

# Widgets

More GUI goo

# Outline

- What is a widget?
- Buttons
- Combo boxes
- Tool bars
- Text components
- Dialog boxes
- Sliders
- Scrollbars

# What is a Widget?

- Widgets are interactive objects - the basic building blocks of graphical user interfaces
- Used for input and output
- Input - information and commands
- Output - information and application status
- An important part of user interface design is choosing and laying out widgets on the screen

# Categories of Widgets

- Two broad categories
- Meta-level widgets – widgets that contain other widgets (aka compound component)
- Atomic widgets – atomic components (cannot contain other components)
- Examples...
- Meta-level widgets
  - Windows
  - Dialog/message boxes
  - Toolbars
  - Panels (JPanel)
- Atomic widgets
  - Buttons
  - Text components
  - Sliders
  - Scrollbars
  - Combo boxes
  - Menus

# Outline

- What is a widget?
- Buttons
- Combo boxes
- Tool bars
- Text components
- Dialog boxes
- Sliders
- Scrollbars



# What is a Button?

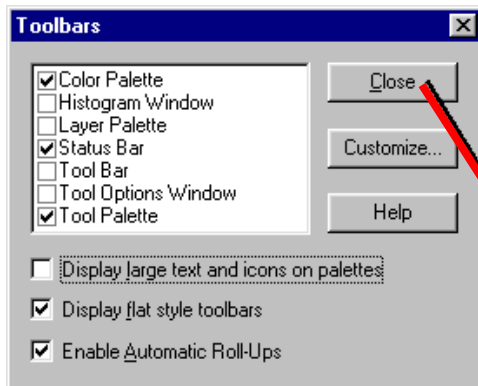
- An individual and isolated region of a display that is selectable by the user to invoke a state or command
- Designed to look like physical buttons to suggest they can be “pushed” (like physical buttons)
- Four types
  - Push buttons - invoke a command
  - Radio buttons - enable a state and indirectly disable one or more other states
  - Checkboxes - toggle a state on/off
  - Modal buttons – hybrid of push button and checkbox (aka toggle buttons)

# Push Buttons

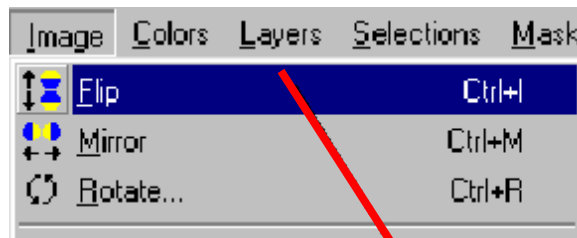
- The most common type
- Pushed (“clicked”) to invoke a command
- Three ways to organize push buttons...
  - Isolated
    - An individual button in a window
    - Pushed (“clicked”) to invoke a command (e.g., Cancel, OK, Close, Preview)
  - Menus
    - A menu is simply a popup collection of buttons
  - Toolbars
    - A collection of buttons with commonly used functions
    - Presentation is “persistent” (rather than popup, as with menus)
    - Usually employ a small icon, rather than text

# Push Button Examples

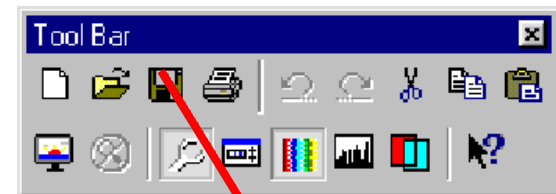
Isolated



Menu



Toolbar



Push button

Push button

Push button

# Radio Buttons

- Allow a user to select one feature from a set of mutually exclusive features
- Name derived from radio station selector buttons

# Radio Button Example

Three mutually-exclusive options

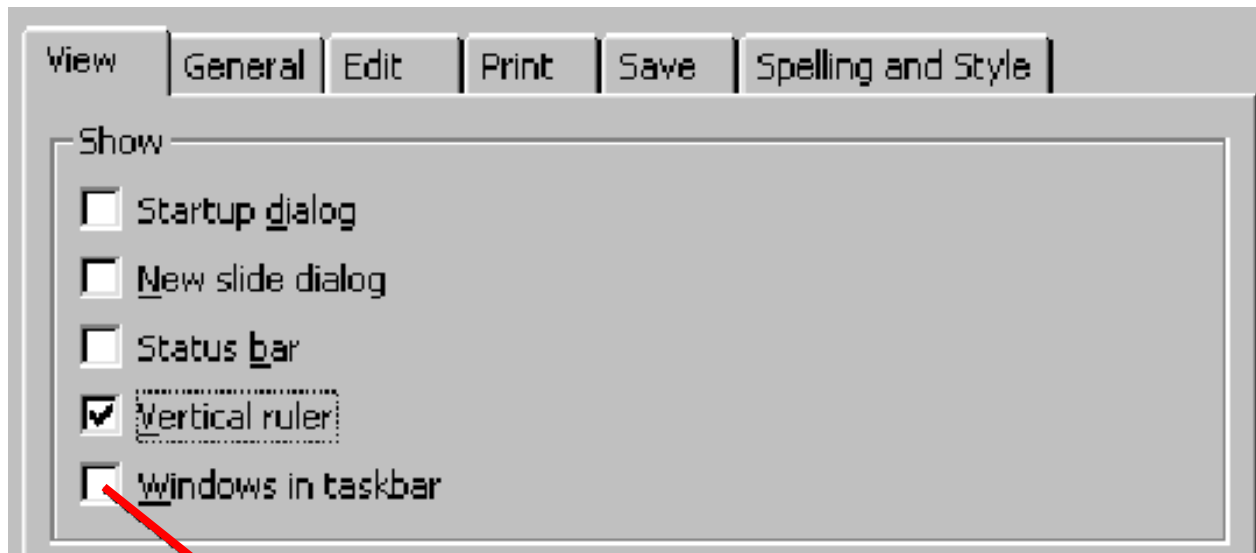
Navigator starts with

- Blank page
- Home page
- Last page visited

# Checkboxes

- “Checked” to turn a state on/off
- Use checkboxes...
  - When a feature or characteristic has precisely two states and these are appropriately described as being “enabled” or “on” vs. “disabled or “off” (e.g., lightbulb = on | off)
- Do not use checkboxes...
  - When a feature or characteristic has more than two states (e.g., season = spring | summer | autumn | winter)
  - When enabled/on and disabled/off are inappropriate descriptions of the states (e.g., gender = male | female)

# Checkbox Example

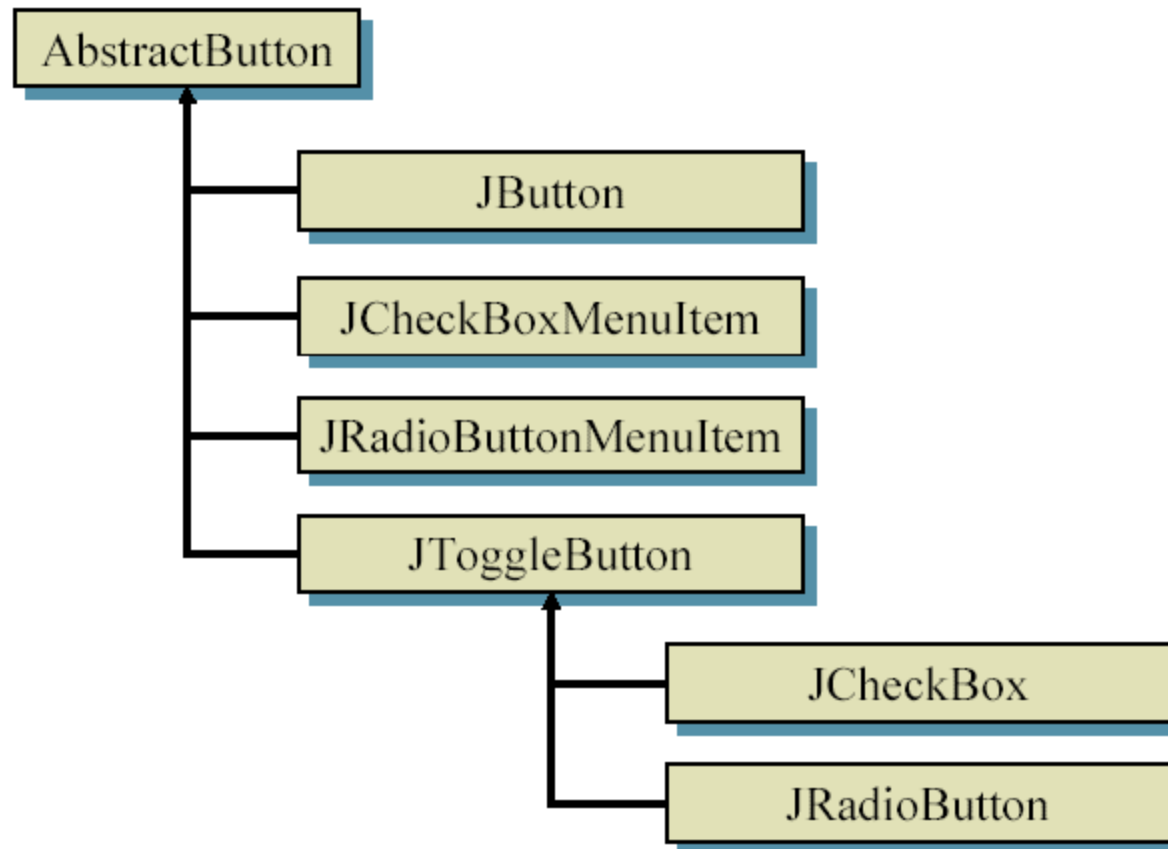


Checked = Enable "show windows in taskbar"  
Unchecked = Disable "show windows in taskbar"

