

Sorting and OO Design

2-13-2003

Opening Discussion

- What did we talk about last class?
- Do you have any questions about the assignment? Let's look at how you will be submitting it.
- You aren't supposed to see the code in the game infrastructure, just as you can't see the code in the standard Java libraries.

Polymorphic Code for Other Sorts

- Let's go ahead and write the code for doing polymorphic sorts with bubble sort, selection sort, and Shell sort.
- We will also write some different comparators that might make their use more clear.

Separation of Interface from Implementation

- In a well designed OO program you should be able to completely change HOW a class does something as long as the end result adheres to the same interface.
- This is why you hide data members and put in accessor methods.
- Extra care must be taken with anything visible to the outside world because changing it can break other code.

Running Example

- To help show good design outside of the context of your project, we will be looking at this example of a drawing program. I have written some of the preliminary aspects of it so that you can see the overall design.

Minute Essay

- Now do you feel comfortable with the different sorts? What about making them polymorphic? Why is making things private important in good OO design?
- Make sure you turn in the code to me by midnight tonight.
- Quiz #2 is next class.
