
MONASH University
MONASH
INFORMATION
TECHNOLOGY
FIT5192 Lecture 3: GUI Application and Event Handling
GROUP OF EIGHT AUSTRALIA

MONASH University

This Lecture

- Graphical User Interface (GUI) in Java
- How to layout the GUI of an application
- Using components in Swing library to build a GUI
- Threading for Swing application
- Event handling

GROUP OF EIGHT AUSTRALIA

Graphical User Interfaces

What is GUI?

- Provides a graphical way to operation a software
- Replaces the traditional command line interface application
- It offers a wide range of visual items such as:
  - – Windows
  - – Buttons
  - – Text boxes

GUI in Java

- Abstract Windows Toolkits & Swing library
- Abstract Windows Toolkits (AWT)
  - The look and behaviors of components are platform dependent
  - Lost its popularity
- Swing library
  - Built on top of AWT

Code Generation VS Manual Coding

- Generated by IDE
  - Fast, can create a graphical user interface within a very short period of time
  - Code can be messy
  - IDE dependent
  - Often more difficult to maintain
- Written by programmer
  - Slow, take more time to develop
  - Code is clear and usually more readable
  - Easier to maintain

Top-Level Container

- Three types of top-level containers:
  - JFrame
  - JDialog
  - JApplet

Content Pane

- Every top level container has a content pane
- A container that holds all the visible components

Layout of Application

- Absolute positioning
  - Specify the X and Y coordinates for each components
  - Not recommended
- Layout Manager
  - Divides a frame into sections
  - Java offers many Layout Manager to fit different needs
  - Focus on:
    - FlowLayout
    - BoxLayout
    - BorderLayout
    - GridLayout

FlowLayout

Default layout

- Display components horizontally, from left to right, according to the order they are added

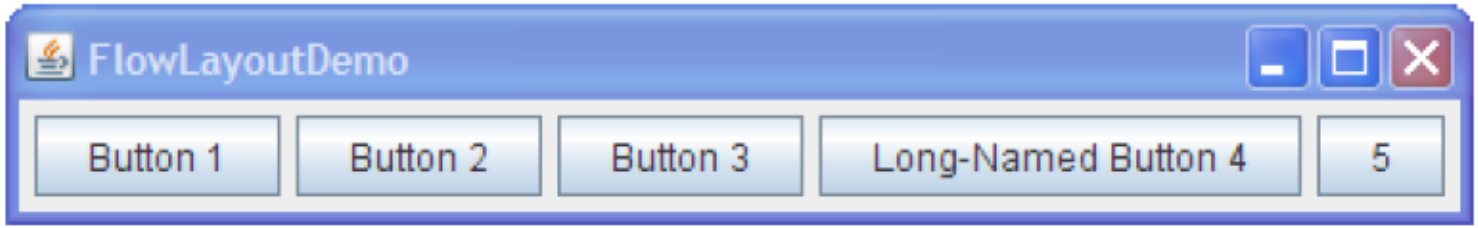


Image source: http://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html

BoxLayout

- Display components vertically, from top to bottom, according to the order they are added

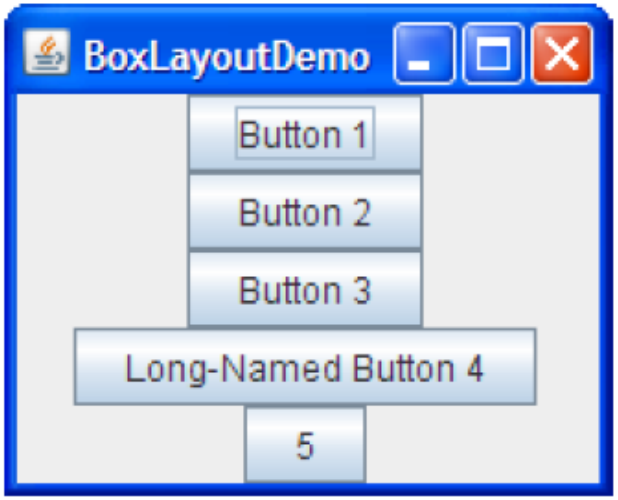


Image source: http://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html

BorderLayout

- Divide a container into 5 sections, including top, bottom, center, right and left

