DM550 / DM857
Introduction to Programming

Peter Schneider-Kamp

petersk@imada.sdu.dk

http://imada.sdu.dk/~petersk/DM550/

http://imada.sdu.dk/~petersk/DM857/

GRAPHICAL USER INTERFACES

HelloWorld Reloaded

- Java standard GUI package is Swing

- from popup message to professional user interface

- Example:

```java
import javax.swing.*;
public class HelloWorldSimple {
    public static void main(String[] args) {
        JOptionPane.showMessageDialog(null, "Hello World!");
    }
}
```

- more challenging to do anything more complicated

- multi-threaded event-driven model-based UI design :-o

UNIVERSITY OF SOUTHERN DENMARK.DK

Dialogs

- user dialogs are created using JDialog class

- basically like JFrame (next slide), but with a parent window

- often used via static JOptionPane methods

- Example:

```
Object[] options = {1, 2, 3, 4, 5, 10, 23, 42};
Object result = JOptionPane.showInputDialog(null,
    "Select number", "Input",
    JOptionPane.INFORMATION_MESSAGE, null,
    options, options[0]);
int selectedInt = (Integer) result;
```

UNIVERSITY OF SOUTHERN DENMARK.DK

Creating a Window

- windows are represented by objects of class JFrame
- constructor gets title displayed at top of window
- Example:

```
JFrame window = new JFrame("My first window!");

window.setSize(400, 250);      // set size of window to 400x250
window.setLocation(50, 50);   // top-left corner at (50, 50)

// exit program when window is closed
window.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

window.setVisible(true);        // show window on the screen
```

UNIVERSITY OF SOUTHERN DENMARK.DK

Creating Content

- content is placed in objects of class JPanel
- on these we can either
  - draw directly on it using the paintComponent method
  - add ready-made components using the add method
- every window has a JPanel as its main "content pane"
- Example 1 (draw directly):

```
public class MyPanel extends JPanel {
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.drawString("My first panel!", 100, 100);
    }
}
```

Creating Content

- content is placed in objects of class JPanel
- on these we can either
    - draw directly on it using the paintComponent method
    - add ready-made components using the add method
- every window has a JPanel as its main "content pane"
- Example 2 (add a button):

```
JButton button = new JButton("My first button!");
button.addActionListener(new ButtonHandler());
JPanel panel = new JPanel();
panel.add(button);
window.setContentPane(panel);
window.pack();
```

Listeners and Events

- events = changes in the user interface
- mouse movement, key pressed, button clicked, …

- listeners = objects that respond to events
- Example (ActionListener for button from previous slide):

```java
import java.awt.*;
import java.awt.event.*;
public class ButtonHandler implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
}
```

Mouse Events

- interface MouseListener for mouse events
- needs to be added using addMouseListener methods
- often component class implementing the interface itself
- Example (panel that changes color during click):

```
public class Clicky extends JPanel implements MouseListener {
    public Clicky() {  this.addMouseListener(this);  }
    public void mousePressed(MouseEvent event) {
        this.setBackground(Color.RED);
    }
    public void mouseReleased(MouseEvent evt) {
        this.setBackground(Color.GRAY);
    } … }
```

Anonymous Event Handlers

- interface MouseListener for mouse events
- needs to be added using addMouseListener methods
- alternatively use anonymous class
- Example (panel that changes color during click):

```
public class Clicky extends JPanel {
    public Clicky() {  this.addMouseListener(new MouseAdapter() {
        public void mousePressed(MouseEvent event) {
            this.setBackground(Color.RED);
        }
        public void mouseReleased(MouseEvent evt) {
            this.setBackground(Color.GRAY);
        }   });   }  …s  }
```

Colors

- colors are represented by objects of class Color
- define by RGB values or use pre-defined constants
- Example:

```
import java.awt.*;

…

JPanel panel = new JPanel(new BorderLayout());
JPanel panelA = new JPanel();
panelA.setBackground(new Color(192, 64, 128)); // strange color
JPanel panelB = new JPanel();
panelB.setBackground(Color.RED));
panel.add(panelA, BorderLayout.NORTH);
panel.add(panelB, BorderLayout.SOUTH);
```

Labels

- simple component to display strings or images
- labels are objects of class JLabel
- text, colors, fonts etc. can be changed during runtime
- Example:

JLabel label = new JLabel("My first label!", JLabel.CENTER);

…

label.setText("something more interesting");

label.setForeground(Color.BLUE);

label.setBackground(Color.YELLOW);

label.setOpaque(true);   // background filled

label.setFont(new Font("Serif", Font.ITALIC, 15));

