More about GUIs

GEEN163

"The best programmers are not marginally better than merely good ones. They are an order-of-magnitude better, measured by whatever standard: conceptual creativity, speed, ingenuity of design, or problem-solving ability."

Randall E. Stross

Teaching Evaluation

- The official University teaching evaluation survey is on Blackboard
- Complete the survey for ALL classes

Recitation Quiz

- This week there will be a quiz in recitation
- The quiz will cover arrays and classes

TuringsCraft

- Read chapter 8 of the textbook on arrays
- Do any 25 questions in section 8 of the TuringsCraft.com tutorial system
  - 4 points for each correct answer
  - maximum of 100 points
- Due by midnight Wednesday, Nov. 13, 2013

Lab Quiz

- Next week their will be a lab quiz
- You will have to write a program by yourself

Making an Applet GUI

1. Create a class that extends JApplet and implements java.awt.event.ActionListener
2. Create component objects
3. Specify the layout manager
4. Add the component objects to the container
5. Add an action listener to the components that respond to user input
6. Create an actionPerformed method

Interesting Objects

Many GUI applets can be created with a few simple components

- **JLabel** – Displays text in the GUI
- **JButton** – Creates a button you can press
- **JTextField** – Creates an input text box

- All of these objects are from the package javax.swing

What method is called when a JApplet starts?

A. main
B. init
C. actionPerformed
D. JApplet

Implementing ActionListener

- Implementing the ActionListener interface makes the program respond to GUI input
- Programs implementing ActionListener must have an actionPerformed method

```
public class MyApplet
        extends javax.swing.JApplet
        implements java.awt.event.ActionListener {
```

Adding a Listener

- The method addActionListener should be called on each GUI component that can do something
- Do not call addActionListener on components that do not do anything, like JLabel

```
javax.swing.JButton hitMe = new
                        javax.swing.JButton("Go");
hitMe.addActionListener( this );
```

“this” in Java means

A. this program
B. this method
C. this object
D. this class
E. this variable

Taking Action

- When a user clicks an object with an ActionListener, Java calls the method

```
public void actionPerformed(
                java.awt.event.ActionEvent   event) {
    // do something here
}
```

- This method should do what the user expects to happen when the button is pressed

Getting the Text

- You can get the text from a GUI component with the getText() method
- This is useful to get the text a person entered in the JTextField

```
javax.swing.JTextField llama = new
                        javax.swing.JTextField();
// in the actionPerformed method
String answer = llama.getText();
```

Changing a Component's Text

- A useful method is setText It can be used to change the text of any object that has text, such as JButtons, JLabels and JTextfields

```
javax.swing.JLabel  msg   =
        new javax.swing.JLabel("Do something");

msg.setText("Don't do something");
```

Who Dunit?

- Many GUIs have more than one active object
- You can determine where the action was taken by using the **getSource()** method of the event
- The result of **getSource()** should be compared to each GUI object

```
public void actionPerformed(
                java.awt.event.ActionEvent   event) {
    if ( event.getSource() == GUIobject ) {
```

Example Use of getSource()

```
JButton delete = new JButton("kill");
JButton save = new JButton("allow");
        - - - - - - - - - - - - - - - - - - - - - - - - -
public void actionPerformed(
                java.awt.event.ActionEvent   what) {
    if ( what.getSource() == delete ) {
        // delete something
    } else {
        // don't delete anything
    }
```

Looping in GUI Programs

- When you want a user to enter multiple values in a GUI program, you probably do **not** need a for loop or a while loop
- Each time the user enters a name, it will call actionPerformed

What will select the pressed button?

```
JButton hitMe = new JButton();
public void actionPerformed(
        java.awt.event.ActionEvent  bear)
```

A. `if( bear == hitme )`
B. `if( bear.getSource() == JButton )`
C. `if( bear.getSource() == hitMe )`
D. `if( event == JButton.getSource() )`
E. `if( this == bear.getSource() )`

GUI Applications

- The graphical user interface programs we have written so far were applets that ran in a web browser or the appletviewer
- You can also write application programs that have a Graphical User Interface

JFrame

- The javax.swing.JFrame class creates a window or frame that holds a GUI
- A JFrame object behaves almost the same as a JApplet except it runs outside a browser

JApplet to JFrame

To convert an applet to an application:

- Extend JFrame instead of JApplet
- Write a constructor instead of init method
- Write a main method to create the object
- Additional commands in constructor

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setVisible(true);
```

Extending JFrame

- Instead of extending the JApplet class, extend the javax.swing.JFrame class

```
public class MyFrame
    extends javax.swing.Jframe
    implements java.awt.event.ActionListener{
```

main Method

- An Java application with a GUI must have the usual main method
- The main method only needs to make the JFrame object