# Modal Buttons

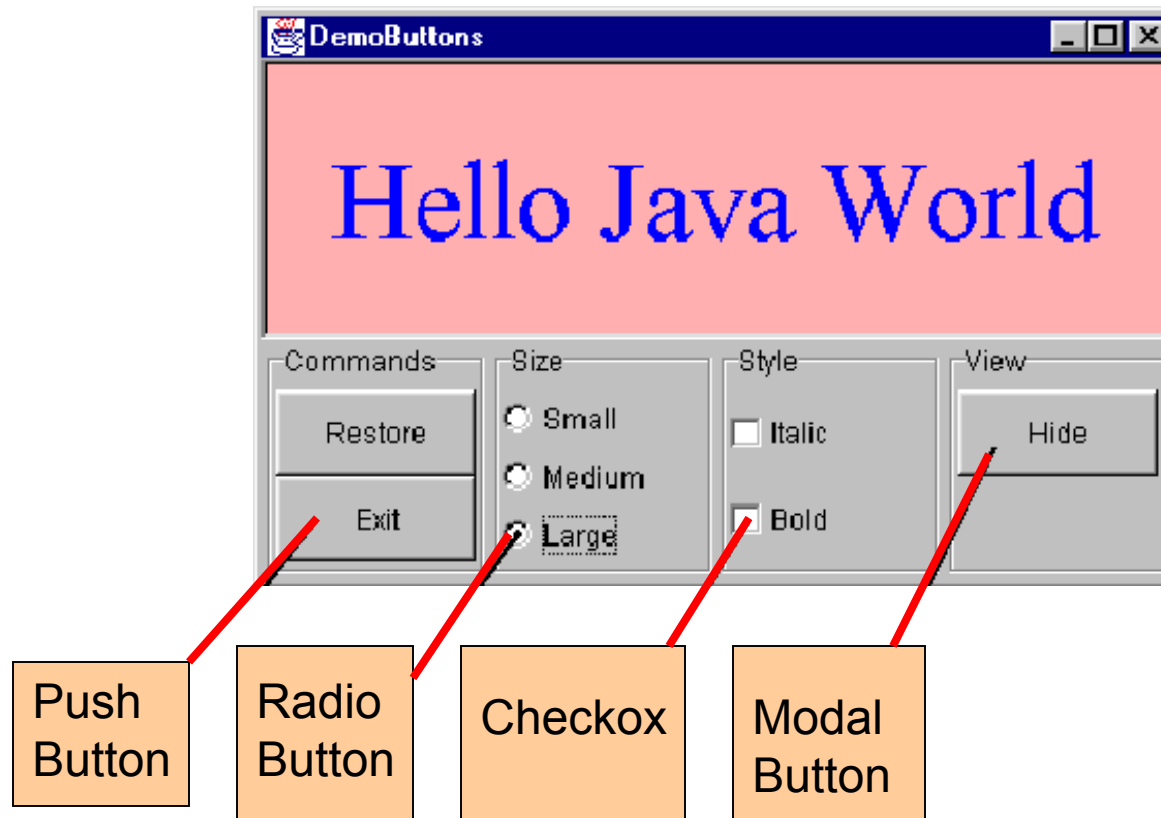
- A hybrid of push buttons and checkboxes
- A modal button...
  - Looks like a push button
  - Toggles a state
- Text or icon in button indicates the next state
  - E.g., “show details” vs. “hide details” (example soon)

# Java's Button Classes



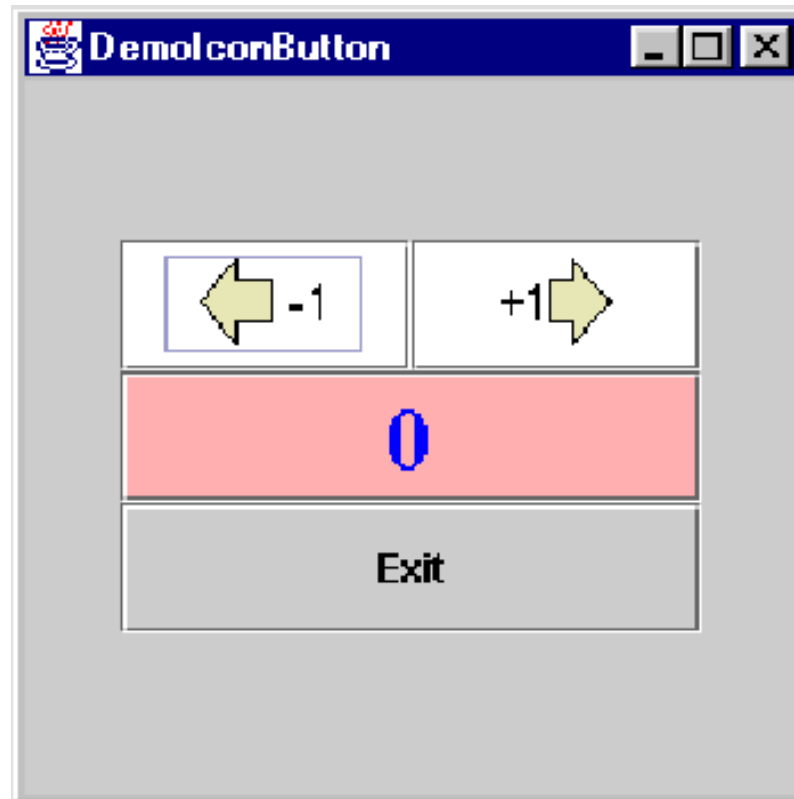
# Example Program

## DemoButtons.java



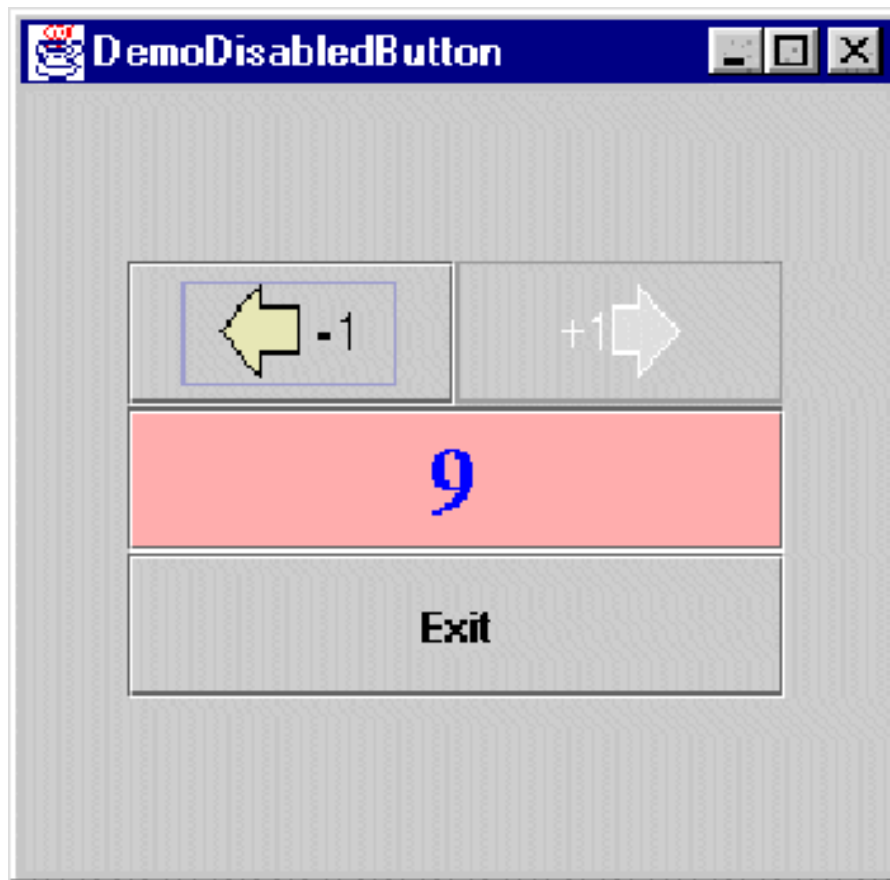
# Example Program

## DemolIconButton.java



# Example Program

## DemoDisabledButton.java



# Example Program

**DemoHTMLButton.java**



# Outline

- What is a widget?
- Buttons
- Combo boxes
- Tool bars
- Text components
- Dialog boxes
- Sliders
- Scrollbars

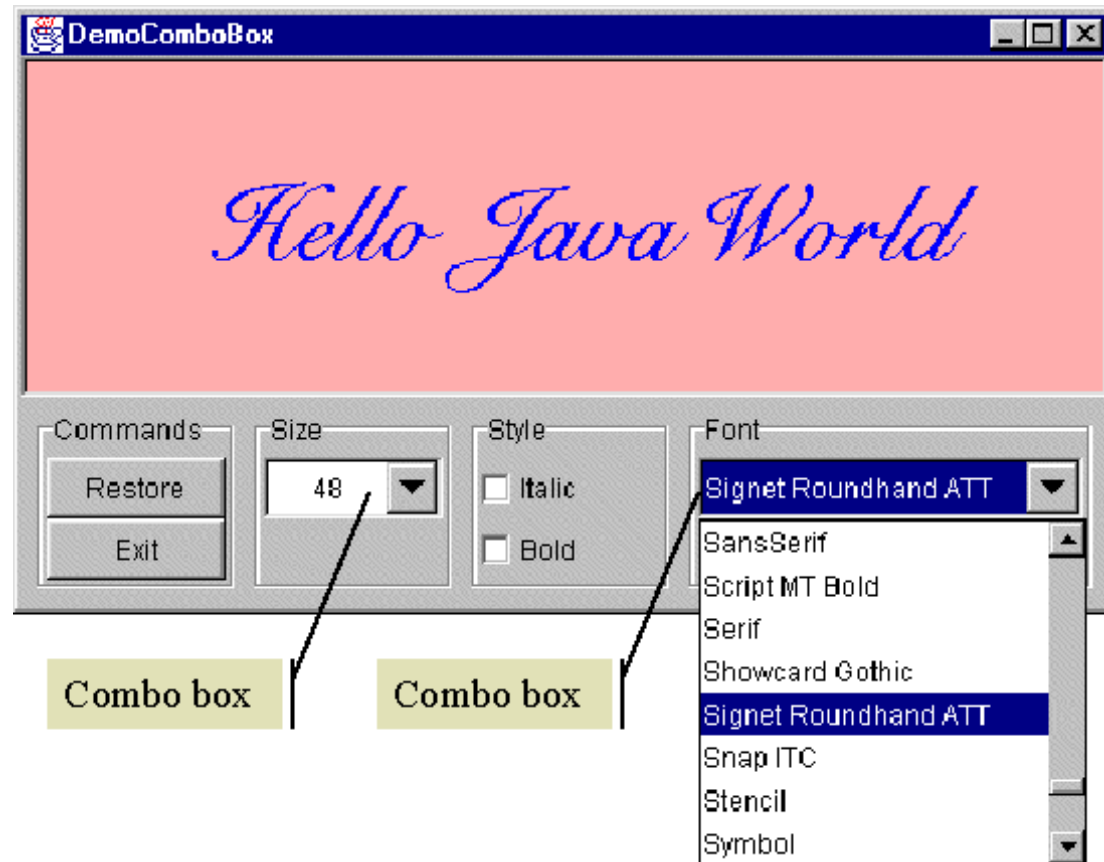


# Combo Boxes

- An alternative to radio buttons when the number of mutually-exclusive options is large
- Advantage over radio buttons
  - More choices can be displayed in less screen space
- Disadvantage over radio buttons
  - Choices are not displayed until combo box is selected (remember 'recognize' vs. 'recall')

# Example Program

## DemoComboBox.java



# Outline

- What is a widget?
- Buttons
- Combo boxes
- Tool bars
- Text components
- Dialog boxes
- Sliders
- Scrollbars



# What is a tool bar?

- A group of buttons arranged in a row or column
- Tool bar buttons usually display an icon, rather than text
- Provide convenient access to common menu commands
- Often, the tool bar can be dragged into a separate window
- Advantage of tool bars
  - Commands are displayed (remember 'recognize' vs 'recall')
- Disadvantage of tool bars
  - Consume display space (less space for application)

