Types and the Class Diagram/ First Look at Code

2/23/2009

Opening Discussion

- Recipes vs. algorithms
  - Specificity
  - Simplicity/logic
  - Furniture assembly/order matters
  - Making the objects/ingredients
- Nature of programming
  - Figure out step to solve a problem.
  - Look for ideal organization.
  - Put in terms/language a computer understands.

Minute Essay questions

- Could you make the same game with different icons?
- Why can't you move a circle into an uninhabited space?
- Should you know how these codes are written?
- Do algorithms have to be rewritten for each new program?
- More information on the project.

Greenfoot Class Diagram

- You might have wondered some about the area in Greenfoot that shows you classes and why it is organized the way it is.
- The organization relates to the type system in Java.
- Arrows represent an is-a relationship.

Inheritance/Subtyping

- In programming we call this type of relationship inheritance because the subclass gets all the things in the superclass.
- It also provides subtyping. This means that anyplace where you use the supertype, any subtype can be substituted.
- Greenfoot is programmed to work with Actors in general. It will work with any subtype of actor as well.

First Look at Java

- Now we will take our first look at real Java code. You can look at the program behind any class by double clicking on the class. This brings up an editor window.
- Java programs can be edited with any text editor. Word is not a text editor. Notepad is.
- Let's start with the Leaf class in Wombat.
- Now the Wombat itself.
- Last we can look at the grid in Goober's Lab.

Minute Essay

- Do you have any questions about the material we discussed today?
- Next class I will give you your first set of interclass problems for you to work on for the “show your code” day.