Inheritance and Subtyping

1-19-2011
Opening Discussion

- Minute Essay Comments
  - Switching editors. “:w” in your Eclipse files.
  - Autoformatting and being “helpful”.
  - The purpose of packages.
- Solutions to the IcP.
Special Methods

- **apply**
  - If you have an apply method, it will be called when you use an object in a function like syntax.
  - This is how you index into collections.
  - \( c(i) \) is the same as \( c\text{.apply}(i) \)

- **update**
  - Takes two arguments and gives you assignment into an index.
  - \( c(i)=o \) is the same as \( c\text{.update}(i,o) \)
More Special Methods

- **Symbolic Operators**
  - You can use symbolic method names.
  - If they take only one argument you can use them as operators.

- **Property Assignment**
  - A method name that ends in `_=` and takes a single argument is interpreted and creating an assignment to a member.
  - Public vars actually compile to a private var and an accessor method with an assignment method.
Polymorphism

- Literally this means many shapes.
- In programming it means many types.
- The code we wrote in the first semester was generally monomorphic. It worked for one type.
- Polymorphic code is much more powerful. You write it once and it works for lots of types.
- A lot of this semester and the benefit of OO is about abstraction and polymorphism.
- Universal polymorphism means code can work with an infinite number of types.
Subtyping

- If type B is a subtype of type A, then anyplace in the code that requires type A, you can provide an object of type B.
- This gives us a form of Universal polymorphism called inclusion polymorphism.
The way we get inheritance in Scala (and most class based OOPLs) is through inheritance.

If B inherits from A it becomes a subtype of A.

B also gets all the members (data and functions) of A. This is effectively why it is a subtype. Anything you could do to A you can do to B because B got it all from A.

You can only inherit from one class and you do so with the extends keyword.

- class B extends A { ... }
Part of the real power of inheritance comes from the fact that you can override methods.

An overridden declaration is one in the subclass with the same name and type as the superclass. Methods need to match arguments as well.

You must label overriding members with the keyword override. We did this for paint last semester.

toString is a good method to override.
Questions?

If you use a local copy of the book, remember to pull it down occasionally so you get updates.