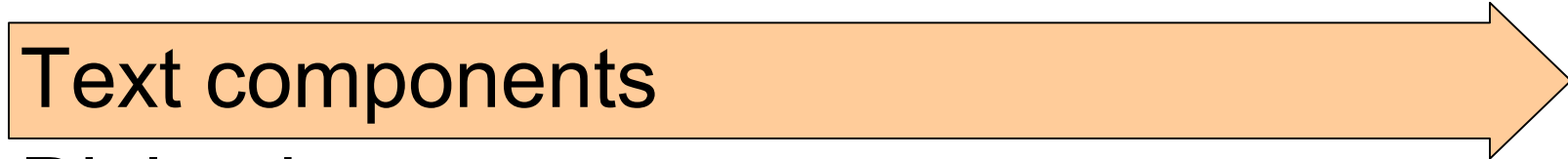
# Example Program

## DemoToolBar.java



# Outline

- What is a widget?
- Buttons
- Combo boxes
- Tool bars
- Text components
- Dialog boxes
- Sliders
- Scrollbars



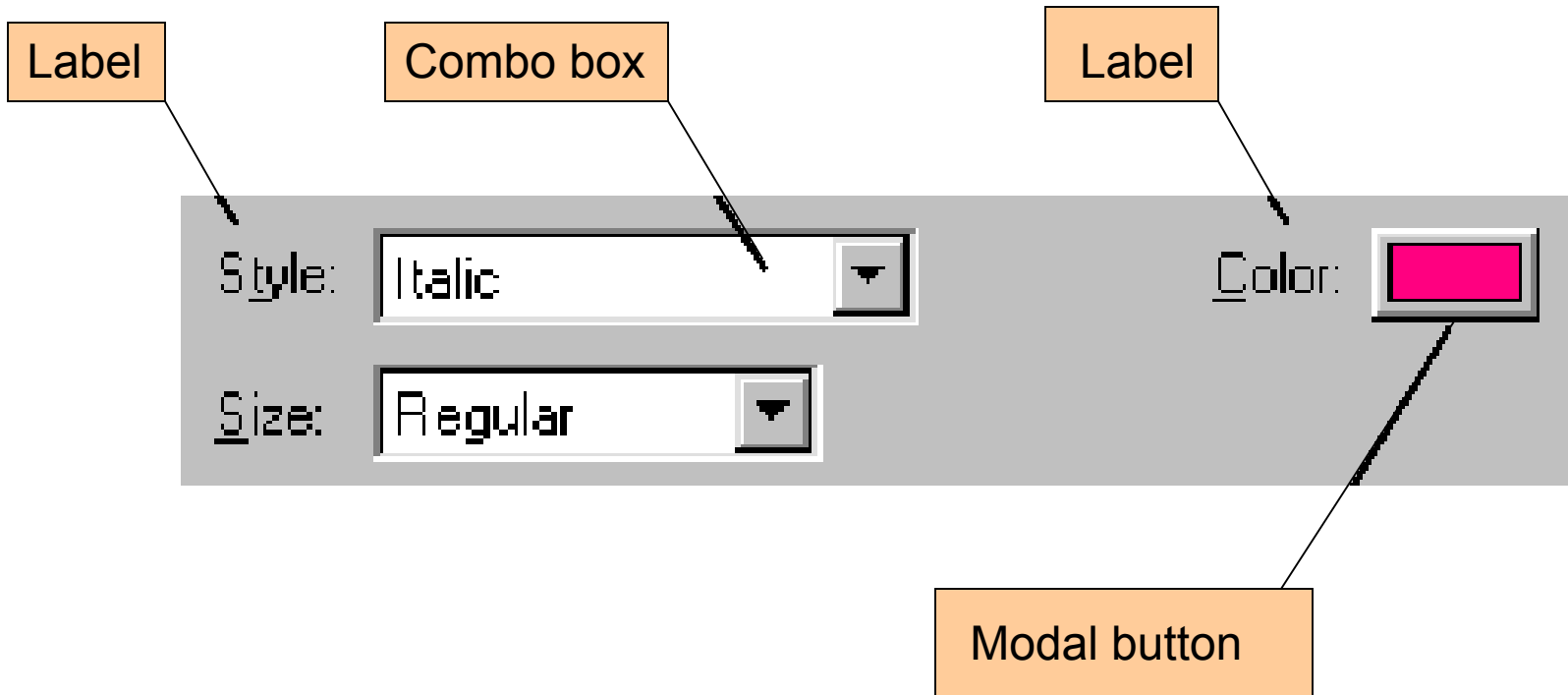
# Types of Text Components

- Output (aka non-editable)
  - Labels
  - Labeled borders
  - Tool tips
  - Message Boxes
- Input/output (aka editable)
  - Text fields
  - Text areas
  - Editable combo boxes
  - Dialog boxes

# Labels

- Placed near (above, below, left, or right) a widget to provide additional information on the purpose
- Advantage
  - Aides in command recognition
- Disadvantage
  - Uses display space
  - Terse wording may be a poor indicator of command purpose

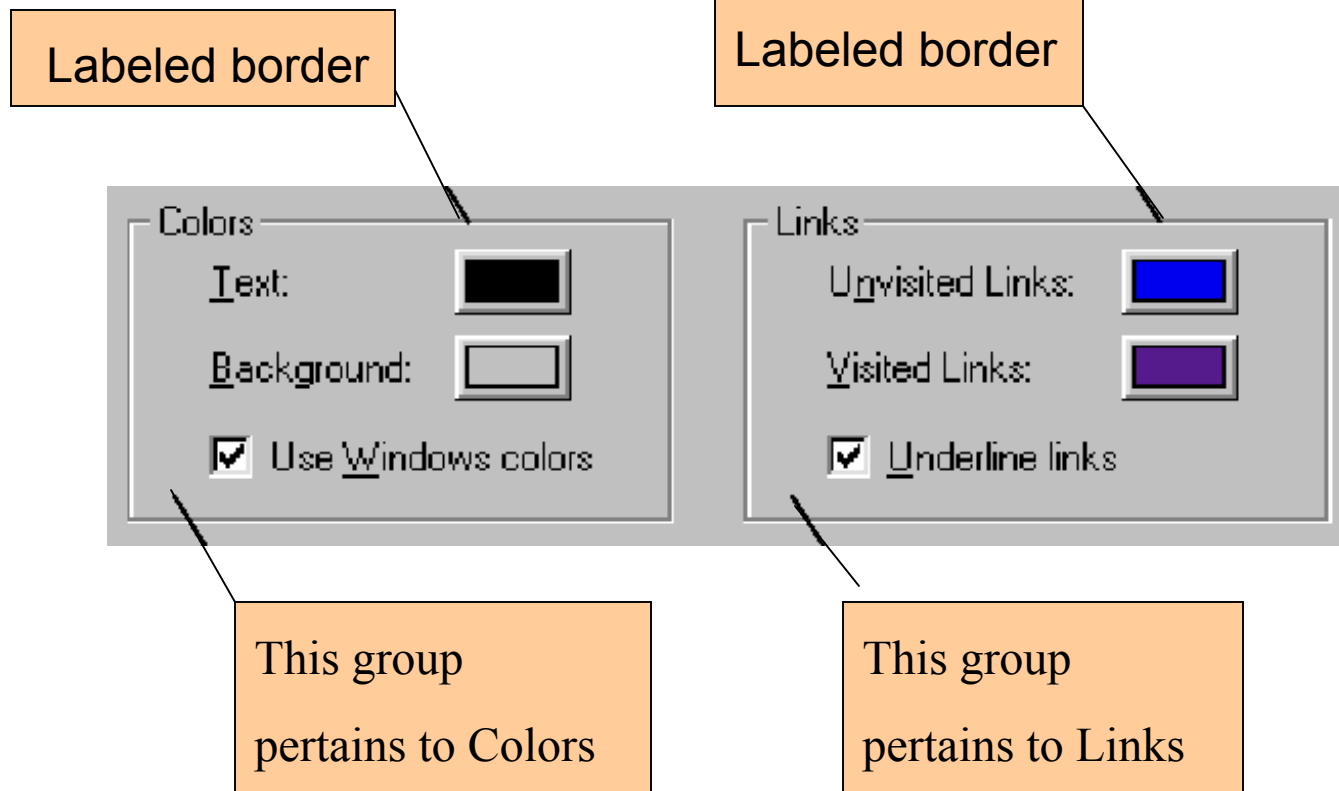
# Label Example



# Borders and Labeled Borders

- Placed around a collection of widgets
- Use to create 'groups'
- Advantage
  - Grouping reduces *cognitive load*
    - E.g., 3 groups, each with 4 items
    - With borders around groups: user identifies 1 of 3 groups, then 1 of 4 items within group (two-step process)
    - Without borders: user identifies 1 of 12 items (this is harder!)
- Disadvantage
  - Uses display space
  - Group 'boundaries' may not be obvious
- Labels may be added to improve group identification

# Labeled Borders



# Design Opportunity

Could the organization of these option widgets be improved using borders to create groups?

POP account:

Check for mail every  minutes

Skip messages over  K in size

Send on check

Save password

Leave mail on server

Delete from server after  days

Determine first unread message by:

First message not read by this machine

Status: headers

POP3 LAST command

Delete from server when emptied from Trash

Authentication style:

Passwords  Kerberos  APOP

# Tool Tips

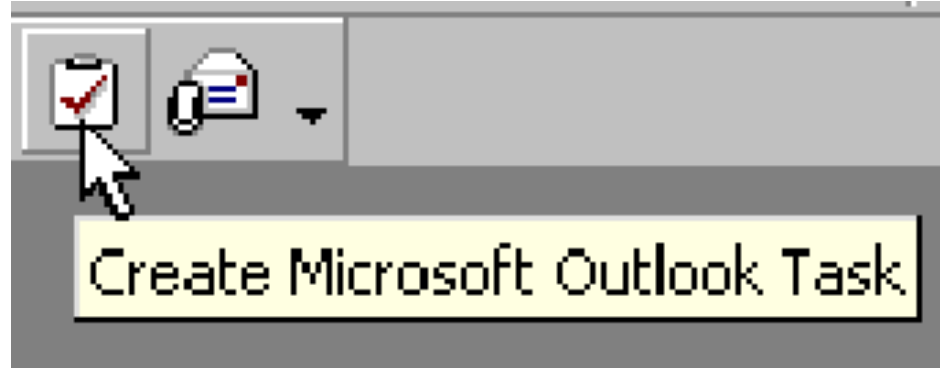
- A tool tip is a small text field that pops up when the cursor pauses over a GUI component
- The text provides a terse supplementary description of the command associated with, or the purpose of, the component
- Advantage
  - Aides in command recognition
  - Does not consume screen space until shown
- Disadvantage
  - Terse wording may be inadequate
  - Frequent cycles of appear/disappear (i.e., animation) annoying to some users

# Tool Tip Example



# Tool Tip Example (2)

- From MS PowerPoint...



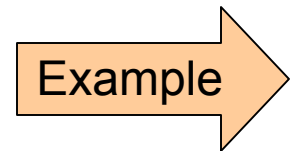
?

