Generics and Enums

1-26-2005

Opening Discussion

- Do you have any questions about the reading?
- Do you have any questions about the assignment?
- Let's look at how you will log into the grade submission program and make sure everyone can get in.

Coding Polymorphism

- Before we go on we want to finish up inheritance and polymorphism. We didn't write any code last time to actually display inheritance. We should do that today.
- I want you to write code that represents simple math formulas. You will need an interface that you can call Formula and you will make a number of subclasses of it (Value, Plus, Minus, Multiply, Divide). Formula will have a method “double getValue()”. Each type can have its own constructor. Put everything in a class called FormulaParser.

Restrictions in Java

- Java places some restrictions on inheritance to simplify the language. The main restriction is that you can only extend one class. Doing otherwise, multiple inheritance, tends to make things very complex.
- There are times when you want to have a class be a subtype of two different types though. To allow this Java has a construct called an interface. Interfaces have no data (they can have static data) and all methods in them are abstract. They only define what you can do with them, not how to do it. You can implement as many interfaces as you want.

Java 5.0 Features

- Java 5.0 adds quite a few features to the Java language that make it easier to write real code in Java. Most of class today we will focus on two big ones, generics and enums. There are a few others worth mentioning.
- Autoboxing – primitive types are not objects, but sometimes you need to treat them that way. This automates the process.
- Varargs – Java now has syntax for passing a variable number of arguments. We won't use this in this class.
- Foreach loop – A handy shortcut for when you want to do something with all elements of a container.

Questions for Generics

- What is the syntax for making a generic class in Java?
- What is the meaning of a generic class?
- What is the syntax of making a generic method?
- What neat thing do generic methods allow you to do?

Generics

- The most significant feature added to Java 5.0 was that of generics. Generics provide a form of parametric polymorphism, typically for code that can take any type, but might be limited to a specific type for one instance.
- The most common use of this is for containers as container classes typically should be able to hold anything, but any one container is generally intended to hold only one type.
- In practice, generics give you extra type safety and prevent you from doing a lot of type casts.

Enums

- What is the point of enums?
- What is the syntax of enums?
- Why would you use the enum keyword instead of declaring constants the way people had done previously?
- I typically avoid the use of switch in programming, especially with OO programming I find that polymorphism and the “state” pattern provide a more robust mechanism for getting the same result.

Inner Classes

- What is an inner class?
- What types of inner classes are there?
- What type should you use by default? Why?

Code

- Let's write some code together. This code is going to show the power of polymorphism first, as well as the lack of safety you can have.
- We will then change it so that it uses generics to be more type safe.
- The code I want to write is a function called map which takes a list of data and a function and returns a list where the elements are the results of applying the function to the original elements. This function is extremely common in functional languages.

Minute Essay

- Does the concept of polymorphism make more sense now? How does the fact that all variables in Java are references help enable the polymorphism in Java?
- Remember to turn in your assignment code today.
- Your first quiz is next class.