Introduction to the Java SWING API

CSCI 2300

Introduction

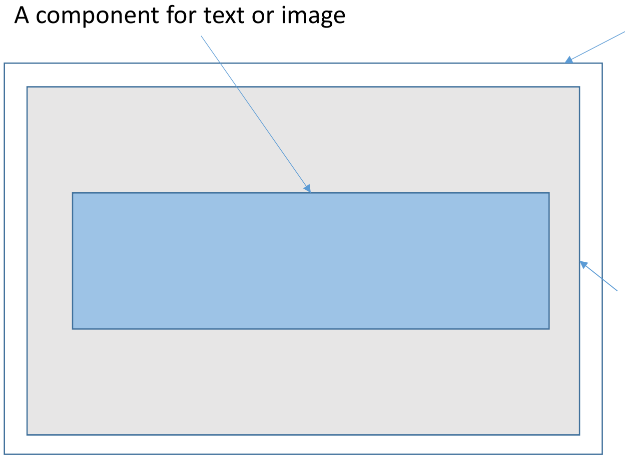
- Swing – A set of GUI classes
  - Part of the Java's standard library
  - Much better than the previous library: AWT
    - Abstract Window Toolkit

- Highlights
  - A rich set of widgets
    - Widget: Any GUI element (also called: components)
  - Contents and shape are separated (MVC support)
  - Fine-grained control over the behavior and look and feel
  - Platform independent
    - Isolates the programmer from the operating system's GUI

Swing Components

- Containers
  - Contain and manage other components.
  - Top Level/Internal
  - Examples: `JFrame` (Top Level), `JScrollPane`, `JPanel`.
- Basic controls
  - Atomic components
  - Used for showing ouput and/or getting some input
  - Inherits JComponent
  - Examples: `JButton, JLabel, JTextArea, JTable, Jlist`
- Usually every Swing class extends the corresponding AWT class
  - For backward-compatibility reasons

First SWING application



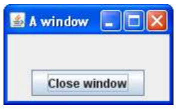
JFrame
Top-level container
Main window

JLabel
A component for text or image

JPanel
A container for other components

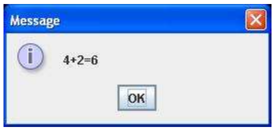
Top Level Containers: JFrame

- **`javax.swing.JFrame`**:
  - Top-level window with a title and a border.
  - Usually used as a program's main window



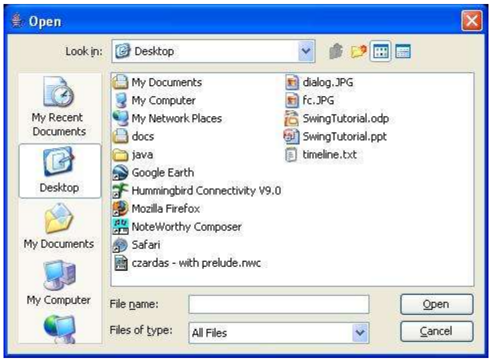
Top Level Containers: JDialog

- **`javax.swing.JDialog`**:
  - More simple and limited than frames
  - Typically used for showing a short message on the screen
  - Also has a border and a title bar
  - May have an owner
    - If the owner is invisible the dialog will also be invisible
  - Use the static method of JoptionPane to show standard dialog boxes:
    `JOptionPane.showMessageDialog(null, "4+2=6");`



4+2=6

Top Level Containers: JFileChooser



- **`javax.swing.JFileChooser`**:
  - Allows the the user to choose a file
  - Supports “open” and “save”: `showOpenDialog()`,`showSaveDialog()`

```
JFileChooser fc = new JFileChooser();
int returnVal = fc.showOpenDialog(null);
if(returnVal == JFileChooser.APPROVE_OPTION)
   System.out.println("File: " + fc.getSelectedFile());
```

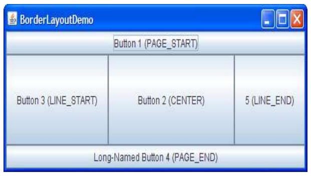
Containers - Layout

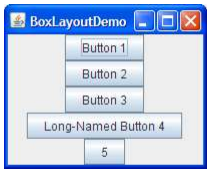
- Each container has a layout manager
  - Determines the size, location of contained widgets.

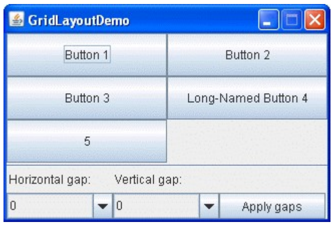
- Setting the current layout of a container:
  `void setLayout(LayoutManager lm)`

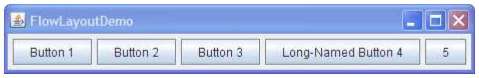
- *`LayoutManager`* implementing classes:
  - `BorderLayout`
  - `BoxLayout`
  - `FlowLayout`
  - `GridLayout`

