

CS1007: Object Oriented Design and Programming in Java

Lecture #14

Mar 7

Shlomo Hershkop
shlomo@cs.columbia.edu

Outline

- Custom Layout manager code
- More patterns
- Recognizing Patterns

- Reading: 5.4.2 - 5.8

Announcements

- I've been sick (it's not the exams), so delay on grading the exams and releasing the homework
- For the most part, very good work on the exams
- If something doesn't make sense in today's class, please ask (it might be me)

More announcements

- Have not released the Othello programming homework....we need to cover specific topics before you will be ready
- Might release another theory homework first to make sure you are up to reading/knowledge

Patterns

- We started to cover idea of patterns in the design of software
- Why patterns?
- Why these specific ones?

To note

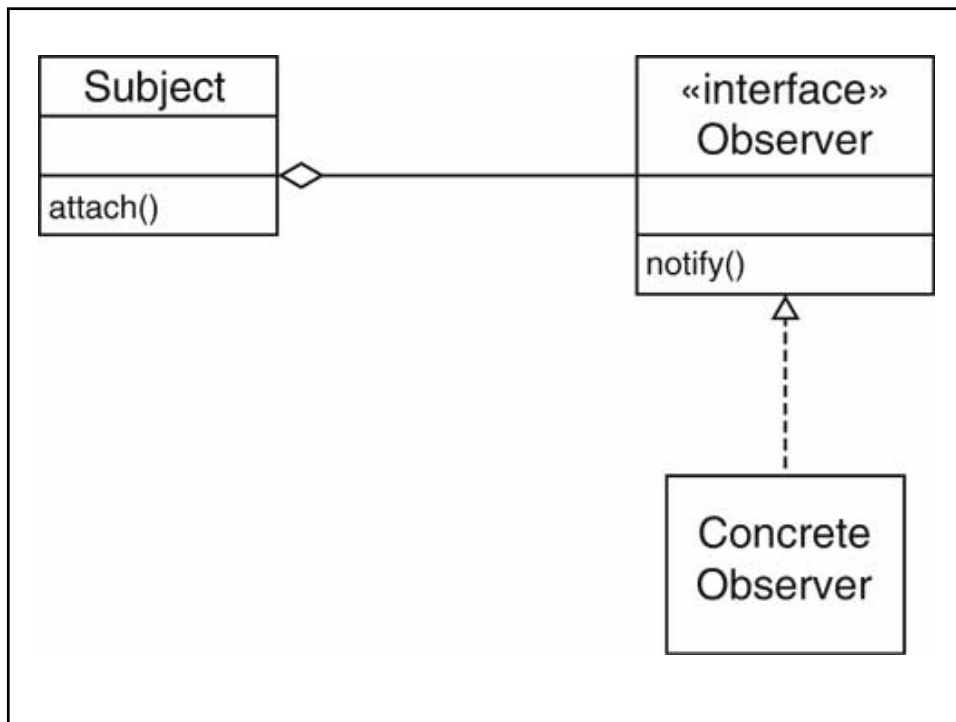
```
java.util.Iterator i;  
for (i = something.iterator();  
     i.hasNext(); ) {  
    Object obj = i.next();  
    // the following code processes  
    obj  
    ...  
}
```

From last time

1. An object, called the subject, is source of events
2. One or more observer objects want to be notified when such an event occurs.

One approach

- Define an observer interface type. All concrete observers implement it.
- The subject maintains a collection of observers.
- The subject supplies methods for attaching and detaching observers.
- Whenever an event occurs, the subject notifies all observers.