```
public static void main(String[] unused) {
    MyFrame  aardvark = new MyFrame();
}
```

Constructor Method

- The browser calls the init, run, start and stop methods
- Since a JFrame is not run in a browser, these methods are not automatically called
- Everything that was done in the init method can be done in the constructor method
- Just change the name "init" to the name of the class and remove the "void"

setDefaultCloseOperation

- Everyone knows that a program will terminate if you press the red **X** in the upper right corner
- In Java pressing the red X calls a method
- A program can do whatever you want it to do when you press the red X

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

- This makes the Java program terminate when you press the red X

setVisible

- Most GUI components have a setVisible method

  setVisible(boolean)

- If the boolean parameter is true, the object shows, if false it does not show
- Calling **`setVisible(true)`** makes the frame appear onscreen

Disabling GUI Components

This has little to do with JFrames

- Individual GUI components (i.e. JButtons) can be made invisible or disabled
- A disabled component is gray and cannot be selected

```
JButton myButton = new JButton("OK");
myButton.setEnabled(false);
```

- This will make the button gray and unusable

If your GUI applications does not call setDefaultCloseOperation, what happens when the red X is pressed?

A. Program terminates
B. Returns to beginning of the program
C. Window is minimized
D. Nothing

Example GUI Application part 1

```
import javax.swing.*;
public class LeftRight extends JFrame implements
                                java.awt.event.ActionListener {
   private JButton left, right;  // left right buttons
   private JLabel label;         // directions results
    public LeftRight() {
     left = new JButton("Left");
     right = new JButton("Right");
     label = new JLabel("Push a button");
     left.addActionListener ( this );
     right.addActionListener ( this );
     java.awt.Container pane = getContentPane();
     javax.swing.BoxLayout where = new
              javax.swing.BoxLayout(pane,javax.swing.BoxLayout.X_AXIS);
     setLayout( where );
     pane.add (left);
     pane.add (label);
     pane.add (right);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      pack();
      setVisible(true);
    }
```

Example GUI Application part 2

```
public void actionPerformed(
            java.awt.event.ActionEvent event) {
    if (event.getSource() == left)
       label.setText("   Left    ");
    else
       label.setText("   Right   ");
}

public static void main(String[] unused) {
     LeftRight aardvark = new LeftRight();
}
```

pack

- The pack method sizes the frame so that all its contents are at their preferred sizes
- When you use the pack method, you do not have to call setSize for the frame
- If you use absolute positioning of components instead of a layout manager, you do not need to call pack

A GUI **application** usually does **NOT**

A. have a constructor
B. extend JApplet
C. have a main method
D. implement actionListener

setTitle

- The setTitle method of a JFrame sets the text in the upper border of the window

```
setTitle("My GUI program");
```

Lazy Applet to Frame Conversion

- Change JApplet to JFrame
- Add to the end of the init() method:

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setVisible(true);
```

- create a main method

```
public static void main(String[] unused) {
    LazyFrame aardvark = new LazyFrame();
    aardvark.init();
}
```

Array Summary

- Arrays are declared by

```
int[] dog = new int[size];
```

- When you create an array of objects, you have to create the objects separately
- Array elements are always used with brackets
- If you pass an array element to a method, you cannot change its value in the calling program
- If you pass an entire array to a method, you can change its value in the calling program

Array Summary (cont.)

- Arrays are almost always used with loops
- Array indexes range from zero to one less than the size of the array
- You can get the maximum array size with arrayname.length

Write with your team

- Create a class called Room that contains a String giving the name of the course and an int holding the number of students
- Create a constructor that initializes the course
- Create a method to add a parameter value to the student count

Possible Solution

```
public class Room {
        String course;
        int  numStudents = 0;
        public Room( String user ) {
                course = user;
        }
        public addCount( int howMany) {
                numStudents += howMany;
        }
}
```

Schedule

Monday, November 11 Arrays sections 8.3	Wednesday, November 13 More on GUIs	Friday, November 15 More on GUIs
Monday, November 18 Programming practice Lab	Wednesday, November 20 Programming practice Quiz Lab	Friday, November 22 review Quiz
Monday, November 25 **Exam 3**	Wednesday, November 27 *Thanksgiving Holiday* *(no classes)*	Friday, November 29 *Thanksgiving Holiday* *(no classes)*
Monday, December 2 Software engineering	Wednesday, December 4 review	

TuringsCraft

- Read chapter 8 of the textbook on arrays
- Do any 25 questions in section 8 of the TuringsCraft.com tutorial system
  - 4 points for each correct answer
  - maximum of 100 points
- Due by midnight Wednesday, Nov. 13, 2013

Recitation Quiz

- This week there will be a quiz in recitation
- The quiz will cover arrays and classes

Lab Quiz

- Next week their will be a lab quiz
- You will have to write a program by yourself

Teaching Evaluation

- The official University teaching evaluation survey is on Blackboard
- Complete the survey for ALL classes