Containers - Layout

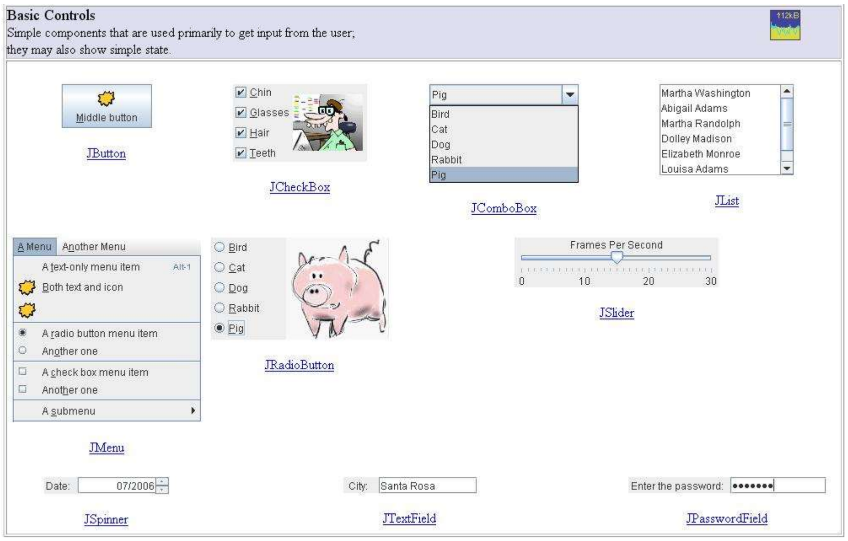








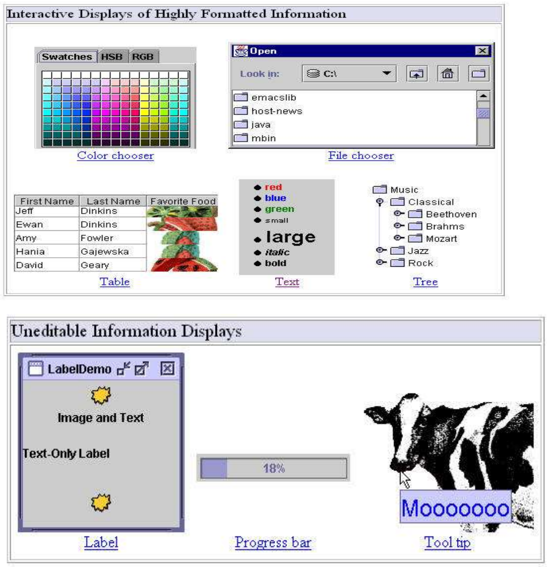
Swing Components



Basic Controls

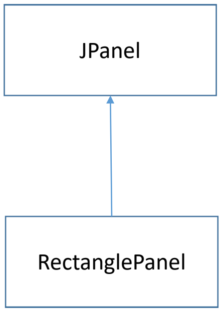
Simple components that are used primarily to get input from the user; they may also show simple state.

Swing Components

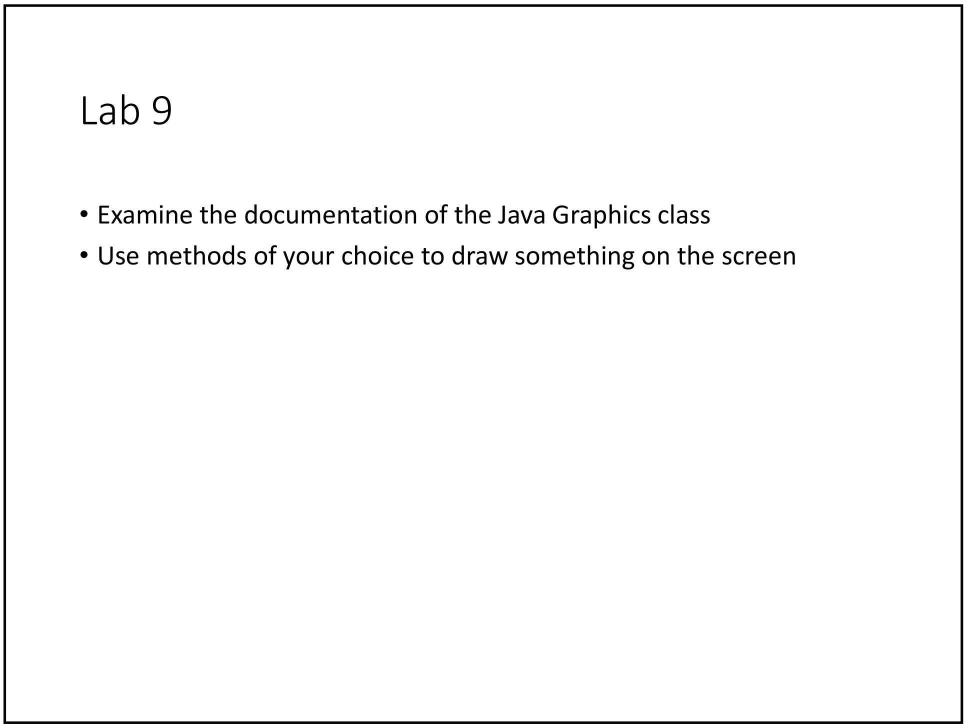


First Name	Last Name	Favorite Food
Jeff	Dinkins	
Ewan	Dinkins	
Amy	Fowler	
Hania	Gajewska	
David	Geary	

We can create our own components



```
public class RectanglePanel extends JPanel
{
    private int height;
    private int width;
    public RectanglePanel(int h, int w)
    {
      setPreferredSize(new Dimension(w, h));
      this.width = w;
      this.height = h;
    }
    @Override
    protected void paintComponent(Graphics g)
    {
      super.paintComponent(g);
      g.setColor(Color.BLUE);
      g.fillRect(0, 0, width, height);
    }
}
```



Lab 9

- Examine the documentation of the Java Graphics class
- Use methods of your choice to draw something on the screen