Layout Managers

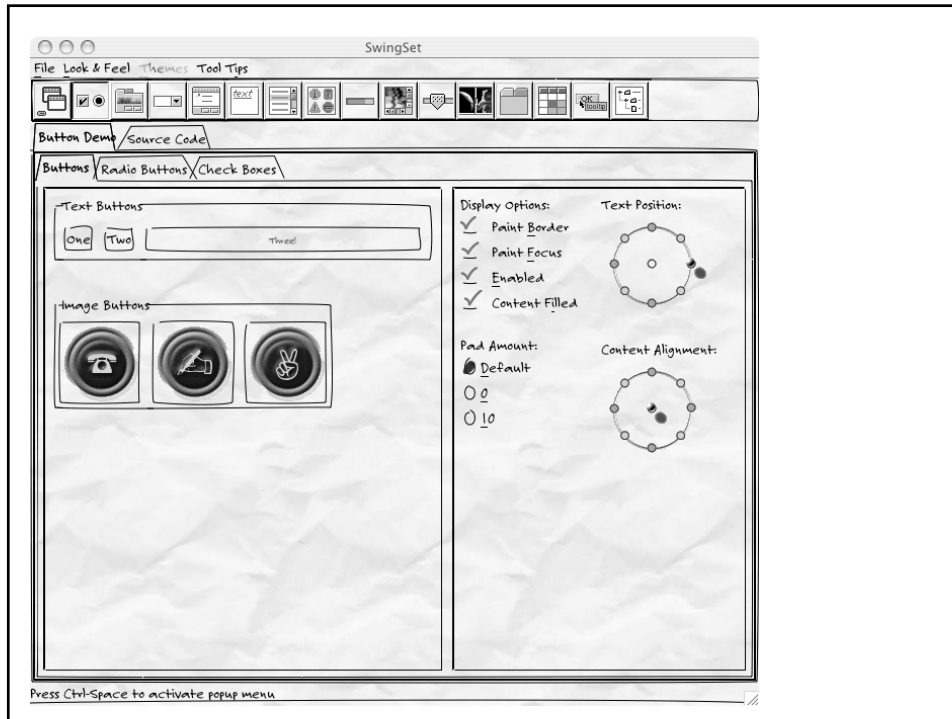
- We've covered the basics of layout managers

Look and Feel

- Libraries to define a standard look and behavior exactly the same across platforms

Example 1