# Message Boxes

- A message box is popup window that presents a text message to the user
- Output only (except for confirmation to close the box)
- Purpose
  - Notify the user of a problem (e.g., invalid choice)
  - Notify the user of potentially destructive outcome (e.g., overwrite a file)
  - Provide information

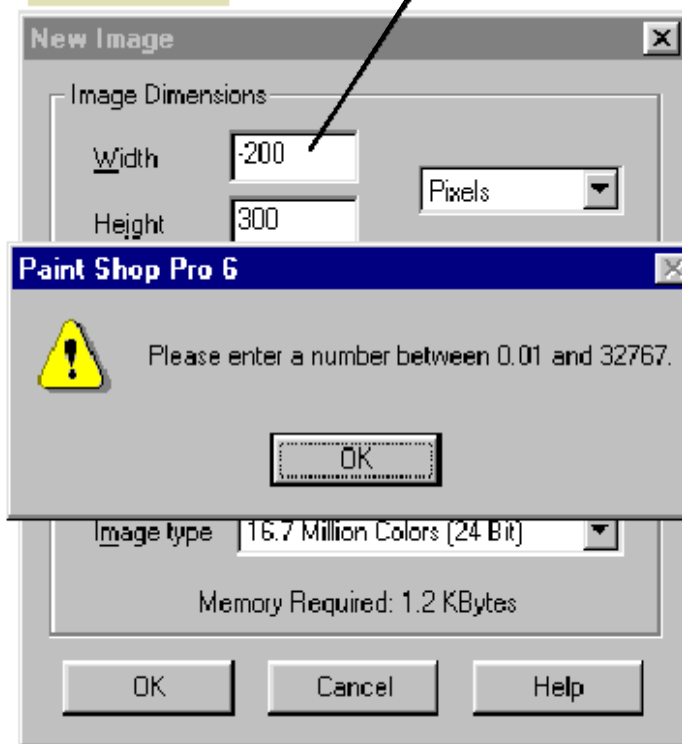
# Message Boxes (2)

- Advantage
  - Comprehensive messages are possible (unlike tool tips)
- Disadvantage
  - Slows interaction (because underlying thread is halted until confirmation is received)



Message box

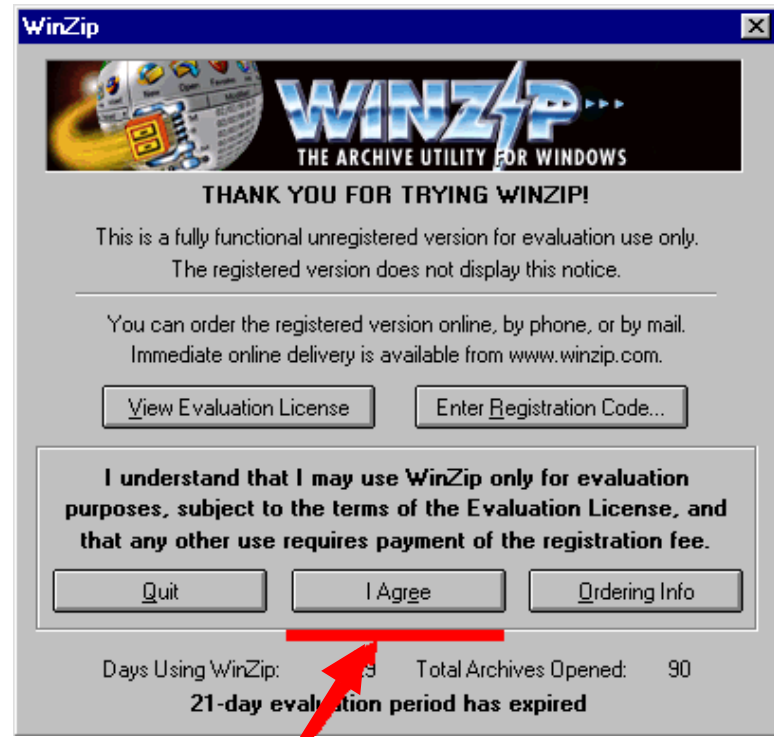
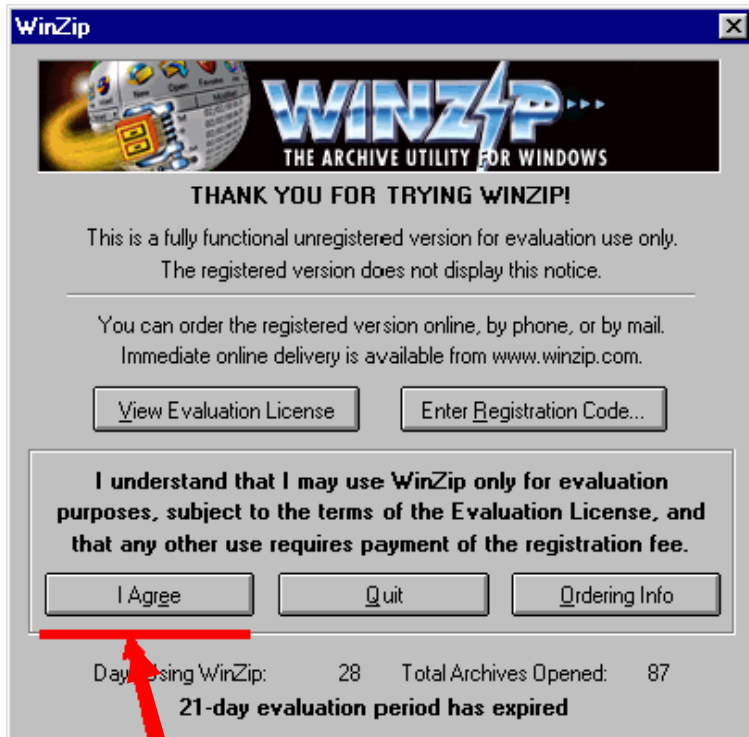
Invalid input



Message box

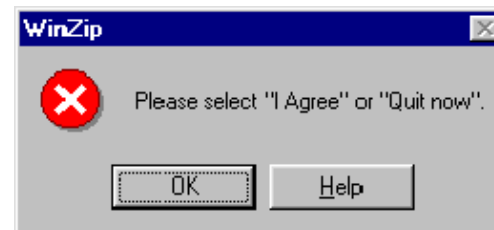
# Confirmation and Error Prevention

- Confirmation and Error Prevention
- Most dialog or message boxes require the user to make a choice or acknowledge a message
- Many don't have close boxes (user cannot choose other widgets while dialog/alert box is active)
- Three error prevention techniques
  1. Require an explicit button click or key press before proceeding (i.e., there is no default action by pressing Enter)
  2. Sound an alarm tone for invalid input
  3. Change the position of buttons from one invocation to the next (prevents "errors due to habit!")
- E.g., Winzip (next slide)



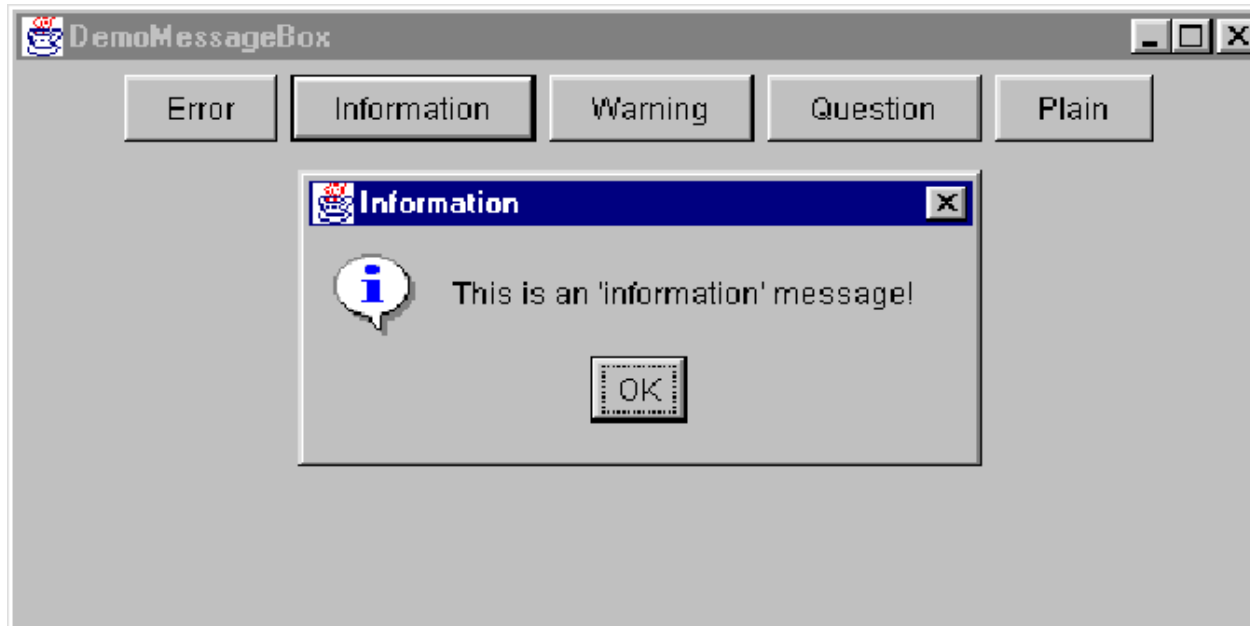
Button positions change from one invocation to next

Hitting ENTER produces...



# Example Program

## DemoMessageBox.java



# Text Fields and Text Areas

- Text Field: single-line input/output of text
- Text Area: multi-line input/output of text
- Design challenges
  - Validating input
  - Navigating

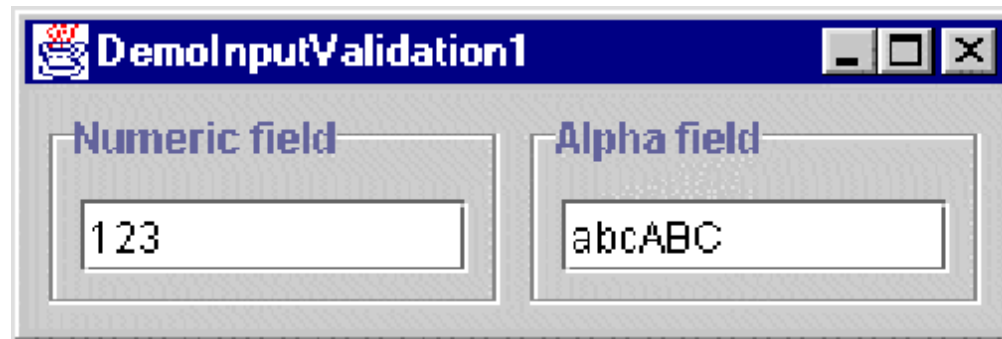
# Validating Input

- Input data must be in the format required by the application
- Examples:
  - Numeric field with value in certain range
  - Postal or zip code
  - Date/time formats
- Invalid input must be corrected before proceeding
- If invalid input, can present a popup message, generate audio alarm, etc.

