

GUIs 2: Event Handling

10-22-2002

Opening Discussion

- Do you have any questions about the test?
- What did we talk about last class?
- Do you have assignment questions?
- We looked briefly at code that handled user input in the earlier classes we did talking about polymorphism and inner classes. Does anyone remember what that code was like?

Events in Java

- Since Java 1.1 the event model in Java has used object 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. 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 a bit more complex GUI code that could be used as a panel in a game and have it so that it has to handle some events from the user.

Minute Essay

- Write a small piece of code that create 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 #5 is due next class.