- Windows look and feel
- Allows your GUI to look like a standard windows program
 - On windows
 - On linux



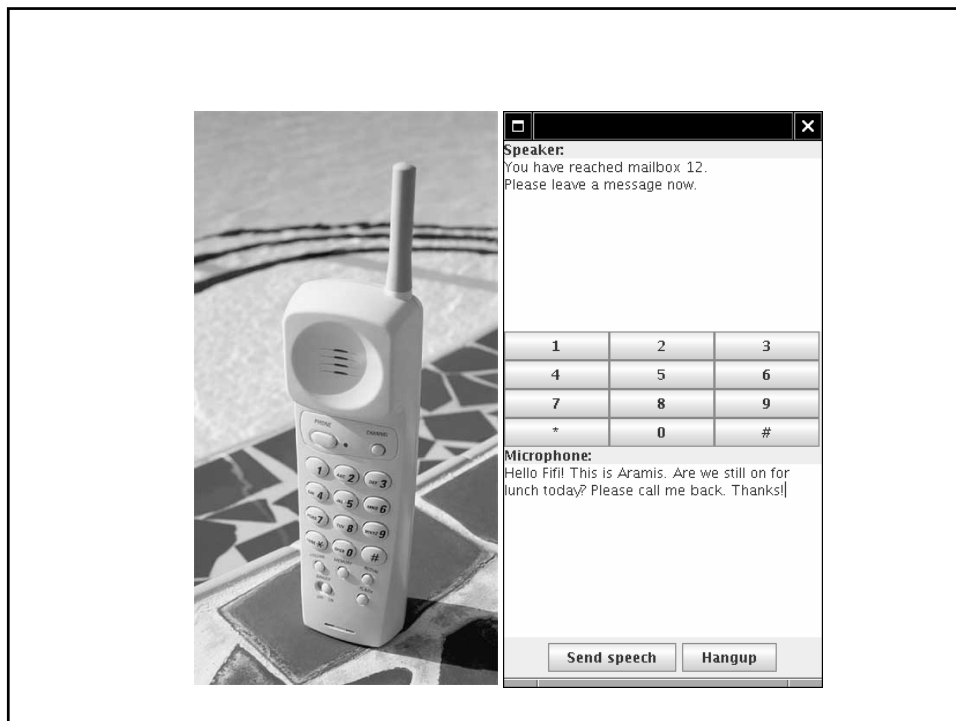
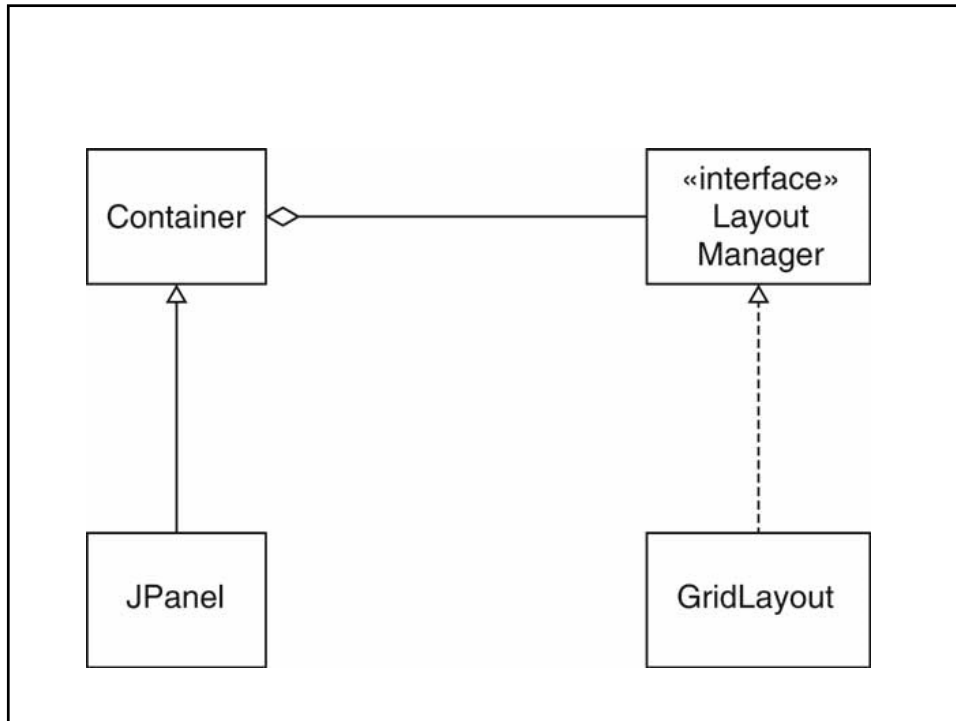
Layout Managers

