

CSE 8B—Intro to CS: Java

Winter, 2006

February 16 (Day 12)
Swing

Instructor: Neil Rhodes

Overview

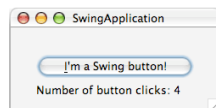
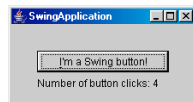
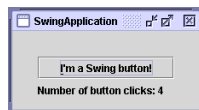
Java Foundation Classes (JFC) contain:

- Swing GUI framework for creating
 - Java applications
 - Java frameworks
- Pluggable look and feel
 - Your Java application can have Java look and feel or native look and feel
- Accessibility API
 - For screen readers, Braille displays, etc.
- Java 2D API
 - Two-dimensional graphics
- Drag-and-drop support
 - Between Java apps and native apps
- Internationalization
 - Input methods for lots of different languages

2

Look and feel

Same application can run with different look and feel



3

Simplest Swing application

```
import javax.swing.*;  
  
public class Minimal {  
    public static void main(String[] args){  
        JFrame win = new JFrame("Minimal");  
        win.setVisible(true);  
    }  
}
```

4

setVisible

Frames start out *not* visible

- So that various information can be set in the window before it is displayed.
- If window were displayed immediately:
 - There would be flicker and resizing as the contents of the window were added/updated

5

Swing application that exits on close box

```
import javax.swing.*;

public class Minimal2 {
    public static void main(String[] args) {
        JFrame win = new JFrame("Minimal2");
        win.setDefaultCloseOperation(
            JFrame.EXIT_ON_CLOSE);
        win.setVisible(true);
    }
}
```

6

What happens on close box?

Four possibilities:

- EXIT_ON_CLOSE
 - When the window's close box is tapped, the application exits
- HIDE_ON_CLOSE
 - When the window's close box is tapped, the window is hidden (but still exists; can be redisplayed later).
- DISPOSE_ON_CLOSE *The default*
 - When the window's close box is tapped, the window is closed and disposed (it would need to be re-instantiated to be displayed again).
- DO_NOTHING_ON_CLOSE
 - When the window's close box is tapped, nothing happens (except that your program is notified of the tap).

7

Commonly, create a JFrame subclass for each window

```
import javax.swing.*;

public class Minimal3 extends JFrame {
    public static void main(String[] args) {
        Minimal3 window = new Minimal3();
        window.setVisible(true);
    }

    Minimal3() {
        super("Minimal3");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

8

Specifying the size of a window

```
import javax.swing.*;

public class Minimal4 extends JFrame {
    public static void main(String[] args) {
        Minimal4 window = new Minimal4();
        window.setVisible(true);    }

    Minimal4() {
        super("Minimal4");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 100);
    }
}
```

9

Contents of a frame

Can add:

- Buttons
- Text fields
- Scrollbars
- Sliders
- Labels
- Checkboxes
- Radio buttons
- ...

10

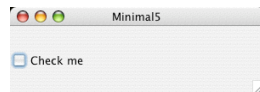
Window with a checkbox

```
import javax.swing.*;

public class Minimal5 extends JFrame {
    public static void main(String[] args) {
        Minimal5 window = new Minimal5();
        window.setVisible(true);
    }

    Minimal5() {
        super("Minimal5");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 100);

        JCheckBox checkbox = new JCheckBox("Check me");
        add(checkbox);
    }
}
```



11

Adding multiple components

Layout manager

- Controls location of components added
- Different kinds of layout managers
 - **FlowLayout**
 - Lays out components left to right
 - **BorderLayout**
 - Lays out components in specified location
 - BorderLayout.NORTH
 - BorderLayout.SOUTH
 - BorderLayout.EAST
 - BorderLayout.WEST
 - BorderLayout.CENTER
 - **GridLayout**
 - Number of rows and columns specified in constructor
 - Specify 0 for rows or 0 for columns
 - If both are non-zero, columns are ignored
 - Lays out components along first row, then second, and so on

12

Multiple checkboxes using FlowLayout

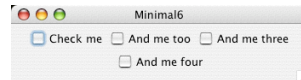
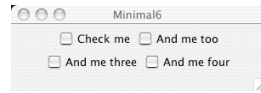
```
import javax.swing.*;
import java.awt.*;

public class Minimal6 extends JFrame {
    public static void main(String[] args) {
        Minimal6 window = new Minimal6();
        window.setVisible(true);
    }

    Minimal6() {
        super("Minimal6");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 100);

        setLayout(new FlowLayout());

        add(new JCheckBox("Check me"));
        add(new JCheckBox("And me too"));
        add(new JCheckBox("And me three"));
        add(new JCheckBox("And me four"));
    }
}
```



13

Multiple checkboxes using BorderLayout

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

public class Minimal7 extends JFrame {
    public static void main(String[] args) {
        Minimal7 window = new Minimal7();
        window.setVisible(true);
    }

    Minimal7() {
        super("Minimal7");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 100);

        setLayout(new BorderLayout());

        add(new JCheckBox("Check me"), BorderLayout.NORTH);
        add(new JCheckBox("And me too"), BorderLayout.SOUTH);
        add(new JCheckBox("And me three"), BorderLayout.WEST);
        add(new JCheckBox("And me four"), BorderLayout.EAST);
        add(new JCheckBox("And me five"), BorderLayout.CENTER);
    }
}
```