# Example Program

**DemoInputValidation1.java**

**DemoInputValidation2.java**



# Navigating

- The process of moving from one component to the next
- The currently active component has focus (identified with
- I-beam cursor or special highlighting)
- E.g., completing a form with numerous fields
- Navigation possibilities
  - Clicking with mouse pointer
    - Should accept and validate entry in current field and advance focus to the selected field
    - Sequence determined by user, not the application
  - Tab or Enter
    - Should accept and validate entry and go to the 'next' field
    - Sequence determined by application, not the user

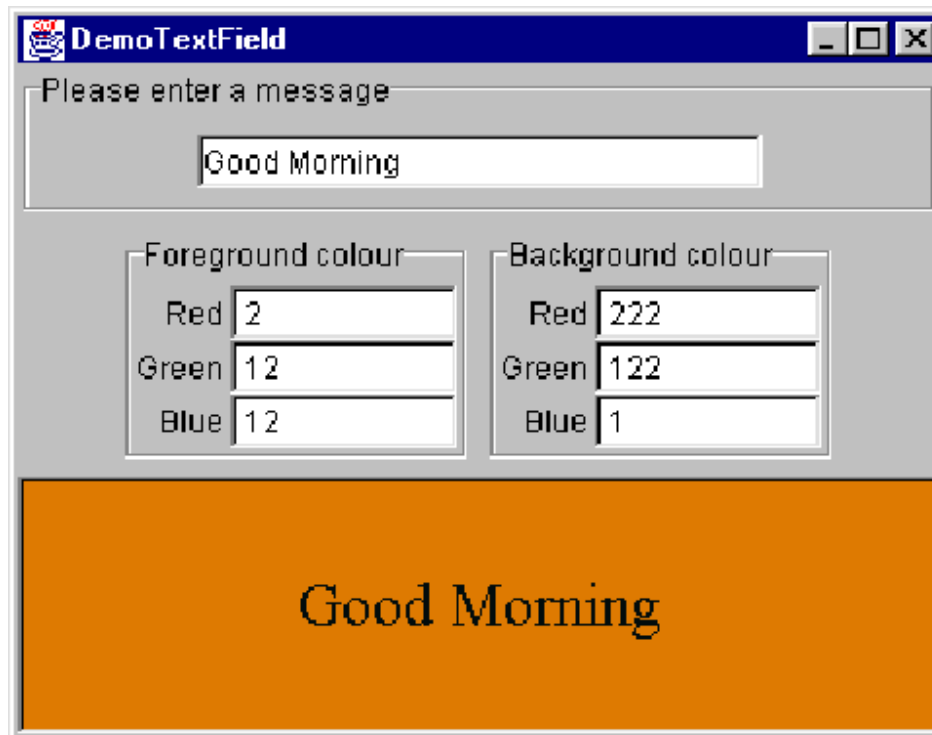
# Design Guideline

- A user should be able to interact with a graphical user interface using...
  - mouse only
  - keyboard only
  - combination of mouse and keyboard
- A tall order
  - E.g., text input via a mouse is difficult (but not impossible!)

# Example Program

**DemoTextField.java**

**DemoTextField2.java**




# Editable Combo Boxes

- Like a combo box, except allows the user to directly type in an entry
- Much more complicated due to...
  - Validating input
  - Navigation

# Example Program

## **DemoComboBox2.java**

Same as DemoComboBox.java,  
except uses an editable combo box to  
specify font size.



# Outline

- What is a widget?
- Buttons
- Combo boxes
- Tool bars
- Text components
- Dialog boxes
- Sliders
- Scrollbars



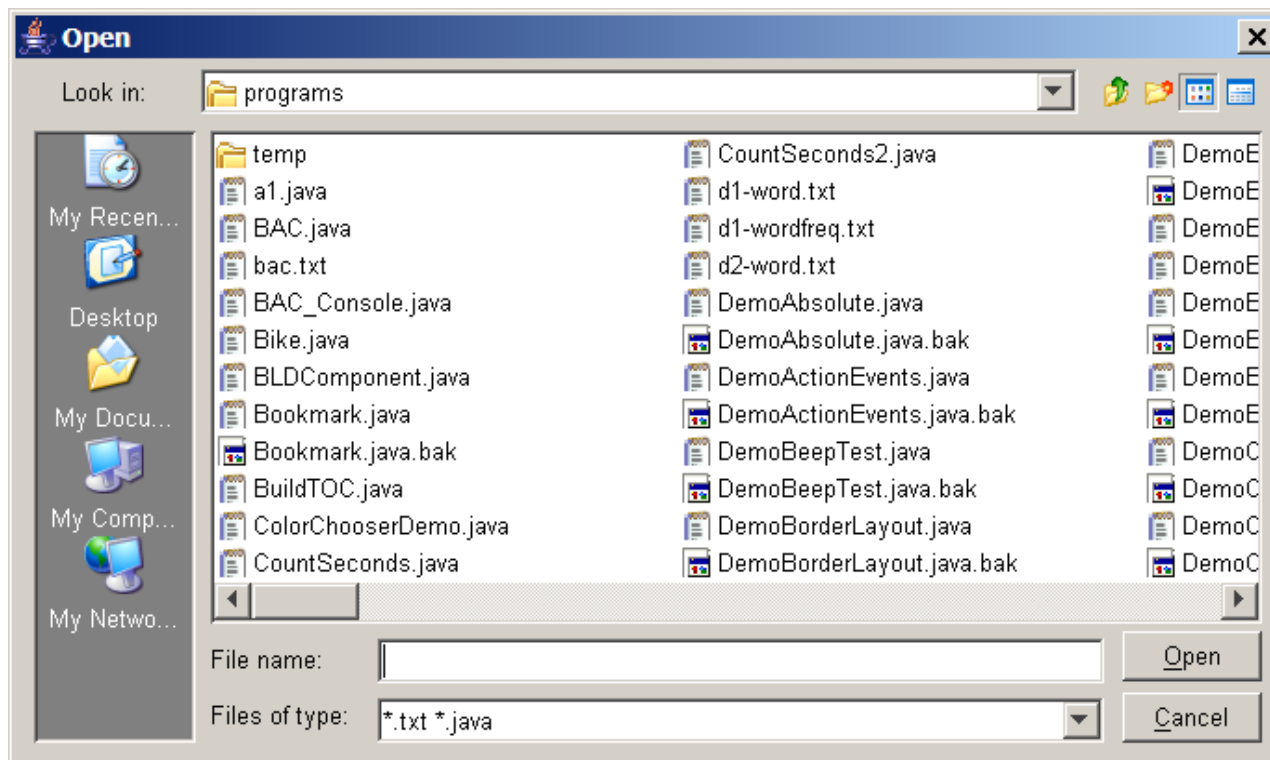
# Dialog Boxes

- File Chooser
- Color Chooser
- Custom dialogs

# File Chooser Example Program

**DemoFileMenu.java**

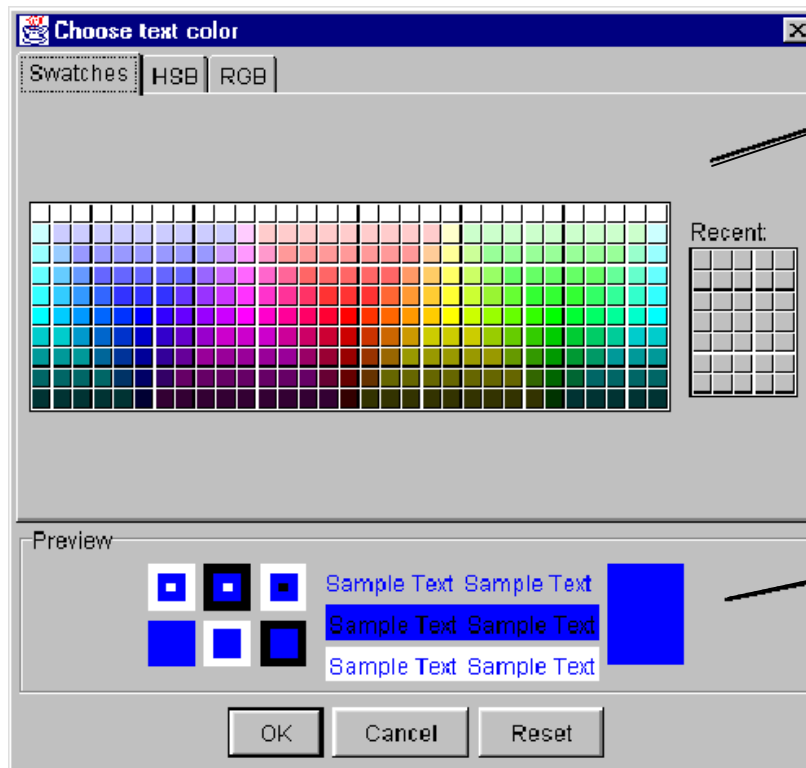
Shown earlier



# Color Chooser Example Program

**DemoMenu2.java**

Shown earlier

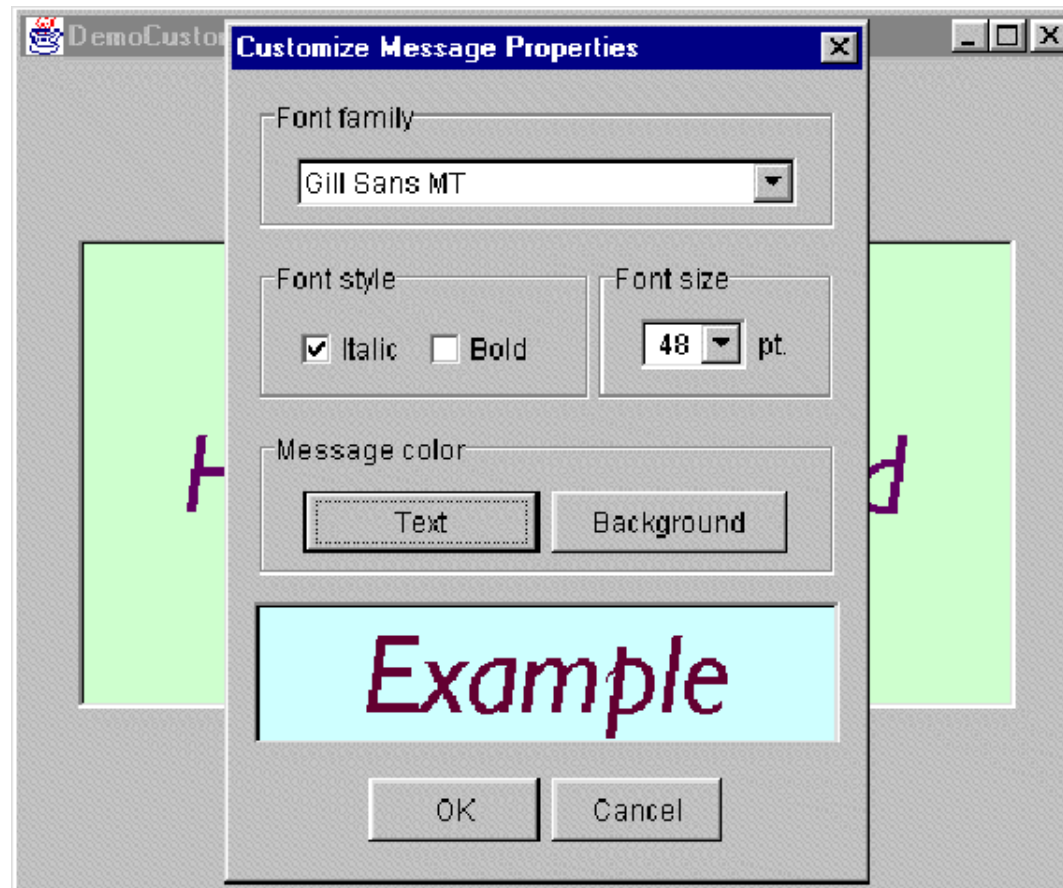


Tabbed pane

Preview panel

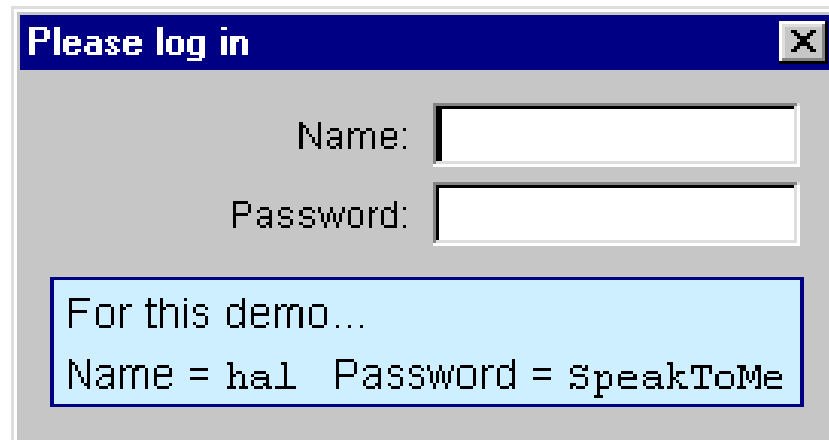
# Custom Dialog Boxes

## DemoCustomDialog.java



# Password Dialog

## DemoPassword.java



Please log in

Name:

