CS 213 – Software Methodology

Spring 2017

Sesh Venugopal

Lecture 7 – Feb 7

Interfaces – Part 1

Comparing for inequality in an algorithm implementation

```
public class Searcher {
   ...
   public static<T> boolean
   binarySearch(T[] list, T target) {
       ...
       list[index].___?___target

       ...
   }
   ...
}
```

How to compare for inequality? All we know Is `T` is some `Object`, but `Object` does not define an inequality comparison method

Need to have a type definition for T that will guarantee the existence of a method that can be used for inequality comparison

Comparing for inequality in an algorithm implementation

```
public class Searcher {
   ...
   public static<T> boolean
   binarySearch(T[] list,
                T target) {
       ...
       list[index].___?___target
       ...
   }
}
```

Solution is to use a pre-existing interface that is known to prescribe an inequality comparison method.

Or, define an appropriate interface if none exists.

The interface introduces a type that can be checked by the compiler for match between caller and callee

e.g. `java.lang.Comparable` interface, which defines a `compareTo` method

```
public static
<T extends Comparable<T>>

    list[index].compareTo(target)
```

Type `T` is not just any class, but one that implements the `java.lang.Comparable` interface, or extends a class (any number of levels down the inheritance chain) that implements this interface

Interfaces

The term “interface” GENERALLY refers to the means by which an object can be manipulated by its clients – in this sense the public methods of an object comprise its implicit interface.

For example, public methods `push`, `pop`, `isEmpty` (as well as constructors) in a `Stack` implicitly define its interface – these methods/constructors will be used by clients to create and manipulate stacks

Java provides a way (keyword `interface`) to define an explicit interface that can be implemented (keyword `implements`) by classes

```
public interface I { . . . }
public class X implements I { . . . }
```

Interfaces

Interface defined in `java.lang` package

```
public interface Comparable<T> {
   int compareTo(T o);
}
```

For method `compareTo`, keywords `public` and `abstract` are omitted by convention (redundant if written)

Prescribes a single, `compareTo` method, but there is no method body, just a semicolon terminator