- Set layout manager

```
JPanel keyPanel = new JPanel();  
keyPanel.setLayout(new GridLayout(4, 3));
```

- Add components

```
for (int i = 0; i < 12; i++)  
    keyPanel.add(button[i]);
```



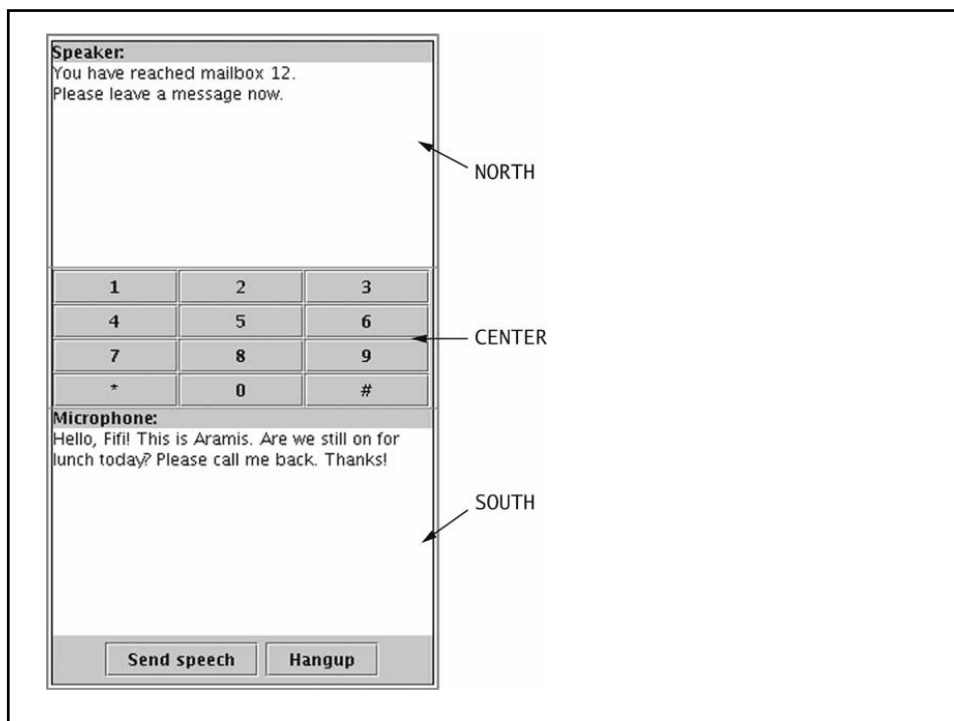
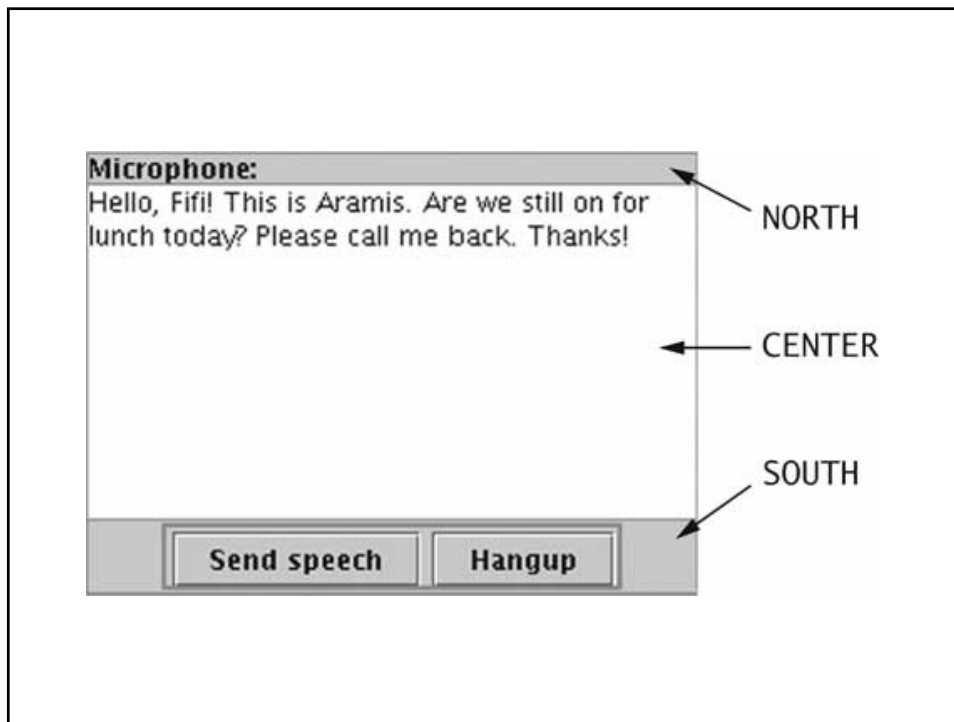
Keys

- Arrange keys in panel with GridLayout:

```
JPanel keyPanel = new JPanel();
keyPanel.setLayout(new GridLayout(4, 3));
for (int i = 0; i < 12; i++)
{
    JButton keyButton = new JButton(...);
    keyPanel.add(keyButton);
    keyButton.addActionListener(...);
}
```

- Panel with BorderLayout for speaker

```
JPanel speakerPanel = new JPanel();
speakerPanel.setLayout(new BorderLayout());
speakerPanel.add(new JLabel("Speaker:"),
    BorderLayout.NORTH);
speakerField = new JTextArea(10, 25);
speakerPanel.add(speakerField,
    BorderLayout.CENTER);
```



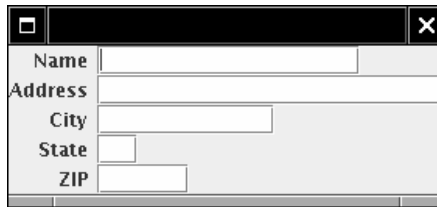
- [Ch5/code/mailgui/Telephone.java.html](#)

Design layout manager

- All layout managers implement the same interface
- First we need to identify what our layout manager goals are.

Custom Layouts

- Form layout
- Odd-numbered components right aligned
- Even-numbered components left aligned
- Implement `LayoutManager` interface type



LayoutManager Interface

```
public interface LayoutManager
{
    void layoutContainer(Container parent);
    Dimension minimumLayoutSize(Container parent);
    Dimension preferredLayoutSize(Container parent);
    void addLayoutComponent(String name, Component
    comp);
    void removeLayoutComponent(Component comp);
}
```

Form Layout

- Ch5/layout/FormLayout.java
- Ch5/layout/FormLayoutTester.java
- Note: Can use GridBagLayout to achieve the same effect

Next Time

- Do reading

- Reading chapter 5