Password:

For this demo...  
Name = hal Password = SpeakToMe

# Outline

- What is a widget?
- Buttons
- Combo boxes
- Tool bars
- Text components
- Dialog boxes
- Sliders
- Scrollbars



# Sliders

- Input control to establish a setting for a pseudo-continuous variable
- Examples
  - RGB values in a color chooser
  - Time delays for animations

# Example Program

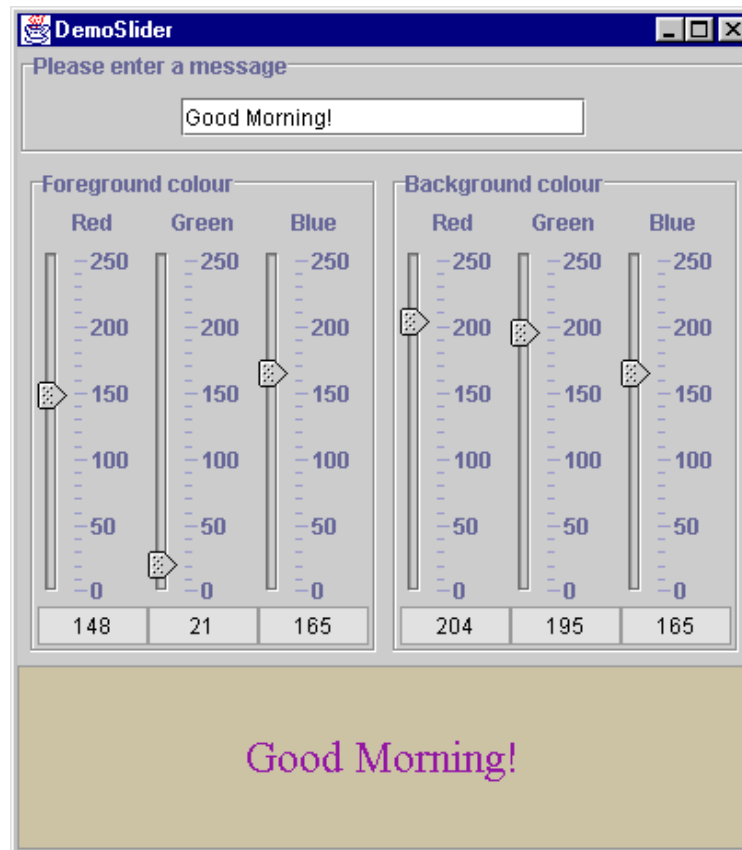
Bare Bones!

## DemoSliderBB.java



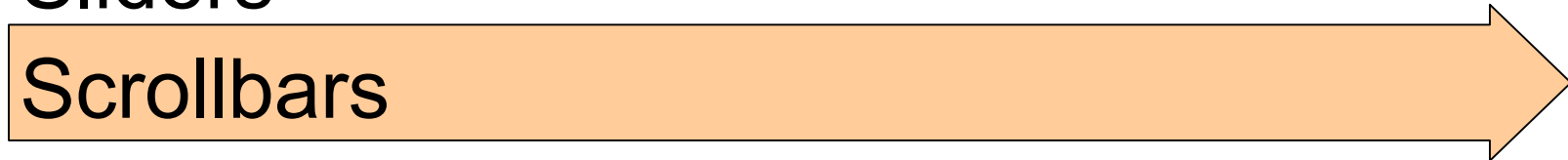
# Example Program

## DemoSlider.java



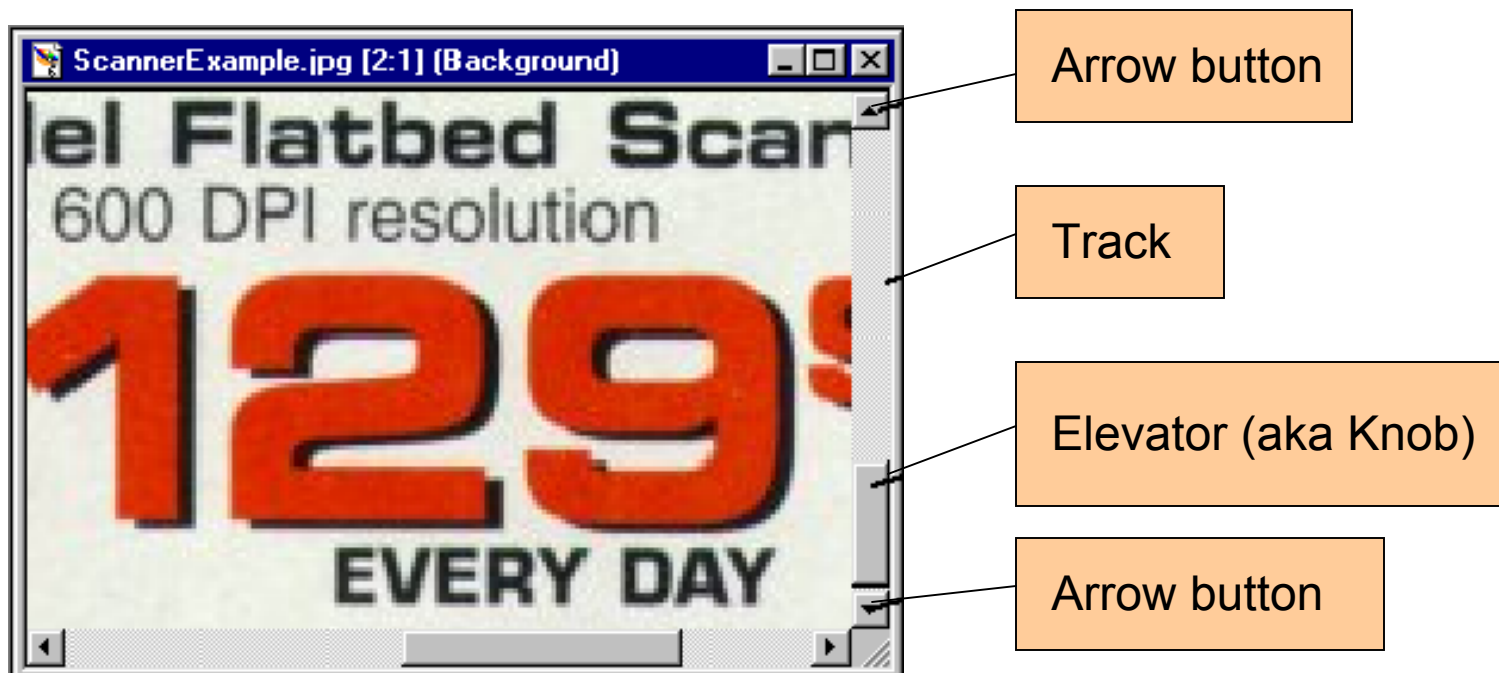
# Outline

- What is a widget?
- Buttons
- Combo boxes
- Tool bars
- Text components
- Dialog boxes
- Sliders
- Scrollbars



