# **CSSE 220 Day 16**

**Event Based Programming** 

# Questions?

# Helpful hint on BigRational

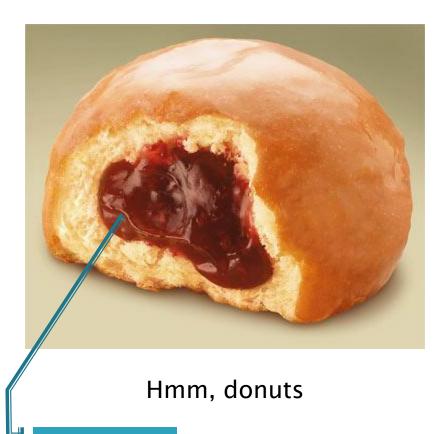
```
@Override
public boolean equals(Object object) {
     // throws a new ClassCastException if
     // the object isn't THIS_TYPE
     THIS_TYPE other = (THIS_TYPE)object;
     // Then compare this and other's fields.
}
This code is a good template for any time you are
 implementing equals.
Here, THIS_TYPE is ...
```

# Today

- Implementing a Graphical User Interface (GUI)
  - Adding components to containers
  - Layout Managers
  - Event-driven programming
    - Buttons, Mouse
  - Drawing on a component (review)
  - Applets
- A brief introduction, then SwingDemo2 time

#### Graphical User Interfaces in Java

- We say what to draw
- Java windowing library:
  - Draws it
  - Gets user input
  - Calls back to us with events
- We handle events



Gooey

### **Handling Events**

- Many kinds of events:
  - Mouse pressed, mouse released, mouse moved, mouse clicked, button clicked, key pressed, menu item selected, ...
- We create event listener objects
  - that implement the right interface
  - that handle the event as we wish
- We register our listener with an event source
  - Sources: buttons, menu items, graphics area, ...

### JButton example

- Three key steps:
  - 1. The JButton says which object(s) will respond when the JButton is pressed.
  - 2. The responding object(s) *implements ActionListener*.
  - 3. This means that there is an *actionPerformed* method that specifies what is to happen when the JButton is pressed

#### JButton example

2. Responder (this JButton) declares that it implements ActionListener

```
public class ExampleButton extends JButton
                              implements ActionListener {
    private ButtonAndMouseFrame frame;
    public ExampleButton(ButtonAndMouseFrame frame) {
         this.frame = frame;
                                               1. JButton says that it
         this.setText("Grow");
                                              will respond to its own
         this.addActionListener(this);
                                              button presses
                        Who responds to them?
     Who is generating
                        3. Responder (this JButton) implements the
     events?
                        required actionPerformed method, that says
                        what to do when the JButton is pressed
    @Override
    public void actionPerformed(ActionEvent buttonEvent)
         this.frame.grow();
```

A JButton often refers to one or more other objects (here, the ButtonAndMouseFrame) that it receives in its constructor and stores in a field. See buttonAndMouseExample in SwingDemo2 for the complete example.

#### Another example: Button in a Panel

- Button is the event source
- Panel has to respond to the event and therefore can easily listen for events.

```
public TopPanel extends JPanel implements ActionListener {
    private JButton changeColor;
    ...
    public TopPanel(){
        this.changeColor = new JButton("Click to change color");
        this.changeColor.addActionListener(this);
        this.add(changeColor);
    }

public void actionPerformed(ActionEvent e){
        //Change the background color of the panel
    }
}
```

### Key Layout Ideas

- Containers like JFrame and JPanel have an add (Component c) method
  - Adds a new component to be drawn
  - JFrame for the top-level container, JPanel to organize subcomponents
- You control how the components are placed on the window, and how they change when the window is resized, with a LayoutManager
  - You will experience FlowLayout and BorderLayout today

## SwingDemo2 Teams - Boutell

n	Team			n	Team		
01	evansea	houstoef		11	channmn	shumwanm	
15	ernsteac	wardsr					
03	baldwicd	maderli		12	roserrm	lamantas	
04	degrotpc	geislekj		13	hannantt	lint	
05	klaassmj	lapresga		14	zimmerka		
06	draycs	vermiljb					
07	audretad	fryjc	Driver (and ONLY the Driver): Check out <a href="mailto:SwingDemo2">SwingDemo2</a> from SVN  • The Navigator will check out the project in the				
08	kautzjr	wieganda					
09	cahilltr	hopkinaj	next session, after today's changes are committed.				
10	knightbk	weavergg	The project instructions are on the course web site, at <u>Programs ~</u>				
			<u>S</u> 1	<u>SwingDemo2 ~ instructions.htm</u>			
				Follow the practices of pair programming!			

Team number used in repository name:

http://svn.csse.rose-hulman.edu/repos/csse220-201020-**swingdemo2**-teamXX

## SwingDemo2 Teams - Mutchler

n	Team		n	Team	
21	Ahmed Alshaali		31	Elizabeth Hines &	
22	Kyle Apple & Donnie Quamme			Rebecca McCarthy	
23	Tom Atnip & Ryan Fuller		32	Chase Mathison & Alex Mullans	
24	Jeremy Bailey & Richard Thai		34	Susan Cisneros	
25	Devon Banks & Nathan Varner	Driver (and ONLY the Driver): Check out <i>SwingDemo2</i> from SVN			
26	Franklin Totten & Ruben Rodriguez		• The Navigator will check out the project in the next session, after today's changes are committed.		
27	Brian Collins & George Mammarella	The project instructions are on the course web site, at <a href="Programs">Programs</a>			
28	Ian Cundiff & Ben McDonald				
29	Katie Greenwald & Ann Say	~ SwingDemo2 ~ instructions.htm Follow the practices of pair programming!			
30	Alex Gumz & Jackson Melling				

Team number used in repository name:

http://svn.csse.rose-hulman.edu/repos/csse220-201020-**swingdemo2**-teamXX