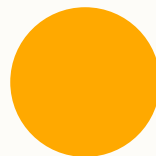# Intro to Java GUIs

Reese Hatfield

0

Review

- A good class is
    - Testable
    - Has Robust Exceptions
    - Encapsulated Behavior
    - etc.

Moving Forward

- That's cool I guess

- We have a lot of tools
  - But how do we build real applications?

Moving Forward

- Not like command line apps
- Real graphical applications

- CLI = Command Line Interface apps
- GUI = Graphical User Interfaces

Moving Forward

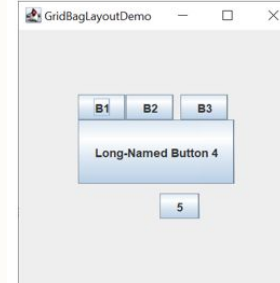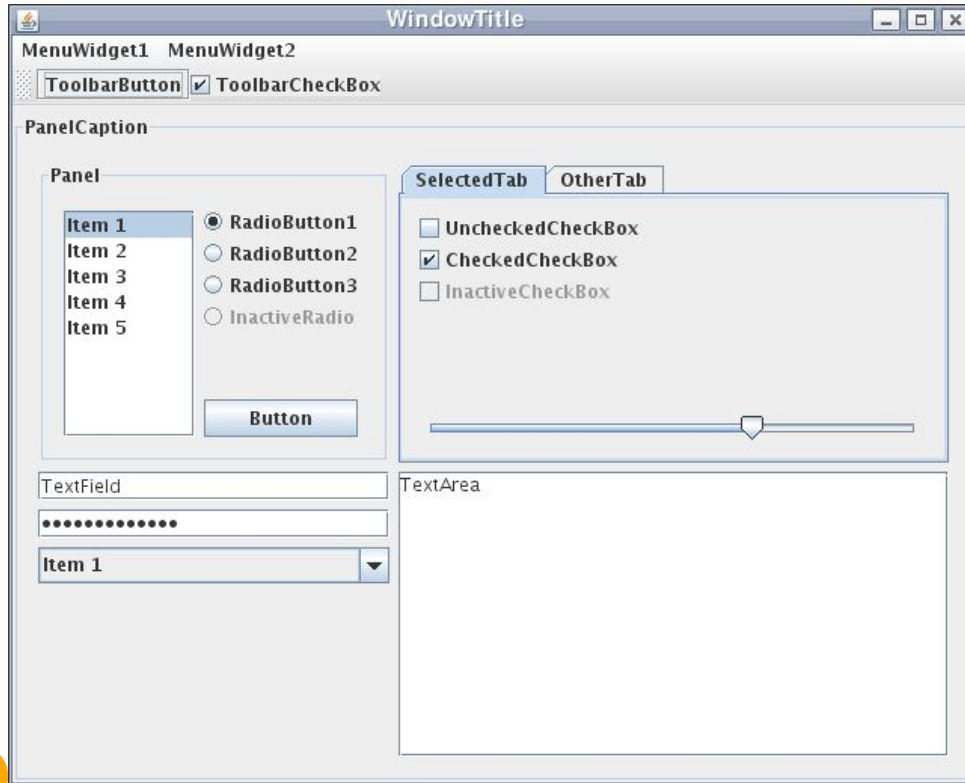- CLI's are usually developer/power-user facing
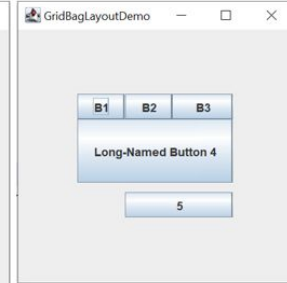
- GUI's are more user focused

GUIs

- Java's primary GUI library is *Swing*
- Built into java
- No external libraries needed
- Cross platform GUI framework

GUIs

WindowTitle
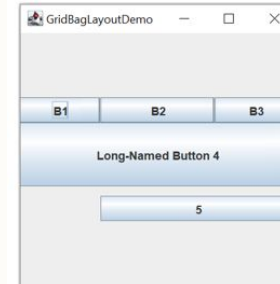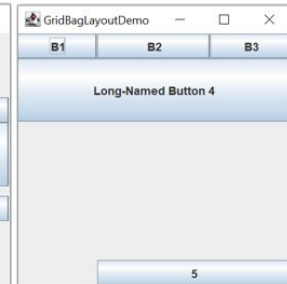
MenuWidget1   MenuWidget2