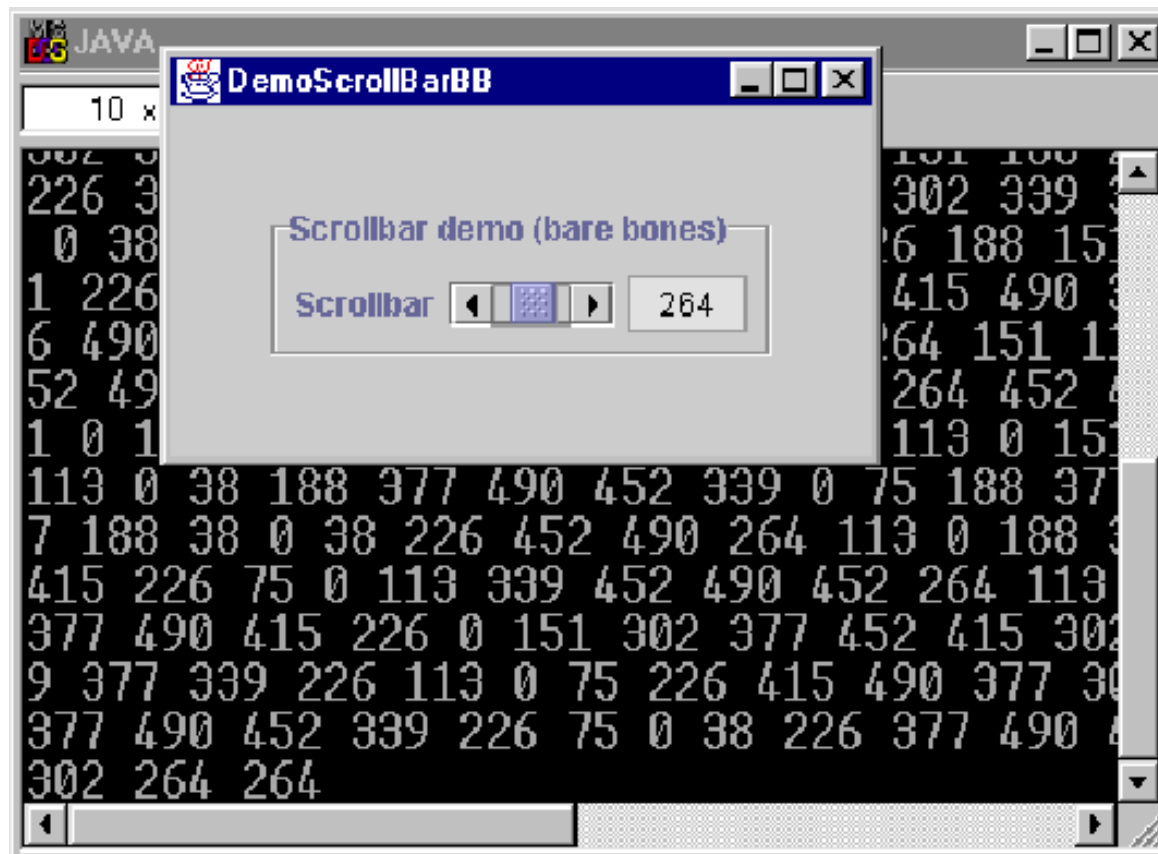
# Scrollbars

- Ubiquitous in Graphical User Interfaces
- Parts of a scrollbar

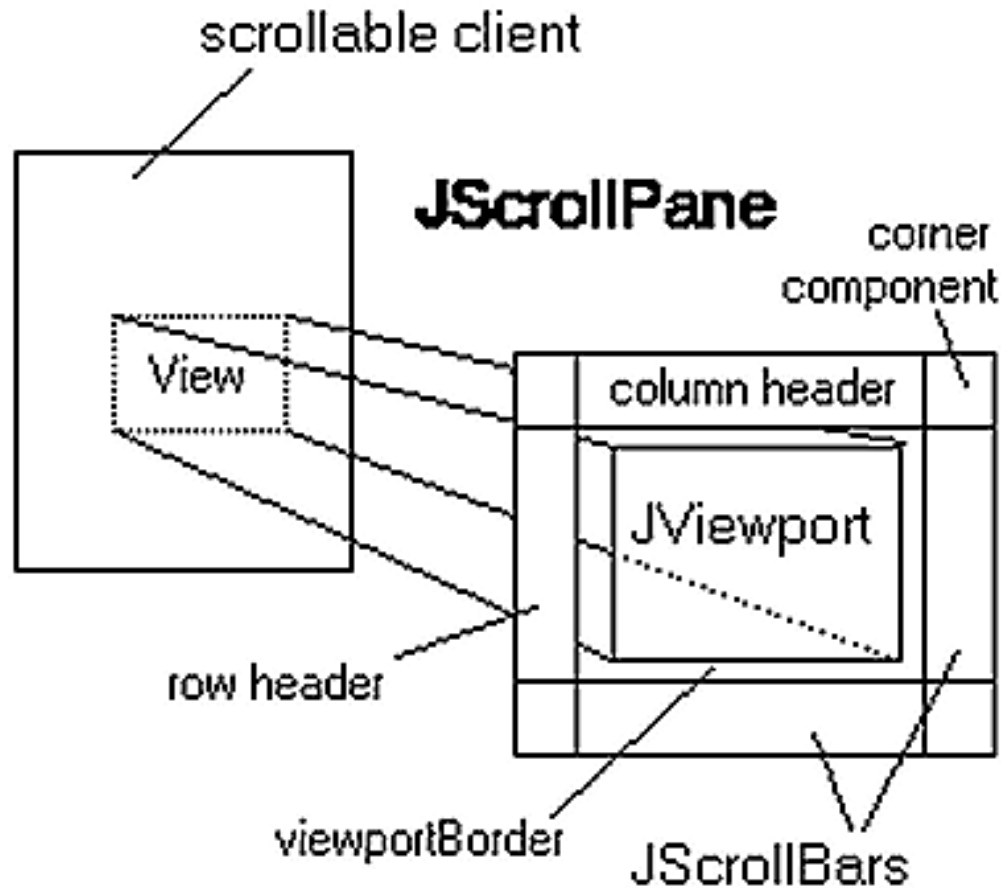


# Example Program

## DemoScrollBarBB.java



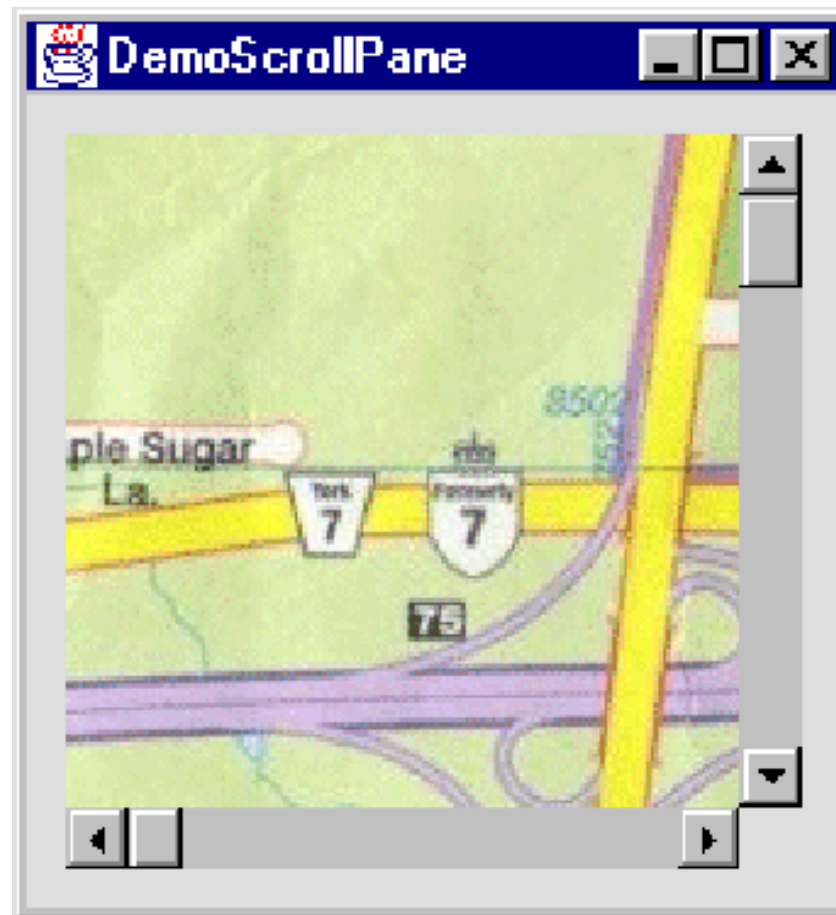
# Viewports and Scrollable Clients



Source: JScrollPane API Specification

# Example Program

**DemoScrollPane.java**



# Example Program

## DemoScrollPane2.java

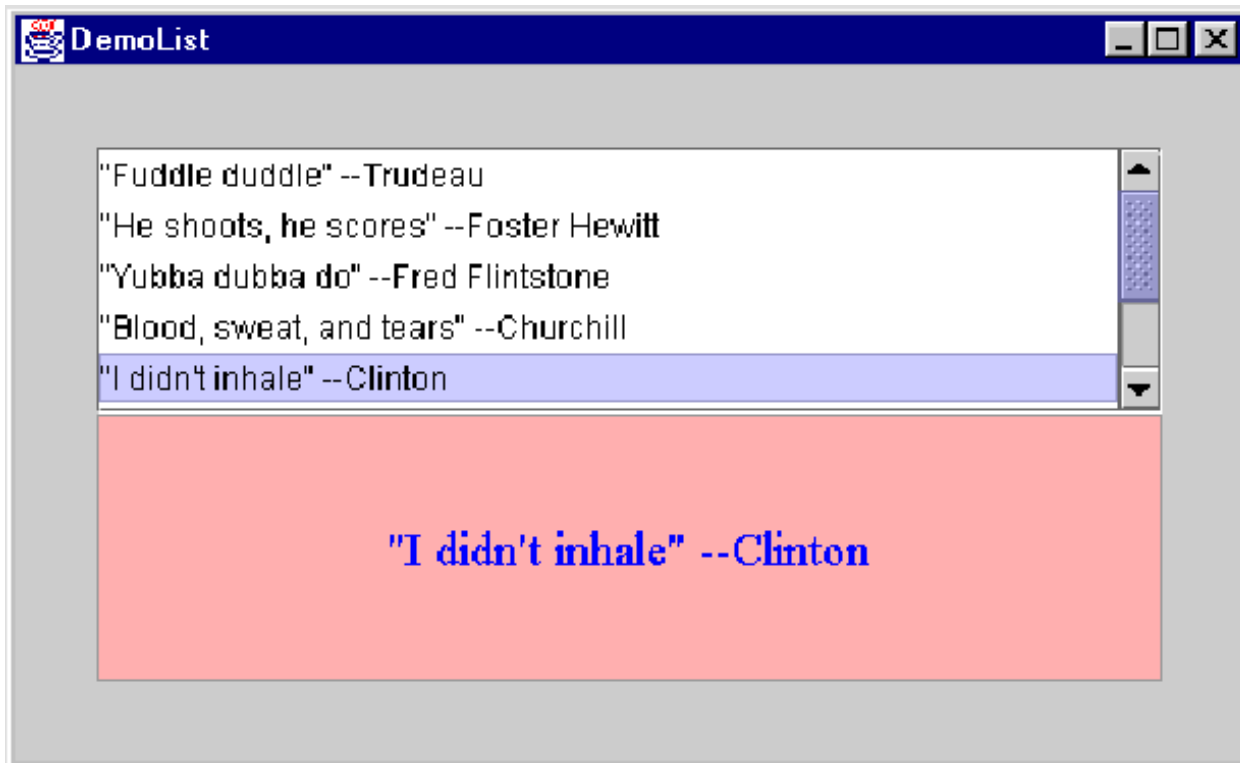


# More Widgets

- JList
- JTable
- JTabbedPane

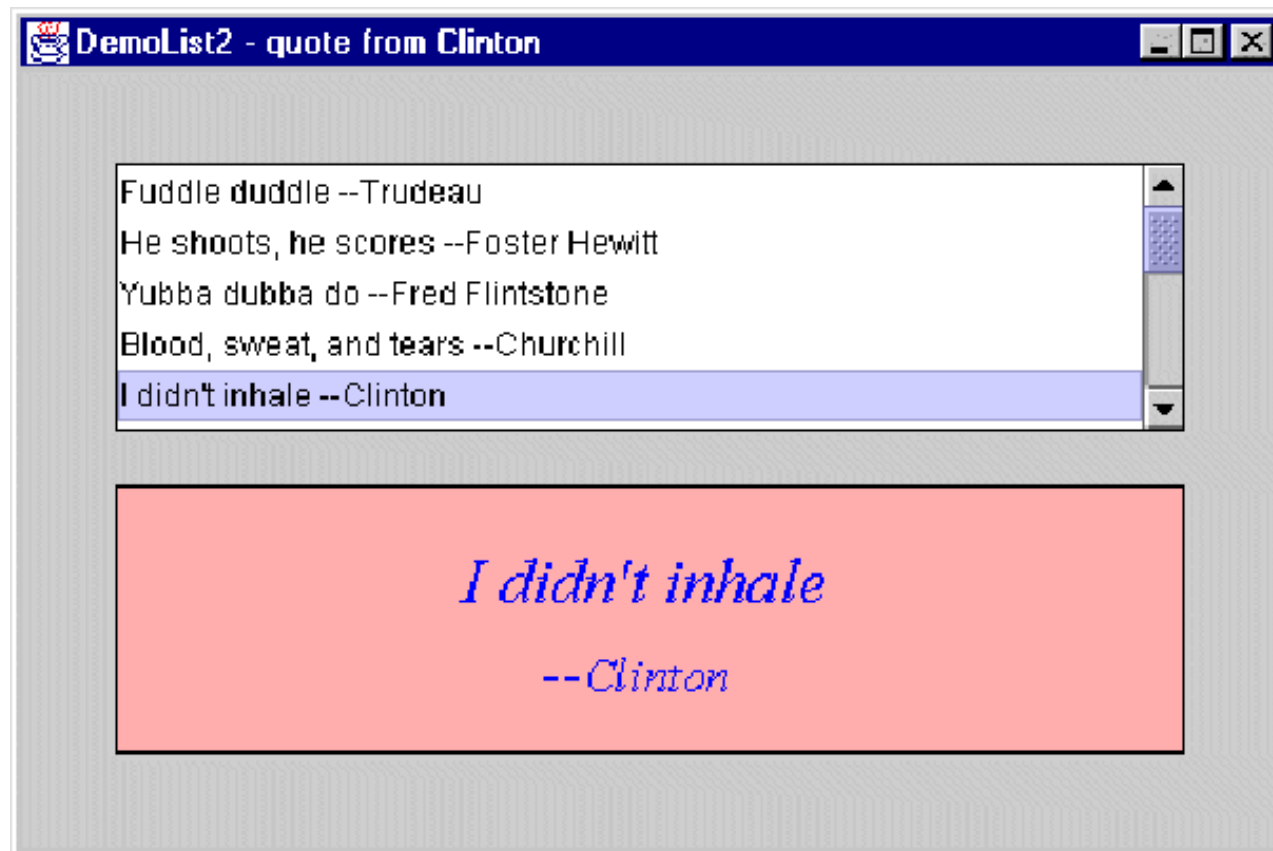
# Example Program

## DemoList.java



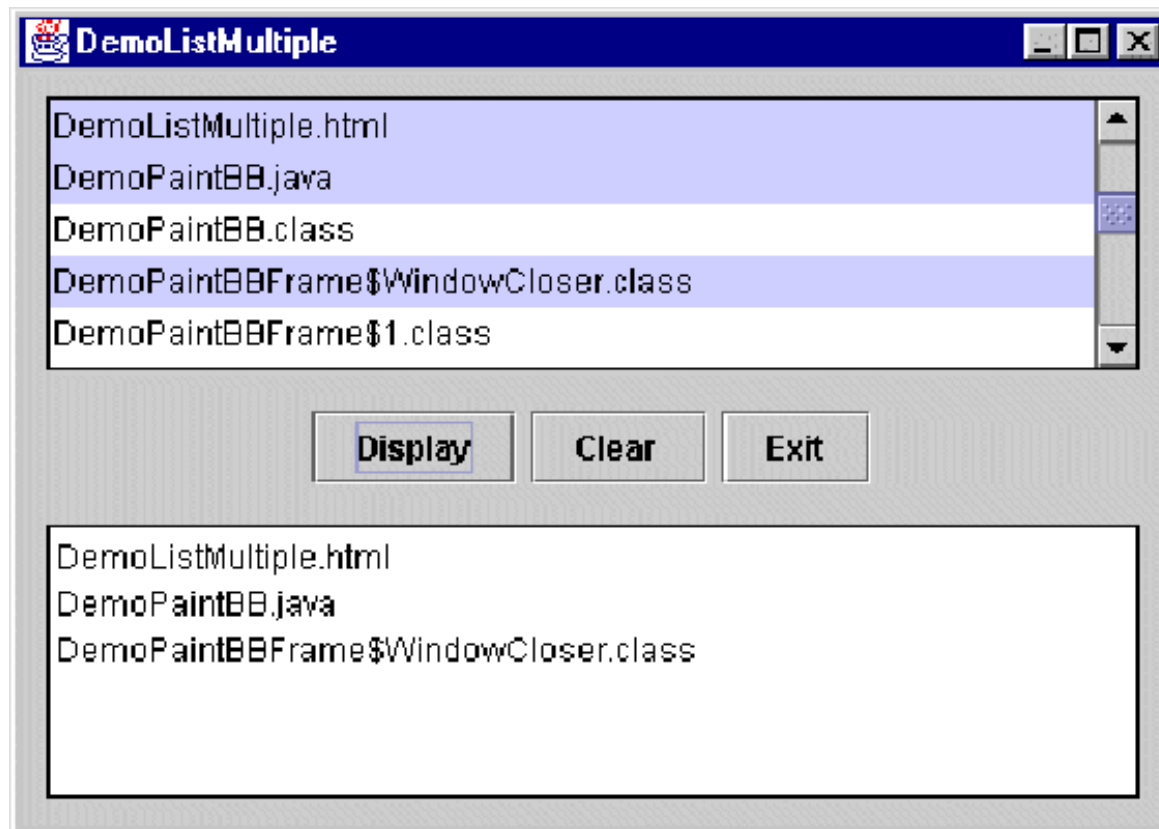
# Example Program

## DemoList2.java



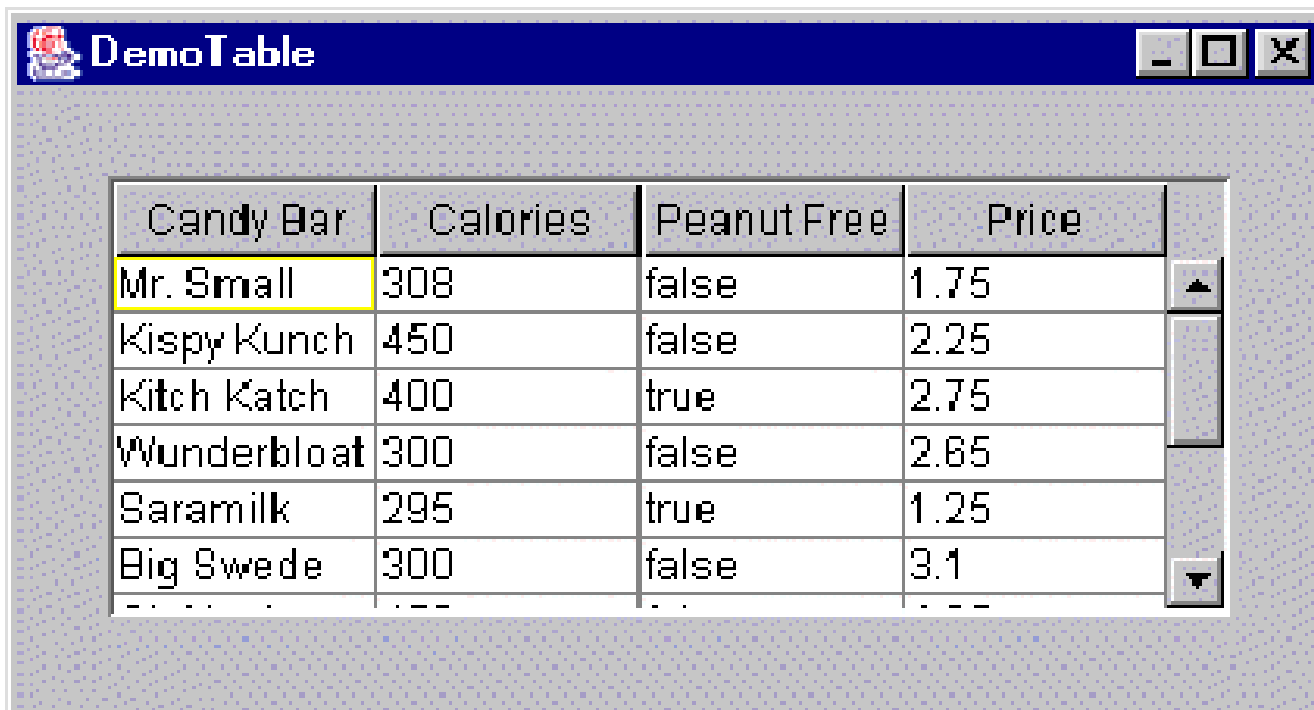
# Example Program

## DemoListMultiple.java



# Example Program

## DemoTable.java



The image shows a screenshot of a Java Swing window titled "DemoTable". The window contains a table with the following data:

Candy Bar	Calories	Peanut Free	Price
Mr. Small	308	false	1.75
Kispy Kunch	450	false	2.25
