

CSSE 220 Day 4

Practice Implementing Classes in Java
and an Intro to Java Graphics

Open WordGames project and specification

Outline

- ▶ WordGames: example and work time
- ▶ Live coding: a Java graphics program

Another WordGames Example

```
1 /**
2  * A NameDropper puts "foo says " before its given string, where foo is the name
3  * that the particular NameDropper drops.
4  *
5  * @author David Mutchler. Created December 4, 2009.
6  */
7 public class NameDropper {
8
9     private String nameToDrop;
10
11     /**
12      * Sets "Madonna" as this NameDropper's name to drop.
13      */
14     public NameDropper() {
15         this.nameToDrop = "Madonna";
16     }
17
18     /**
19      * Sets the given String as this NameDropper's name to drop.
20      *
21      * @param nameToDrop
22      */
23     public NameDropper(String nameToDrop) {
24         this.nameToDrop = nameToDrop;
25     }
26
27     /**
28      * Returns the String that results from putting "foo says " before its given
29      * string, where foo is the name that this particular NameDropper drops.
30      *
31      * @param stringToTransform
32      * @return the String that results from putting "foo says " before its given
33      *         string, where foo is the name that this particular NameDropper
34      *         drops.
35      */
36     public String transform(String stringToTransform) {
37         return this.nameToDrop + " says " + stringToTransform;
38     }
39 }
```

Word Games Work Time



- ▶ Java if statements are like C:

```
if (x < 12) {  
    System.out.println("x is small");  
}
```

- ▶ Use % for modulus, like C:

```
if (x % 2 == 0) {  
    System.out.println("x is even");  
}
```

- ▶ Java for loops are almost like C:

```
for (int i = 0; i < 10; i += 2) {  
    System.out.println("next even is " + i);  
}
```

On to Java Graphics

Check out `IntroToJavaGraphics` project from SVN

Simplest Java Graphics Program

```
import javax.swing.JFrame;
/**
 * From Ch 2, Big Java.
 * @author Cay Horstmann
 */
public class EmptyFrameViewer {
    /**
     * Draws a frame.
     * @param args ignored
     */
    public static void main(String[] args) {
        JFrame frame = new JFrame();
        frame.setSize(300, 400);
        frame.setTitle("An Empty Frame");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

This code is already in your project for today

Creates a graphics frame object

Configures it

Display the frame

Tells Java to exit program when user closes the frame

Live Coding

MyViewer and MyComponent
(Based on **RectangleViewer**
and **RectangleComponent**
from Big Java)

Schedule page has link to detailed instructions if you'd rather work at your own pace.

Other Shapes

- ▶ *new Ellipse2D.Double(double x, double y, double w, double h)*
- ▶ *new Line2D.Double(double x1, double y1, double x2, double y2)*
- ▶ *new Point2D.Double(double x, double y)*
- ▶ *new Line2D.Double(Point2D p1, Point2D p2)*
- ▶ *new Arc2D.Double(double x, double y, double w, double h, double start, double extent, int type)*
- ▶ Try these!
 - Add an ellipse and both kinds of lines to **MyComponent**

Adding Text

- ▶ To add some text to a component:
 - `graphics2.drawString("some text", x, y);`
- ▶ You can change the font *before* drawing the text:
 - `Font f = new Font("Times New Roman",
Font.PLAIN, 72);
graphics2.setFont(f);`

Style. Other alternatives are:
Font.BOLD,
Font.ITALIC, and
Font.BOLD | Font.ITALIC

Font size in
points

Colors

- ▶ To change the Graphics2D object's "pen" color:
 - *Color c = ...; // see below*
graphics2.setColor(c);
- ▶ Lots of colors:
 - *new Color(red, green, blue)*, all from 0 to 255
 - *Color.RED*, *Color.WHITE*, etc. (see Javadocs)
 - *new Color(red, green, blue, alpha)*, all from 0 to 255. *alpha* is transparency
- ▶ To fill interior of shape:
 - *graphics2.fill(box);*

Making Faces

- ▶ Due session 6
 - Look over project as part of HW4
 - Complete project as part of HW5
 - ▶ Implement a class that draws a face of a given size at a given location. You should also be able to mutate (translate & rotate) it.
1. Specification (in HW)
 2. Design together next session (UML)
 3. Code (incrementally)

