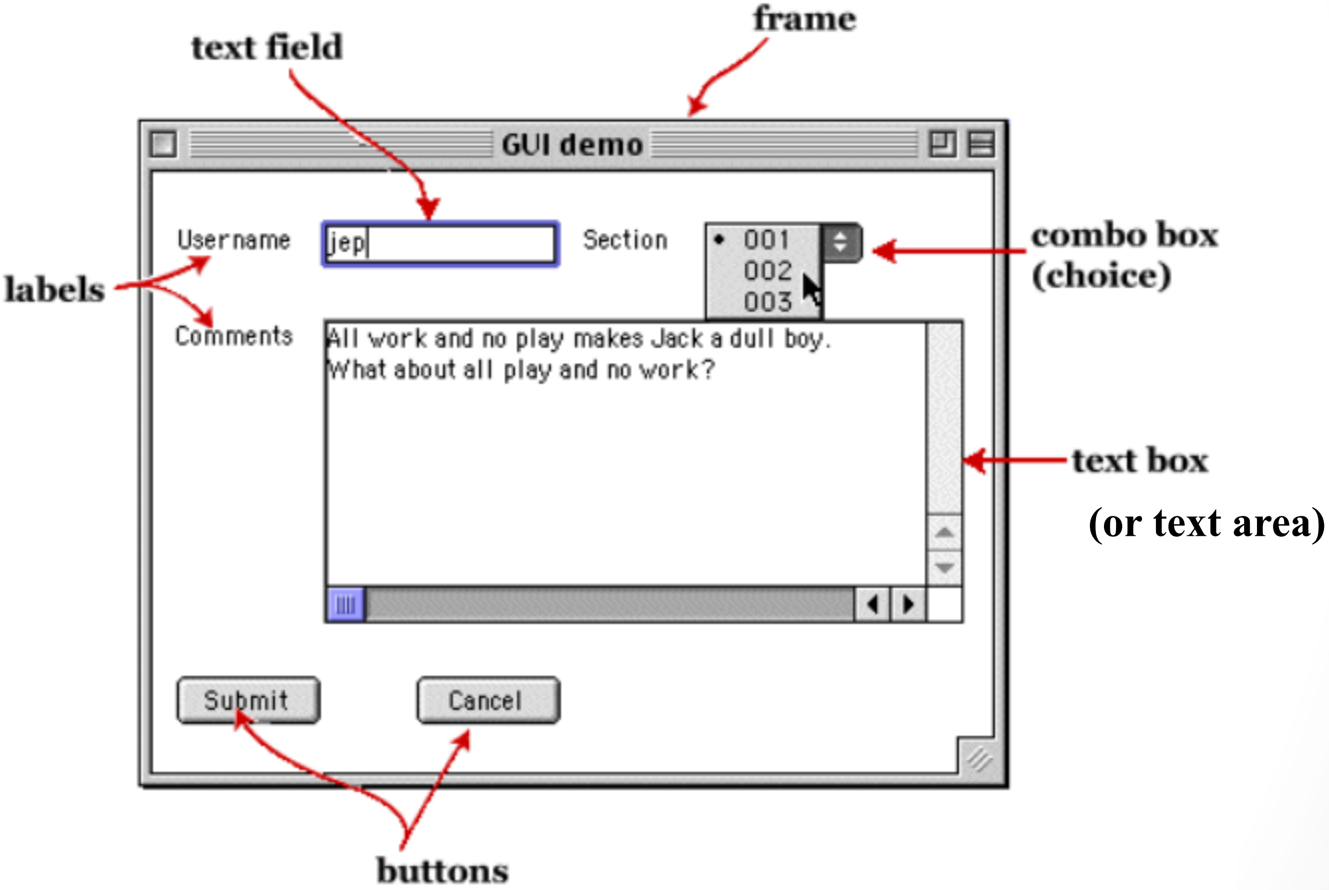
Components, containers, and widgets

- **Components** are graphical elements of GUIs.
  - Component class is the abstract superclass for all widgets.
  - Component is defined in AWT.
- **Containers** are components designed to hold other components (such as windows).
  - Container class extends Component
  - Container is abstract
- **Widgets**, such as labels, buttons and text fields, belong to containers.
  - Containers *add* widgets to their display.
  - In order for a widget to detect when a user interacts with it, the widget must add an **event listener**.

Common components

- **Container**. Can hold other components (including other containers).
- **Frame**. Top level window with a title and border. Designed to hold a container for actual content.
- **Button**. Clickable, labeled rectangle (or roundtangle).
- **Label**. Read-only text.
- **Radio button**. Special kind of button to be used in a group for single choice selection.
- **Choice/combo-box**. Popup menu in a fixed position.
- **Checkbox**. Clickable rectangle for checking on/off toggle.
- **Text field**. Holds a single line of text, editable by default.
- **Text area**. Rectangular area for multiple lines of text.
- **Canvas**. Rectangular area for graphic objects (lines, circles, etc)
- **Panel**. Concrete container.

