Software Practice 1 - GUI

- What is GUI?
- Getting started with Swing
- Examples
- Lab Practice

Prof. Joonwon Lee
T.A. Sujin Oh
T.A. JunSeong Lee
What is GUI?

- **Graphical User Interface**
- **Definition**
  - Type of user interface that allows user to interact with electronic devices through graphical icons and visual indicators.
GUI development tool in Java

- **java.awt.**
  - Generate UI with native libraries
  - Results are slightly different for each OS

- **javax.swing.**
  - Generate UI with JVM support
  - Enable to make consistent interface for any OS
  - Have to be added before implementation
Terminology of Java GUI

- **Component**
  - Basic support for handling event, changing component sizes, controlling fronts, drawing components, etc
  - Ex) Button, TextField, TextArea, List, ...

- **Container**
  - A component that can accommodate other components
  - Ex) Frame, Window, Panel, Dialog, Applet
    - Frame, Window, Applet can be generated by itself
Examples of Component

- JButton
- JCheckBox
- JRadioButton
- JSlider
- JTextField
- JPasswordField
- JSeparator
- JTextArea
- JComboBox
- JList
Examples of Component

JProgressBar

JMenu

JToolTip

JScrollPane

JDialog

JApplet

JFrame
Examples of Component

<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Favorite Color</th>
<th>Favorite Movie</th>
<th>Favorite Number</th>
<th>Favorite Food</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>Albers</td>
<td>Green</td>
<td>Brazil</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>Mark</td>
<td>Andrews</td>
<td>Blue</td>
<td>Curse of the Dem...</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Brian</td>
<td>Beck</td>
<td>Black</td>
<td>The Blues Brothers</td>
<td>2,718</td>
<td></td>
</tr>
<tr>
<td>Lara</td>
<td>Bunni</td>
<td>Red</td>
<td>Airplane (the who...</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Roger</td>
<td>Brinkley</td>
<td>Blue</td>
<td>The Man Who Kn...</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>Brent</td>
<td>Christian</td>
<td>Black</td>
<td>Blade Runner (Dir...</td>
<td>23</td>
<td></td>
</tr>
</tbody>
</table>

\textit{JTable}

\textit{JTree}

\textit{JEditorPane and JTextPane}
Examples of Component

JToolBar

JTabbedPane

JSplitPane
Sample of Swing
Hierarchy of GUI Containers

Diagram showing the class hierarchy:
- Object
  - Component
    - LayoutManager
    - Container
    - Window
      - JFrame
      - JDialog
    - Panel
    - JComponent
      - Applet
Hierarchy of GUI Components

Diagram showing the inheritance hierarchy of Java GUI components, starting from `Object`, through `Component`, `Container`, and ending with various specific components like `JPanel`, `JLabel`, `JFileChooser`, `JTextField`, etc. The diagram also includes a relationship with a `LayoutManager`.
Getting started with Swing

1. Make frame
2. Add components to frame
3. Implement main() method

- Importing libraries
  - import java.awt.*
  - import java.awt.event.*
  - import javax.swing.*
  - import javax.swing.event.*
import javax.swing.*;

public class MyFrame extends JFrame {
    MyFrame () {
        setTitle ("First Frame");
        setSize (300, 300); // pixel
        setVisible (true);
    }
}
Different ways to configure UI

```java
public class MyFrame extends JFrame {
    MyFrame () {
        setTitle ("First Frame");
        setSize (300, 300); // pixel
        setVisible (true);
    }
}
```

```java
public class MyFrame extends JFrame {
    MyFrame () {
        setSize (300, 300); // pixel
        setVisible (true);
    }
}
```

```java
public class Main {
    public static void main (String[] args) {
        MyFrame mf = new MyFrame ();
        mf.setTitle ("First Frame");
    }
}
```

Highly recommend to use the first one!
Do not directly modify the configuration in the outside of the class
Components into frame

- Set new container into frame
  1. Generate container
  2. Set into frame

- Add new component into container
  1. Generate component
  2. Find container (e.g. content pane)
  3. Add component into container
Set new container

```java
public class MyFrame extends JFrame {
    public MyFrame () {
        JPanel p = new JPanel ();
        getContentPane ().setContentPane (p);
    }
}
```

Tip! This feature is usually used when switching between two existing panels.
public class MyFrame extends JFrame {
    public MyFrame () {
        JButton b = new JButton ();
        Container c = getContentPane ();
        c.add (b);
    }
}

Add new component
Implement main () method

```java
public class Main {
    public static void main (String[] args) {
        MyFrame mf = new MyFrame ();
    }
}
```
Insert image via JLabel

- `javax.swing.ImageIcon`
  - An implementation of the icon interface that paints from images
  - Images are created from a URL, filename or byte array which is preloaded using MediaTracker to minor the loaded state of the image
Example of inserting image

```java
public class MyFrame extends JFrame {
    public MyFrame () {
        ...
        ImageIcon icon = new ImageIcon ("path to image file");
        JLabel label1 = new JLabel ("Image and text", icon);
        JLabel label2 = new JLabel (icon);
        ...
    }
}

public class Main {
    public static void main () {
        MyFrame mf = new MyFrame ();
    }
}
```
Locating components

