

Lecture 24: Interfaces (continued)

Last time:

1. Array Copying
2. Interfaces

Today:

1. Wrappers
2. Comparable (Interfaces)



Wrappers

- We may want to treat primitives as though they were objects
- For example, generic routines (like `PsychoAnalyze`) can be implemented using interfaces ... but they are not usable on primitive types
- To overcome this problem, Java provides **wrappers** for primitive types
 - Wrappers: classes whose objects contain single values of the “wrapped type”
 - Wrappers also contain other useful conversion operations (to / from `String`, etc.)
 - Wrappers included in `java.lang`:
 - `Byte`
 - `Short`
 - `Integer`
 - `Long`
 - `Float`
 - `Double`
 - `Character`
 - `Boolean`

The Integer Wrapper

- The documentation is on-line at <http://java.sun.com/j2se/1.5.0/docs/api/>
- Notes
 - Constructors
 - Implements Comparable
 - Documentation says “Comparable<Integer>”
 - Comparable in Java 5.0 is a *generic* interface
 - We’ll understand this more (in a moment)
 - Has equals method, etc.

Wrappers Allow Use of findMin

- What is printed as result of following:

```
Integer i = new Integer (1);
```

```
Integer j = new Integer (2);
```

```
System.out.println (FindMin.findMin(i, j));
```

- Answer: 1

- Integer implements Comparable, so compareTo exists
- It also implements toString, so System.out.println prints integer value correctly

Another Example: findMin

- Wanted: operation findMin for finding least `String` in `String` array
- Assumption: arrays are non-empty
- Method:
 - Store initial array element in temporary variable
 - Iterate through array, comparing each element to temporary variable
 - If element is less than temporary variable, set temporary variable to it
- How can we make this method polymorphic?
(See `FindMin.java` for this class.)

Adapting Cat to Implement Comparable



- Must implement `compareTo` method:
`int compareTo(Object o)`
- What is `Object`?
 - Type of all possible objects in any class
 - Shortcoming of (earlier) Java: no good way to say “same type as `this`”
 - Instead: Implementation must take any object

Implementation of compareTo for Cat



```
public class Cat implements Comparable{
...
    public int compareTo(Object other) {
        Cat x = (Cat)other;
        if (weight == x.weight){
            return 0;
        }
        return (weight > x.weight)? 1 : -1;
    }
}
```

- Note: Any number of interfaces are allowed! (Only one is used above, but could add another one, e.g., Comparable, Printable, ...)

Using Polymorphic findMin

- Note that the same findMin method can be used on `Cat`, `String`

What about `int`? `char`?

- Polymorphic `findMin` can be used on any class implementing `Comparable`
- What about primitive types (`int`, `char`, `double`, etc.)?
 - They are not classes
 - They do not implement `Comparable`
 - Hence `findMin` cannot be used on them
 - That's why we use wrappers!