Component examples



Username: jep

Comments: All work and no play makes Jack a dull boy.
What about all play and no work?

text field

frame

labels

combo box
(choice)

text box

(or text area)

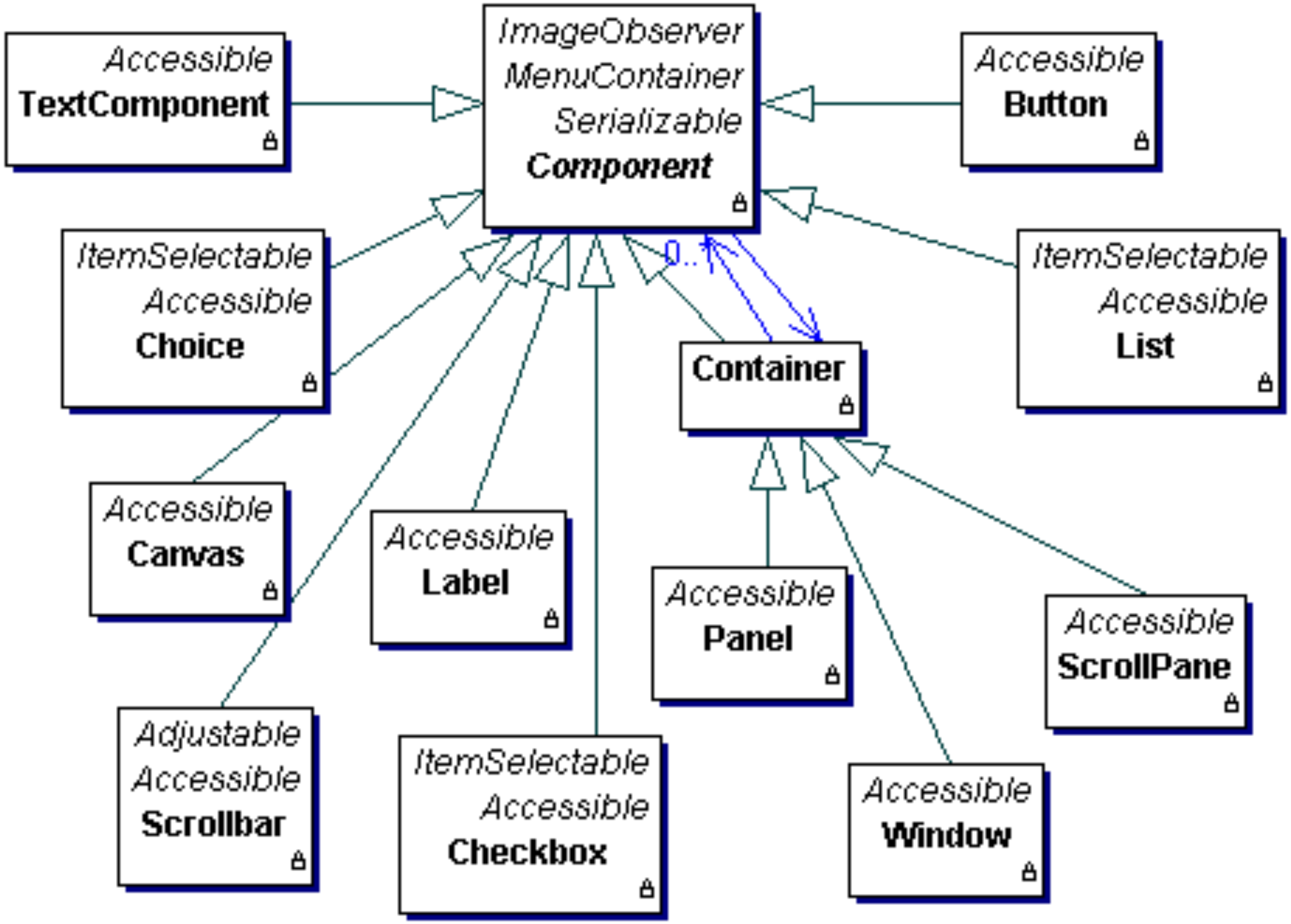
buttons

AWT and Swing

- Two GUI packages from the Java API:
  1. **AWT** (Abstract Windowing Toolkit): original package containing GUI classes and interfaces. Size ~ 450.
  2. **Swing**: newer package that uses the native OS look and feel. (What you see depends on what operating system you use.) Size ~ 1400.
- Typical GUI programs use both packages.
- The two have some corresponding components. Swing class names typically begin with "J":
  - Frame - JFrame
  - Button – JButton
  - Label - JLabel

AWT/Swing organization

AWT has ~ 450 classes and interfaces.



AWT/Swing use

- java.awt (AWT) has interfaces, classes, exceptions, errors, enumerations, and 11 subpackages, including:
  - java.awt.color
  - java.awt.font
  - java.awt.event
- javax.swing (Swing) also has interfaces, classes, exceptions, errors, enumerations, and subpackages.
- Import to use them:
  - AWT: `import java.awt.*;`
  - Events: `import java.awt.event.*;`
  - Swing: `import javax.swing.*;`

Import the classes directly instead of using * to keep a smaller namespace.

Event-driven programs

- A user interaction with a GUI program creates an **event**.
- **Event-driven** programs respond to user interactions or events.
- Events are objects that are generated when the user clicks a button, presses a key, selects text, or such. Events have **sources**, (buttons, text fields etc) and **handlers** to determine the resulting action.
- Some events cause actions to be performed.
- Processes behind the actions can be **backend** (not related to the GUI).
- **Event listeners** are objects that can tell when events occur. Buttons or other components with which the user can interact must register listeners, which handle the events. For example, an action listener for a button can tell ("listen for") when the user clicks the button and supply the appropriate resulting action.
- Listeners are defined in the java.awt.event package.

Listeners

- **Event listeners** are objects that determine when events occur.
- Widgets must *register* listeners if there is an expected response. For example, an action listener for a button can determine ("listen for") when the user clicks the button.
- There are 18 different kinds of listeners, including:
  - Action listeners
  - Mouse listeners
  - Key listeners
  - Window listeners
- Listeners are defined in the java.awt.event package. import java.awt.event.*;