- Each container has unique locating manager called as LayoutManager which manages the location and the size of the components
- When a user manually resizes the container, LayoutManager intelligently rearrange and resize the all components in that container
- Implemented in java.awt package!
Famous LayoutManagers

- **FlowLayout**
  - Default LayoutManager for javax.swing.JPanel
  - Simply lays out components in a single row
  - Starts a new row if its container is not sufficiently wide
Famous LayoutManagers

- **BorderLayout**
  - Every content pane is initialized to use this
  - Places components in up to five areas;
    - top, bottom, left, right, center
  - Tool bars that are created using JToolBar must be created within this
Famous Layout Managers

- GridLayout
  - Simply makes a bunch of components equal in size
  - Displays components in the requested number of rows and columns
Famous LayoutManagers

- CardLayout
  - Enables to implement an area that contains different components in different times
  - Is often controlled by a combo box, with the state of the combo box determining which panel should be displayed
Container and LayoutManager

- **Default LayoutManager of each container**
  - BorderLayout: JWindow, JFrame, JDialog
  - FlowLayout: JPanel, JApplet
- **Set new LayoutManager into container**
  - Containers.setLayout (LayoutManager lm)

```java
JPanel p = new JPanel ();
p.setLayout (new BorderLayout ());

JFrame f = new JFrame ();
Container c = f.setContentPane ();
c.setLayout (new FlowLayout ());
```
Container w/o LayoutManager

- Allowed to manage the layout manually
- Used when the input methods like mouse or keyboard decide the location of the components (e.g. game program)

- How to?

```java
JFrame f = new JFrame ();
Container c = f.getContentPane ();
c.setLayout (null);
```
Container w/o LayoutManager

- Able to locate components overlapped
- Have to manage the all components manually with three of following methods
  - Resize
    - Component.setSize (int width, int height)
  - Locate
    - Component.setLocation (int x, int y);
  - Both
    - Component.setBounds (int x, int y, int width, int height)
Example of w/o LayoutManager

```java
import javax.swing.*;
import java.awt.*;

public class MyFrame extends JFrame {
    MyFrame () {
        setTitle ("First Frame");
        setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
        setLayout (null);

        JLabel l = new JLabel ("Hey!");
        l.setBounds (130, 50, 200, 20);
        add (l);

        for (int i = 1; i < 10; i++) {
            JButton b = new JButton (Integer.toString (i));
            b.setBounds (i * 15, i * 15, 50, 20);
            add (b);
        }
        setSize (300, 200);
        setVisible (true);
    }
}

public class Main {
    public static void main (String[] args) {
        MyFrame f = new MyFrame ();
    }
}
```
Event and Listener

- Event
  - An action or occurrence recognized by software
  - May be handled by the software

- Listener
  - So called event handler, that receive the event notification(or command) from the source
  - When event occurs, listener method will be called
Listener classes of each event

- ActionListener
- ComponentListener
- ItemListener
- KeyListener
- MouseListener
- WindowListener
- AdjustmentListener
- ContainerListener
- MouseMotionListener
- FocusListener

https://www.tutorialspoint.com/swing/swing_event_classes.htm
https://www.tutorialspoint.com/swing/swing_event_listeners.htm
Example of event handling

```java
public class Main {
    public static void main (String[] args) {
        MyFrame mf = new MyFrame ();
        JButton b = new JButton ("button");
        JLabel l = new JLabel ("none");
        Container c = mf.getContentPane ();

        c.add (l);
        c.add (b);
    }
}
```
import javax.swing.*;

public class MyFrame extends JFrame {
    private JLabel label;
    private int cnt = 0;

    MyFrame () {
        setTitle ("First Frame");
        JButton b1 = new JButton ("incNum");
        JButton b2 = new JButton ("decNum");
        b1.setActionCommand ("inc");
        b1.addActionListener (new ButtonClickListener ());
        b2.setActionCommand ("dec");
        b2.addActionListener (new ButtonClickListener ());
        label = new JLabel ("0");

        Container c = mf.getContentPane ();
        c.add (label);
        c.add (b1);
        c.add (b2);

        setSize (300, 300); // pixel
        setVisible (true);
    }

    private class ButtonClickListener implements ActionListener {
        public void actionPerformed (ActionEvent e) {
            String command = e.getActionCommand ();
            if (command.equals ("inc"))
                label.setText (Integer.toString (++ cnt));
            else
                label.setText (Integer.toString (-- cnt));
        }
    }
}
Using GUI tool, WindowBuilder

- First, WindowBuilder should be installed in your eclipse
- Create/open java class with WindowBuilder Editor
- Let’s do it!
[Lab – Practice 1]

- Make circle!

- Frame configuration
  - Size: 1000 * 1000
  - Radius: 400
  - Title: studentID_practice_1
  - Each color is created by random.

EXAMPLE)
[Lab – Practice 2]

- Make timer!
- Frame configuration
  - Size: 500 * 150
  - Title: studentID_practice_2
  - There are 3 timer and each timer has different increment value.
  - Every second, increment +1(or +5, +7).

EXAMPLE)