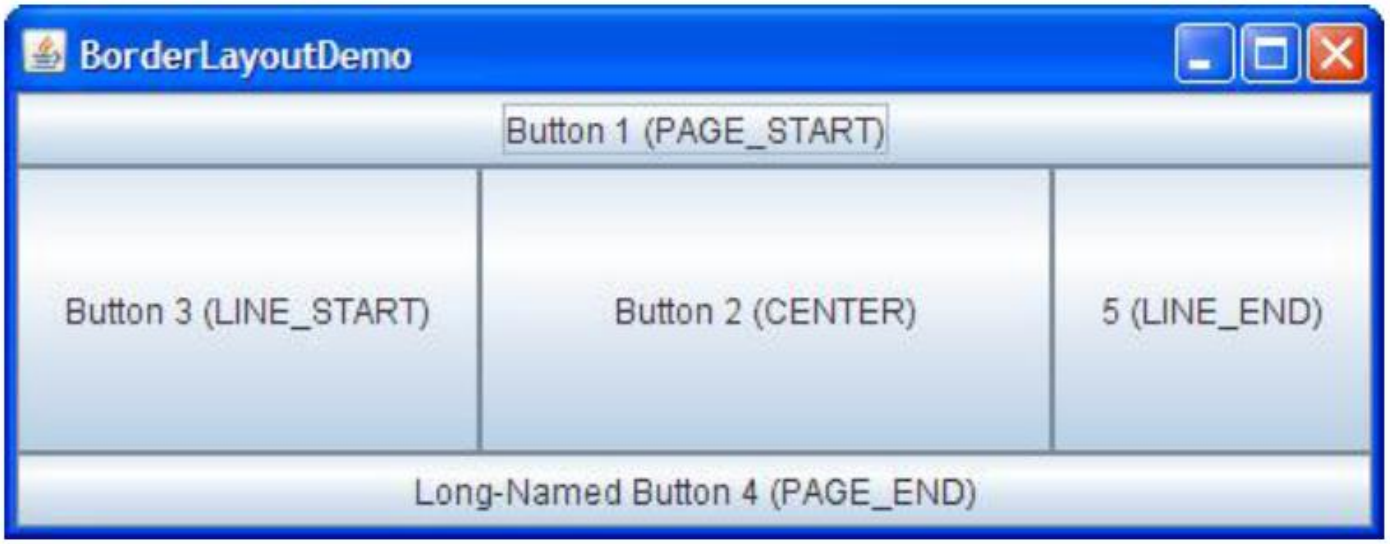


Image source: http://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html

GridLayout

- Divide a containers into a grid (i.e. columns and rows)