14

Multiple checkboxes using GridLayout (2 rows)

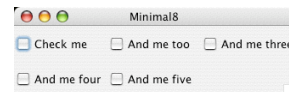
```
import javax.swing.*;
import java.awt.*;

public class Minimal8 extends JFrame {
    public static void main(String[] args) {
        Minimal8 window = new Minimal8();
        window.setVisible(true);
    }

    Minimal8() {
        super("Minimal8");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 100);

        setLayout(new GridLayout(2, 0));

        add(new JCheckBox("Check me"));
        add(new JCheckBox("And me too"));
        add(new JCheckBox("And me three"));
        add(new JCheckBox("And me four"));
        add(new JCheckBox("And me five"));
    }
}
```



15

Multiple checkboxes using GridLayout (2 columns)

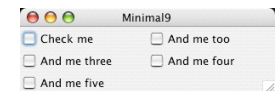
```
import javax.swing.*;
import java.awt.*;

public class Minimal9 extends JFrame {
    public static void main(String[] args) {
        Minimal9 window = new Minimal9();
        window.setVisible(true);
    }

    Minimal9() {
        super("Minimal9");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 100);

        setLayout(new GridLayout(0, 2));

        add(new JCheckBox("Check me"));
        add(new JCheckBox("And me too"));
        add(new JCheckBox("And me three"));
        add(new JCheckBox("And me four"));
        add(new JCheckBox("And me five"));
    }
}
```



16

Hierarchy of containers

The Panel class is a container

- It can have its own layout manager
- It can contain other panels

17

Layout using panels

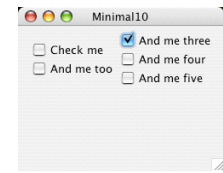
```
public class Minimal10 extends JFrame {
    public static void main(String[] args) {
        Minimal10 window = new Minimal10();
        window.setVisible(true);
    }

    Minimal10() {
        super("Minimal10");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 100);

        setLayout(new FlowLayout());

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(0, 1));
        add(panel);
        panel.add(new JCheckBox("Check me"));
        panel.add(new JCheckBox("And me too"));

        JPanel panel2 = new JPanel();
        add(panel2);
        panel2.setLayout(new GridLayout(0, 1));
        panel2.add(new JCheckBox("And me three"));
        panel2.add(new JCheckBox("And me four"));
        panel2.add(new JCheckBox("And me five"));
    }
}
```



18

Radio buttons

JRadioButton

If you add JRadioButtons to a ButtonGroup, the button group will ensure:

- no more than one button is selected at a time

It's up to you to select the first button

19

Borders

Any component can have a border

- normally used with JPanel and JLabel
- create a border using the BorderFactory
 - Common examples:
 - TitledBorder
 - LineBorder
- Add it using setBorder

20

Sample using radio buttons and borders

```
Minimal11() {
    ...

    JPanel panel = new JPanel();
    panel.setBorder(BorderFactory.createTitledBorder("My title"));
    panel.setLayout(new GridLayout(0, 1));
    add(panel);
    panel.add(new JCheckBox("Check me"));
    panel.add(new JCheckBox("And me too"));

    JPanel panel2 = new JPanel();
    panel2.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    add(panel2);
    panel2.setLayout(new GridLayout(0, 1));
    ButtonGroup group = new ButtonGroup();
    JRadioButton button = new JRadioButton("choose this");
    panel2.add(button);
    group.add(button);
    button = new JRadioButton("or this");
    panel2.add(button);
    group.add(button);
    button = new JRadioButton("or this one");
    panel2.add(button);
    group.add(button);
    button.setSelected(true);
}
```



21

Other components

Labels

- JLabel

Text

- JTextField (one line)
 - Constructor specifies minimum number of characters to be visible
- JTextArea (multiple lines)
 - Constructor specifies minimum number of lines and minimum number of characters to be visible

Button

- JButton

22

Example with text, button, and label

```
Minimal12() {
    super("Minimal12");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setSize(300, 120);

    setLayout(new FlowLayout());

    JPanel panel = new JPanel();
    panel.setBorder(BorderFactory.createTitledBorder("My title"));
    panel.setLayout(new GridLayout(0, 3));
    add(panel);
    panel.add(new JButton("Press me"));
    panel.add(new JTextField(10));

    JPanel panel2 = new JPanel();
    panel2.setLayout(new GridLayout(2, 0));
    panel2.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    add(panel2);
    panel2.add(new JLabel("Enter your text here"));
    panel2.add(new JTextArea(5, 40));

    pack(); // resize window to fit contents
}
```



23