

## Java GUIs 2 (Events)

3-18-2003

---

---

---

---

---

---

---

---

## Opening Discussion

- Did you have a good Spring Break?
- What did we discuss last class?
- Do you have any questions about the assignments? Remember that the design of #4 is due today.
- In the past, when you have gotten input from a user, what have you done? Will that work in a GUI?

---

---

---

---

---

---

---

---

## Midterm Comments

- I was a bit frustrated while grading these, though the grades didn't come out all that poorly.
- It was frustrating when several people would miss a question from one of the quizzes where the answer has been on the web for over a month now.
- Distribution of grades was as follows:  
A=9, B=5, C=5, D=2, F=4

---

---

---

---

---

---

---

---

### **Specific Question Comments**

- Make sure you look at the answers on the web for this exam even if you did well.
- What does `System.out.println(5/3)` print?
- `OddList` does a partial sort. Puts it at the beginning if it belongs, otherwise puts it at the end.
- Too many people missed min sort. Think of what it does, then convert to code.
- Arrays are arrays of references, but only hold subtypes of the declared type.
- Strings are immutable and you can't use `[]`.
- Immutable -> copy to alter, but not to share.

---

---

---

---

---

---

---

---

### **Alteration to BorderLayout**

- Looking at the Javadocs over the break I found that they have altered the way that you are supposed to use a `BorderLayout`.
- Now when you add to a container that uses `BorderLayout` you are not supposed to give a string ("`North`", "`Center`", ...) followed by the object. Instead you should give the object followed by `BorderLayout.NORTH,...`

---

---

---

---

---

---

---

---

### **Events in Java**

- Since Java 1.1 the event model in Java has used objects called Event Listeners that have functions that get called when certain events happen.
- It should be noted that the event code occurs in a separate thread than main execution so virtually all GUI code in Java is multithreaded.

---

---

---

---

---

---

---

---

## **Event Listeners**

- The Event Listener interfaces are part of `java.awt.event` and `javax.swing.event`. There are different types of listeners for the different types of events that components can have.
- You can tell a Component that there is a Listener it needs to notify in case of events with the add methods in the Component for that type of Listener.
- Adapter classes are also provided.

---

---

---

---

---

---

---

---

## **Anonymous Inner Classes**

- Because the event model is based very heavily on creating new classes to handle the events, anonymous inner classes can be remarkably handy.
- Remember that an anonymous inner class is a class defined in line when an object is created and it has to be the subtype of a named class or interface. The overridden methods are defined inline as well.

---

---

---

---

---

---

---

---

## **Events**

- The methods of the Listeners typically each take one argument, which is of an event type. There are different event classes for each of the different events. These can give you different types of information for whatever is relevant to that type of event.
- They all also let you find the Component the event occurred on and other general information.

---

---

---

---

---

---

---

---

## Code

- Now let's go and write and look at a bit more complex GUI code and have it so that it has to handle some events from the user.
- In your project, you will be writing code like this for the functions that get panels (either edit panels or game status panels).

---

---

---

---

---

---

---

---

## Minute Essay

- Write a small piece of code that creates a new JButton and adds a new ActionListener to it such that when the button is clicked the program calls System.exit(0) and stops.
- The design for assignment #4 is due by midnight tonight.

---

---

---

---

---

---

---

---