

Generic Java

Administrivia

- ACM article from March, “Comments are More Important than Code”
- Check out <http://acmqueue.com> to find it
- Not required reading, but interesting
- Otherwise: keep hacking

The bane of casting

```
// Java 1.4 code littered with stuff like this:  
Map m=new HashMap();  
m.put("Hello",new Integer(17));  
m.put("world",new Integer(-4));  
  
// later...  
Integer i=(Integer)m.get("world");
```

Bane of casting

- Issue: Java 1.4 containers want to be generic -- hold any object type
- Java has no way to express “container of any object type”
- Punt by using `Object` as a proxy
- Subtle conceptual problem:
 - `Object` doesn't mean “any one specific type”
 - It means: anything that is a subclass of `Object`
- Issue 1: Programmer has to track types by hand
- Issue 2: Collection can have heterogenous types
- Consequence: lots of casting and icky type checks

Problem w/ Object way

```
Map m=new HashMap();
m.put("Hello",new Integer(42));
m.put("Goodbye",new Integer(-4));

// later, someone evil/stupid comes in...
m.put("Eris",new LinkedList());

// and you find out when...
for (Iterator i=m.keySet().iterator();
i.hasNext();) {
    Integer x=(Integer)m.get(i.next());    // BLAM!
}
```

Problem w/ Object way

```
Map m=new HashMap();
m.put("Hello",new Integer(42));
m.put("Goodbye",new Integer(-4));

// later, someone evil/stupid comes in...
m.put("Eris",new LinkedList());

// and you find out when...
for (Iterator i=m.keySet().iterator();
i.hasNext();) {
    Object o=m.get(i.next());
    if (o instanceof Integer) { ... } // ick...
    else { ... } // Duhhh... What now?
}
```

Generics to the rescue (?)

- New mechanism in Java 1.5: Generic types
 - `List<String> l=new LinkedList<String>();`
- Semantics:
 - “This list holds only `Strings` (possibly including subtypes -- i.e., subclasses -- of `Strings`)”
- Makes life easier in a lot of ways

The Generic way

```
Map<String,Integer> m=new
    HashMap<String,Integer>();
m.put("Hello",42);
m.put("Goodbye",-4);

// later, someone evil/stupid comes in...
m.put("Eris",new LinkedList());
// Hah!  Compile time error!  Can't do that!
```