Kitch Katch	400	true	2.75
Wunderbloat	300	false	2.65
Saramilk	295	true	1.25
Big Swede	300	false	3.1

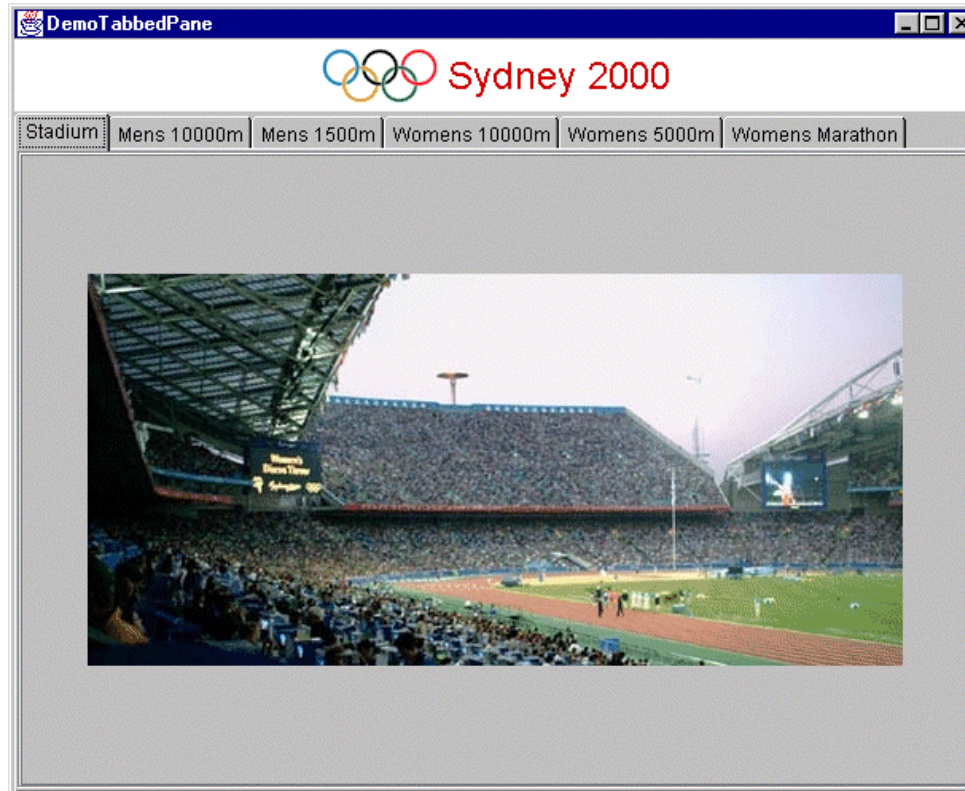
The table is displayed within a window with a blue title bar and standard window controls (minimize, maximize, close). The "Mr. Small" row is highlighted with a yellow border. A vertical scrollbar is visible on the right side of the table.

# Programming Challenge

Modify `DemoTable` to read the table's data from a separate text file, rather than hard-coding the data in the source file. Allow the new application to create tables of various formats; for example, with as many rows and columns as necessary. The data file should be specified as a command-line argument. Two other optional command-line arguments seem worthwhile: one to specify the delimiter character(s), and one to indicate whether or not the first row of data contains column names. Name the program `PC_Table.java`.

# Example Program

## DemoTabbedPane.java



Next Topic...