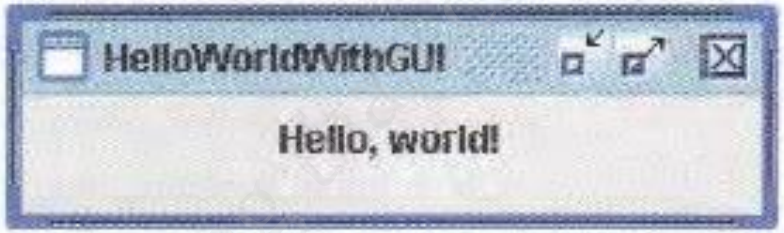
GUIs and Event-Driven Programming

Ho Dac Hung

What is a GUI?

- A GUI is a graphical user interface.



What is a GUI?

- Not only does a GUI use components such as frames, buttons, text fields to communicate with the user, but GUIs are event-driven. An event-driven application executed code in response to events. A GUI responds to an event by executing a method called an event handler.

The Swing Package

- The Swing API is part of the Java Foundation Classes and contains numerous components for creating GUIs.
- A frame is a top-level container for a GUI, which holds and displays all the other components of an interface in a content frame.

The Swing Package

- setDefaultLookAndFeelDecorated(boolean)
- setDefaultCloseOperation(class constant)
- getContentPane()
- setContentPane(Container contentPane)
- pack()
- setVisible(boolean)

The Swing Package

- A Jframe object uses a content pane to hold GUI components, A Jpanel object is one choice for a simple content pane.

  add(Component GUIcomponent)

  remove(int index)

The Swing Package

- The swing package include the JLabel class for creating labels that can be added to a content pane.

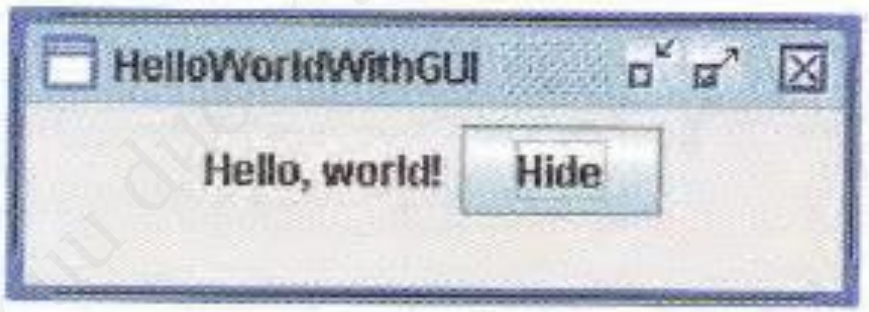
  JLabel(String str)

  JLabel(String str, align constant)

  setText(String str)

The JButton Class

- A button is commonly used GUI component. A button can be clicked by the user to communicate with the application.



The JButton Class

- JButton(String str)
- setActionCommand(String cmd)
- getActionCommand()
- addActionListener(Object)

Handling Events

- Swing components use listeners to determine if an event has occurred. A listener is an object that listen for action events. When an event is heard, the listener responds by executing an event handler named actionPerformed().
- The actionPerformed() method has an ActionEvent parameter passed by the GUI when an event occurs. The ActionEnevt parameter passed by the GUI when an event occurs.

Controlling Layout

- Layout refers to the arrangement of components, In a Swing GUI, the layout of a content pane can be controlled by adding border, using a layout manager, and setting alignments.
- A border can be added to most components, including the content pane. An invisible, or empty, border can be used to add “padding” around a component.

Layout Manager

- A layout manager determines the order of components on a content pane. There are many layout managers to choose including FlowLayout, BoxLayout, GridLayoud,…

The FlowLayout Manager

- The FlowLayout manager places components one next to the other in a row. When a row becomes too lone, a new row is started. The FlowLayout manager is the default manager.

The BoxLayout Manager

- The BoxLayout manager places components one after the other in a column, with one component per line.

The GridLayout Manager

- The GridLayout manager places components into a grid of rows and columns. The intersection of a row and column is called a cell. There is one component per cell in a GridLayout.

Alignment

- Another factor that affects layout is alignment. Alignment refers to the placement of a component within a layout.

Alignment

- setLayout()
- setBorder()
- setAlignment()

Getting Input from the User

- A text field allows a user to enter information at runtime. Text fields are usually placed next to a label to prompt the user for the type of data expected in the text field.

Text Field

- JTextField(int col)
- JTextField(String text, int col)
- getText()
- addActionListener(Object)

Combo Box

- A combo box offers a user a way to select from a limited set of choices. Combo boxes can offer choices without taking up much room on the interface. The user simply clicks the combo box arrow to display additional choices.

Combo Box

- JComboBox(Object[] items)
- setSelectedIndex(int index)
- getSelectedItem()
- setEditable(boolean)
- addActionListener(Object)

Changing Colors

- Swing components have methods for changing their colors. The java.awt package includes the Color class with numerous coloe constant members. The following methods, in the Swing component classes, can be used to change the background and foreground colors of the components:

  setBackground(color constant)

  setForeground(color constant)

Adding Images

- Image can make an application move informative, easier to to use, and fun. Labels and buttons are often used to display an image, but many Swing components suppoer images.

  JLabel(ImageIcon pic)

  JButton(String str, Image Icon pic)

  JButton(ImageIcon pic)

  setIcon(ImageIcon pic)