ToolbarButton ☑ ToolbarCheckBox

PanelCaption

Panel

Item 1
Item 2
Item 3
Item 4
Item 5

Button

● RadioButton1
○ RadioButton2
○ RadioButton3
○ InactiveRadio

SelectedTab   OtherTab

☐ UncheckedCheckBox
☑ CheckedCheckBox
☐ InactiveCheckBox

TextField
••••••••••••
Item 1

TextArea

GridBagLayoutDemo

B1 B2 B3

Long-Named Button 4

5

(a) Natural width and height

GridBagLayoutDemo

B1 B2 B3

Long-Named Button 4

5

(b) fill = HORIZONTAL

GridBagLayoutDemo

B1 B2 B3

Long-Named Button 4

5

(c)
Column 1: weightx = 0.5
Column 2: weightx = 1.0
Column 3: weightx = 0.5

GridBagLayoutDemo

B1 B2 B3

Long-Named Button 4

5

(d)
Row 1: weighty = 0 (not participating)
Row 2: weighty = 0 (not participating)
Row 3: weighty = 1.0 (take all)

JFrame

- A JFrame represents an individual window
- JFrame is a class that you can use yourself
- Let's do it!

JFrame

- JFrame frame = new JFrame()
  - Nothing happened
- Useful methods:
  - setVisible(boolean)
  - setSize(int, int)
  - setTitle(String)
  - setDefaultCloseOperation(int)

Let's add some more things

- Swing provides a series of pre-written components
- We can use these in our program to make our GUI
- All prefixed with "J"

JLabels

- Let's add some text
- JLabel class
- Nice constructor and setter

- Can we add more?

Let's add some more things

- What happened?
- 2nd label clobbered the 1st

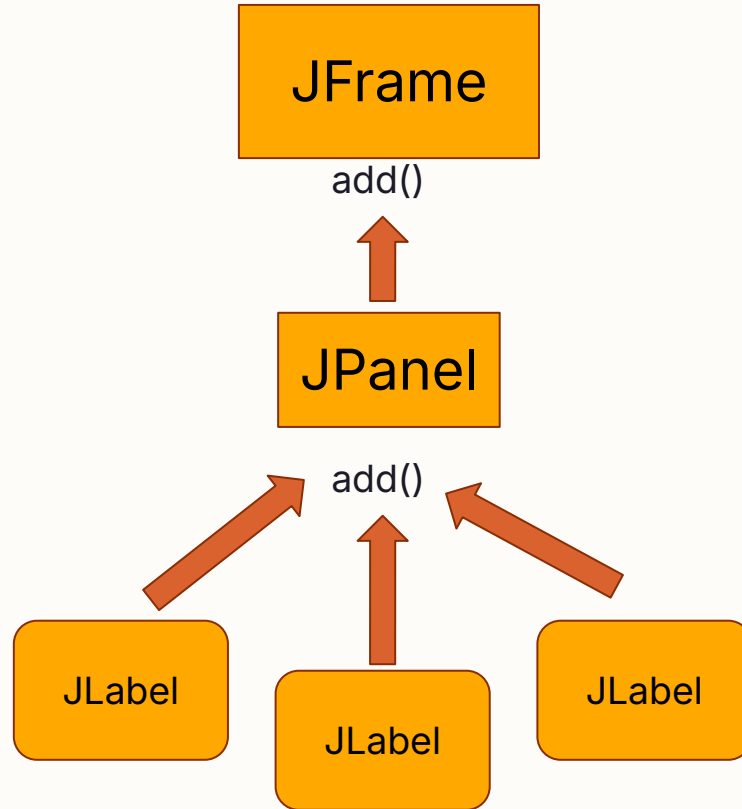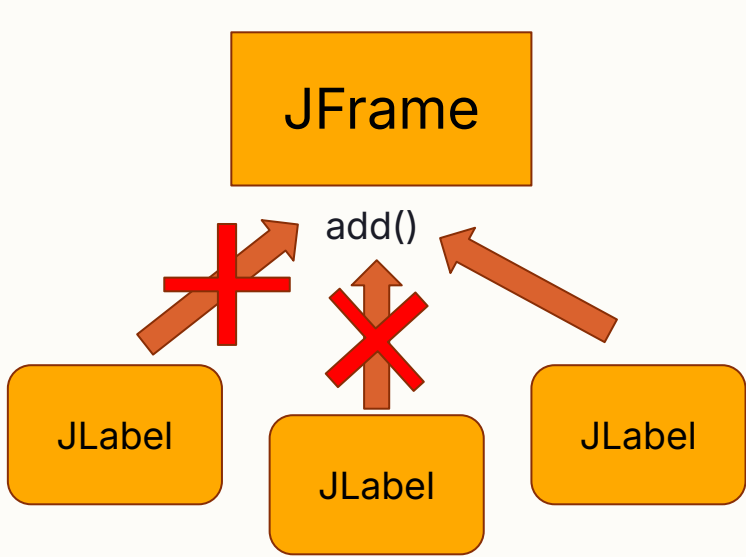- JFrame can only hold a single swing component
- So how do we fix this?

