GUI II

IS 313

2.11.03
Outline

- Homework 1 & Quiz 2
- Homework 2
- Layout conclusion
- JOptionPane
- Event handling
Homework #1 scores

- Actual scores
- w/o late penalty

Score (out of 15)

<table>
<thead>
<tr>
<th>Score</th>
<th># of students</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
</tr>
</tbody>
</table>
Quiz #2

Quiz #2 scores

# of students

Score (out of 20)

6 8 10 12 14 16 18
Swing Topics

- UI Components
  - except advanced components
- Containment hierarchy
- Layout management
- Event handling
Layout Manager

- Maintains topological relationship between components
- Follows a set of layout rules
- Uses constraints specified by the programmer
## Border Layout

<table>
<thead>
<tr>
<th>Button 1 (NORTH)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Button 3 (WEST)</td>
</tr>
<tr>
<td>2 (CENTER)</td>
</tr>
<tr>
<td>Button 5 (EAST)</td>
</tr>
<tr>
<td>Long-Named Button 4 (SOUTH)</td>
</tr>
</tbody>
</table>

Java Applet Window

<table>
<thead>
<tr>
<th>Button 1 (NORTH)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Button 3 (WEST)</td>
</tr>
<tr>
<td>2 (CENTER)</td>
</tr>
<tr>
<td>Button 5 (EAST)</td>
</tr>
<tr>
<td>Long-Named Button 4 (SOUTH)</td>
</tr>
</tbody>
</table>

Java Applet Window
Flow Layout
Grid Layout
GridBag Layout
GridBag Layout

- Most flexible layout
- Topology
  - Two-dimensional distribution
- Rules
  - Components = preferred size
  - Extra space allocated using constraints
- Constraints
  - Grid location, cell span, padding
  - Fill dimension
  - Weight
Box Layout
Box Layout

- Topology
  - One-dimensional stacking

- Rules
  - Components = preferred size
  - Extra space allocated using special purpose “filler components”

- Constraints
  - Alignment
  - Order of insertion
Card Layout

- Topology
  - Indexed stack

- Rules
  - Only “top-most” component displayed

- Constraints
  - Order of insertion
Example I: ListDialog
Example Layout II

Some simple message dialogs:
- OK (in the L&F's words)
- Yes/No (in the L&F's words)
- Yes/No (in the programmer's words)
- Yes/No/Cancel (in the programmer's words)

Click the "Show it!" button to bring up the selected dialog.
Example Layout III
Example Layout IV
Built-in Dialogs

- JOptionPane
  - static methods for creating dialogs
Event Handling
Event handling

- JComponent
  - SomeComponent
    - addActionListener(listener : ActionListener)
  - MyReactionClass
    - actionPerformed(event : ActionEvent)
  - List

- ActionListener
  - actionPerformed(event : ActionEvent)

- "implements"
  - Put reaction code here
Menu item example

JMenu fileMenu = new JMenu ("File");
JMenuItem exitItem = new JMenuItem ("Exit");
exitItem.addActionListener (new ActionListener (){
    new ActionListener (){
        public void actionPerformed (ActionEvent e) {
            System.exit(0);
        }
    }
};
Button example

cancelButton.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        ListDialog.dialog.setVisible(false);
    }
});
Getting a value back

```java
button.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        String selectedName =
            ListDialog.showDialog(null, name.getText());
        name.setText(selectedName);
    }
});
```
MouseListener example

```
blankArea.addMouseListener(new MouseAdapter ()
{
    public void mouseClicked(MouseEvent e)
    {
        MouseEventDemo2.this.saySomething(
            "Mouse clicked (# of clicks: " + 
            e.getClickCount() + ")", e);
    }
});
```
Anonymous class

```java
new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        System.exit(0);
    }
}
```
Anonymous class diagram

```
 ActionListener
 <<implements>>
 Foo$1
```
Mouse example again

```java
blankArea.addMouseListener(new MouseAdapter ()
{
    public void mouseClicked(MouseEvent e)
    {
        MouseEventDemo2.this.saySomething("Mouse clicked (# of clicks: "+
        e.getClickCount() + ")", e);
    }
});
```
Access to External Local Variables

```java
class MyImagePanel {
  public void foo () {
    ...
  }
  ...
}

public void addImageLabel (String caption) {
  JLabel lbl = new JLabel (caption);
  final MyImage img = new MyImage (caption);
  lbl.addMouseListener (new MouseAdapter () {
    public void mouseClicked (MouseEvent e) {
      // need to access the outer class
      bar(MyImagePanel.this);
      // need to call method
      MyImagePanel.foo ();
    }
  });
  ...
}
```
Mouse example diagram

MouseAdapter

MyImagePanel$1

MyImage

img
public void addImageLabels (String [] captions)
{
    for (int i = 0; i < captions.length; i++)
    {
        JLabel lbl = new JLabel (caption);
        MyImage img = new MyImage (caption);
        lbl.addMouseListener( 
            new MouseAdapter ()
            {
                public void mouseClicked (MouseEvent e)
                {
                    // check for double-clicks
                    if (e.getClickCount() >= 2)
                        m_imageDisplayDialog.show(img);
                }
            });
        frm.add (lbl);
    }
}
public void addImageLabels (String [] captions)
{
    for (int i = 0; i < captions.length; i++)
    {
        JLabel lbl = new JLabel (caption);
        MyImage img = new MyImage (caption);
        createListener (lbl, img);
        frm.add (lbl);
    }
}

void createListener (JLabel lbl, MyImage img)
{
    final MyImage theImage = img;
    lbl.addMouseListener (new MouseAdapter ()
    {
        public void mouseClicked (MouseEvent e)
        {
            if (e.getClickCount() >= 2)
                m_imageDisplayDialog.show(img);
        }
    });
}
MouseListener interface

void mouseClicked(MouseEvent e)
void mouseEntered(MouseEvent e)
void mouseExited(MouseEvent e)
void mousePressed(MouseEvent e)
void mouseReleased(MouseEvent e)
Event handling strategies

- **Location of EH code**
  - external
  - component class
  - named inner class
  - anonymous inner class

- **Base class or interface used**
  - Implement XxxListener
  - Extend XxxAdapter
Recommendation

- Use anonymous inner class
  - unless EH is large, then used named inner class
- Extend Adapter
  - unless all interface methods needed
  - Note: no ActionAdapter class
Event handling thread

- Can only process one event at a time
- Time-consuming event handlers
  - tie up the interface
  - no other actions registered
Example