Using Polymorphism

4/21/2008
Let's look at some solutions to the interclass problem.
Last class we talked about inheritance, what it provides to us, and how we do it in Java.
  - Code reuse
  - Subtyping

We use the extends keyword when a class inherits from a class or implements when a class inherits from an interface.

Abstract classes can't be instantiated. Abstract methods aren't implemented in that class.

Interfaces are like completely abstract classes.

All classes in Java inherit from the Object class.
Let's look now at an example of inheritance in code so that we can see not only how to use it, but also the syntax for it in Java.

Note that Eclipse can help you put in your inheritance if you specify the supertype in the dialog when you create a class.

Use the super keyword to refer to the version of a method in the supertype.
The term polymorphism comes from Greek and it literally means many shapes. In programming it means many types.

A polymorphic type is a type that can represent many other types. A polymorphic operation/method is one that can work with many different types.

Polymorphism gives us a lot of power in our programming and can drastically reduce the amount of code we have to write because one method can be used with many different types.

Subtyping from inheritance gives us polymorphism.
The GUI libraries in Java probably use inheritance more than any other set of libraries.

There are two main packages that you use when doing GUIs in Java.

- `java.awt` – the Abstract Window Toolkit was the original GUI library. It still has important helper classes.
- `javax.swing` – this is a “pure” Java library built on top of AWT that we will be using.

Let's look in that API at the Swing classes.
Everything that goes into a GUI in Java inherits from Component. (In Swing they inherit from JComponent.)

This is where polymorphism comes into play because the code can be written to work with a Component and it will work with a button, check box, drop box, or any other component.

The Swing library has all the basic components you would expect to be able to put into GUls as well as many others.
For windows in the Swing library you use JFrames.
Let's look at the JFrame class and write some code to open up a window.
Now let's add one component to that frame.
What we just wrote only has one limitation, we only got to add one component to the frame.

For adding multiple elements we use Layout Managers. These are classes that control how things go into a GUI container.

There are many layout types, but we will only deal with a few.
- FlowLayout
- GridLayout
- BorderLayout

Some Swing components have their own layouts.
Have you done any of the reading recently? What have you read? Has it helped?

Interclass Problem – Write a GUI that brings up a frame and puts one button and one text box in the frame. Experiment with how to lay it out to make it look good.