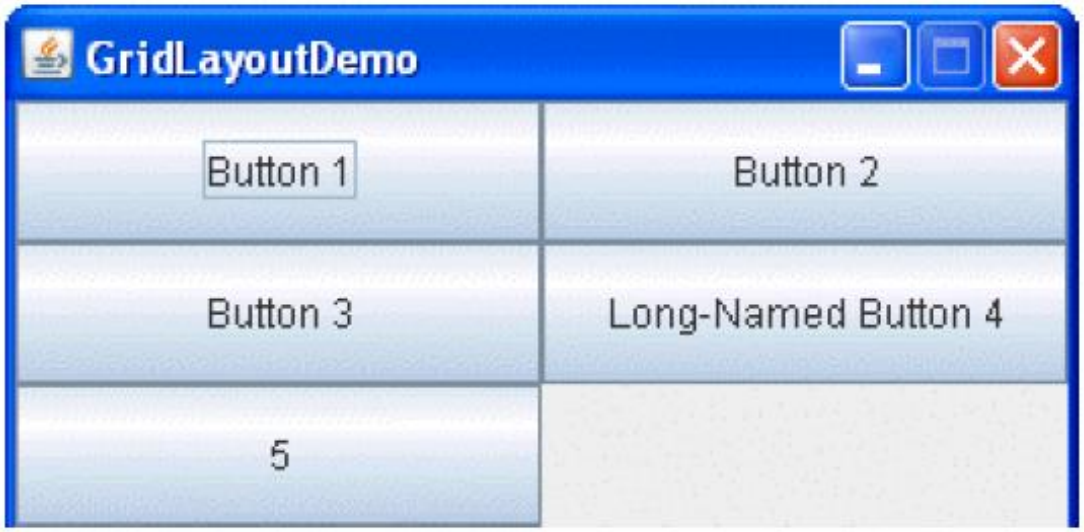


Image source: http://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html

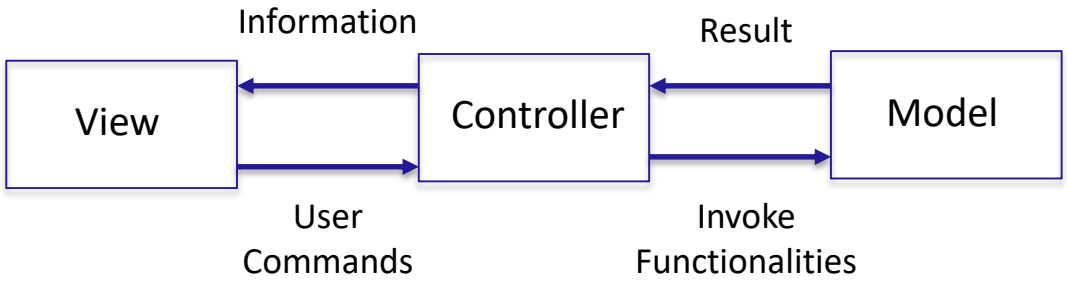
Components in Swing

60+ components

- All starts with ‘J’
- We will focus on 13 of the most commonly used ones, including:
  - JPanel
  - JTextField
  - JTextArea
  - JLabel
  - JButton
  - JCheckBox
  - JRadioButton
  - JComboBox
  - JMenuBar
  - JTable
  - JScrollPane
  - JTabbedPane
  - JDialogPane

Structure of Our Program

- Classes should be separated according to their responsibilities.
- Model-View-Controller (MVC)
- Model: the business logic / repository of the program
- View: accepts user command and displays data to user
- Controller: acts upon commands. Retrieve data or the result of an operation from user and pass that to the view for display.



Threading in Swing Application (1)

- A thread
  - is a light-weighted process
  - allows a computer program to perform multiple tasks concurrently
- Swing application has 3 types of threads:
  - Initial Thread: starts the application
  - Event Dispatch Thread (EDT): handles events. Any code that interacts with Swing library MUST be run on this thread
  - Worker Thread: the thread that time-consuming tasks run on.

Threading in Swing Application (2)

- Threading is beyond the scope of this unit. For our purpose, we only need to ensure that any code that interacts with GUI is run on EDT.
- All code that handles user actions is automatically run on EDT.
- We only need to make sure that the creation of GUI is done on EDT.
- To achieve that, we use SwingUtilities class.

Event Handling

- In Java, an event refers to an action that a user performs on a GUI, such as clicking a button or selecting an option from a combo box.
- An even listener is a class whose object can register to a particular event and will be notified when the event takes place.
- Upon receiving a notification, the registered listener object takes a designated action.
- An event source may be “listening by” many listener objects and a listener object may listen to many different event sources.

Event Listener

- To enable objects of a class to "listen" to a particular event:
  - The class will need to implement the listener interface of the event
  - Provides implementation of the method(s) defined in the interface
- Register the object to "listen" to the event of a particular component

Summary

- Graphical User Interface (GUI) in Java
- How to layout the GUI of an application
- Using components in Swing library to build a GUI
- Threading for Swing application
- Event handling

MONASH University

See you in the Studio !

Readings

- ***Recommended reading -*** *Chapter 3: Bean Validation, in Beginning Java EE 7, Antonio Goncalves*