JPanels

- Swing gives a "container" class
- JPanel
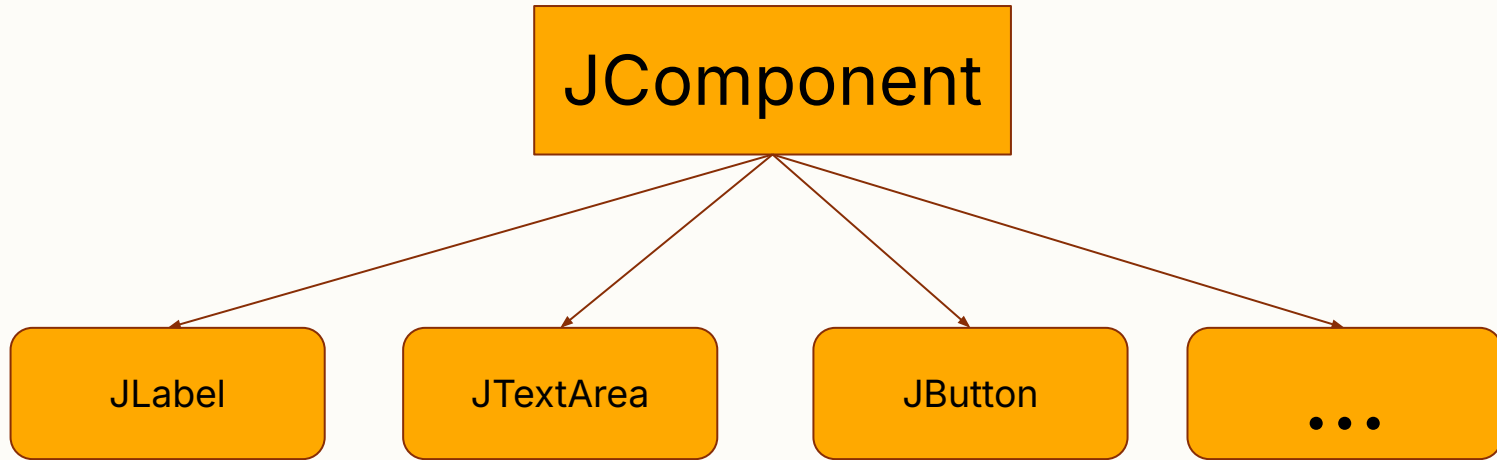 - Let's you add sub-components
  - Can add as many as you one

JPanels

Let's add some more things

```
           ┌──────────────────┐
           │   JComponent     │
           └──────────────────┘
        ┌──────┬──────┴──────┬──────────┐
        ▼      ▼             ▼          ▼
   ┌────────┐ ┌──────────┐ ┌─────────┐ ┌──────┐
   │ JLabel │ │ JTextArea│ │ JButton │ │ ...  │
   └────────┘ └──────────┘ └─────────┘ └──────┘
```

- Let's play around with some of these

JButtons

- JButton btn = new JButton()

- btn.addActionListener(ActionListener)

- ActionListener documentation

JButtons

- JButtons let us make our GUIs do things

- Let's play around with what they can do

- We can compose buttons in existing classes

Aside: Why does it look ancient???

- Sun
  - Developed Java Swing
  - Switched to JavaFX
- Oracle
  - Removed FX from JDK (Why?)

- OpenJDK now maintains Swing
- Third party + open source maintains FX

Look and Feel

- Appearance of application
- OS Defaults

Your swing code

JVM

MacOS

Windows

Linux