Fonts

- fonts represented by objects of class Font

- constructor takes name, style, and point size

- see Java API documentation for more examples

- Example:

import java.awt.*;

…

Font font = new Font("Arial", Font.BOLD, 42);

JButton button = new JButton("Click me!");

button.setFont(font);

…

Borders

- borders are represented by objects of class Border
- borders can be added to any component
- typically created using static methods in BorderFactory
- Example:

```
JPanel panel = new JPanel(new GridLayout(3,3));
for (int i = 0; i < 9; i++) {
    JPanel subPanel = new JPanel();
    subPanel.setBorder(BorderFactory.createLineBorder(
        Color.BLACK));
    panel.add(subPanel);
}
```

UNIVERSITY OF SOUTHERN DENMARK.DK

Panel Layout

- layout   =   spatial organization of components
- components can be either
    - organized by absolute coordinates
    - organized by an object of class LayoutManager
- Example 1 (layout with BorderLayout):

JPanel panel = new JPanel(new BorderLayout());

panel.add(new JButton("North"), BorderLayout.NORTH);

panel.add(new JButton("Center"), BorderLayout.CENTER);

panel.add(new JButton("West"), BorderLayout.WEST);

panel.add(new JButton("South"), BorderLayout.SOUTH);

panel.add(new JButton("East"), BorderLayout.EAST);

Panel Layout

- layout   =   spatial organization of components
- components can be either
    - organized by absolute coordinates
    - organized by an object of class LayoutManager
- Example 2 (layout with GridLayout):

JPanel panel = new JPanel(new GridLayout(2,3));

panel.add(new JButton("North"));

panel.add(new JButton("Center"));

panel.add(new JButton("West"));

panel.add(new JButton("South"));

panel.add(new JButton("East"));

Basic Components

- buttons represented by objects of class JButton
- Example (disabled button with text label):

JButton button = new JButton("Big, bad, and ugly!");

button.addActionListener(new MyButtonHandler());

button.setEnabled(false);

- check boxes represented by objects of class JCheckBox
- Example (initially selected two-state check box):

JCheckBox checkBox = new JCheckBox("more money!", true);

…

boolean wantsMore = checkBox.isSelected();

Basic Components

- selectable options represented by objects of class JComboBox
- Example (select from a list of numbers):

```
Object[] options = {1, 2, 3, 4, 5, 10, 23, 42};

JComboBox optionBox = new JComboBox(options);

optionBox.setSelected(6);

optionBox.addActionListener(new MySelectionHandler());

…

int selectedInt = (Integer) optionBox.getSelectedItem();
```

Basic Components

- selection on a range of values by objects of class JSlider

- Example (select percentage from 0 to 100, initally 50):

JSlider percent = new JSlider(0, 100, 50);

percent.setMajorTickSpacing(25);

percent.setMinorTickSpacing(5);

percent.setPaintTicks(true);

percent.setPaintLabels(true);

percent.addChangeListener(new MyChangeHandler());

Text Components

- text fields represented by objects of class JTextField
- Example (text field for email input):

JTextField email = new JTextField();

…

String userEmail = checkRFC5322(email.getText());

- text areas represented by objects of class JTextArea
- Example (full-window scrollable editable text entry area):

JTextArea entryArea = new JTextArea(5, 20);

textArea.setEditable(true);

JScrollPane scrollPane = new JScrollPane(textArea);

window.getContentPane().add(scrollPane);

Menus

- menus represented by JMenuBar, JMenu, and JMenuItem
- Example (menu bar with a single file menu with three items):

```java
JMenu file = new JMenu("File");          // create drop down menu
JMenuItem open = new JMenuItem("Open");
file.add(open);          open.addActionListener(this);
JMenuItem save = new JMenuItem("Save");
file.add(save);          save.addActionListener(this);
JMenuItem saveas = new JMenuItem("Save as ...");
file.add(saveas);        saveas.addActionListener(this);
JMenuBar menuBar = new JMenuBar();       // menu bar
menuBar.add(file);
```

Menus

- menus represented by JMenuBar, JMenu, and JMenuItem
- Example (menu bar with a single file menu with three items):

```
public class MyMenu implements ActionListener {

    public MyMenu() {

        …           // see previous slide

    }

    public void actionPerformed(ActionEvent e) {

        …           // check which menu item was clicked and react

    }

}
```

UNIVERSITY OF SOUTHERN **DENMARK**.DK