interfaces

Professor Evan Korth

Interfaces

- In Java, only single inheritance is permitted. However, Java provides a construct called an interface which can be implemented by a class.
- Interfaces are similar to abstract classes (we will compare the two soon).
- A class can implement any number of interfaces. In effect using interfaces gives us the benefit of multiple inheritance without many of it’s problems.
- Interfaces are compiled into bytecode just like classes.
- Interfaces cannot be instantiated.
- Can use interface as a data type for variables.
- Can also use an interface as the result of a cast operation.
- Interfaces can contain only abstract methods and constants.

Interfaces (cont)

- An interface is created with the following syntax:

```
modifier interface interfaceID
{
  //constants/method signatures
}
```

Interfaces

- You can assign an object of type C to any variable of the interfaces it implements:

```
interface I1 { void a(); }
interface I2 { void b(); }
class C implements I1, I2 { void a() { … }; void b() { … } }
…
```

- Interfaces can extend a super interface:

```
interface I1 { void a(); }
interface I2 extends I1 { void b(); }
class C implements I2 { void a() { … }; void b() { … } }
```

syntax

```
public class Circle extends
GeometricObject implements
Comparable {

/* define class here make sure to
implement all the abstract methods
contained in the interface(s)*/
}
```

Interfaces (cont)

- An interface can extend other interfaces with the following syntax:

```
modifier interface interfaceID extends
  comma-delimited-list-of-interfaces
{
   //constants/method signatures
}
```

- Obviously, any class which implements a “sub-interface” will have to implement each of the methods contained in it’s “super-interfaces”

Interface vs. abstract class

	Interface	Abstract class
Fields	Only constants	Constants and variable data
Methods	No implementation allowed (no abstract modifier necessary)	Abstract or concrete

Interface vs. abstract class (cont)

	Interface	Abstract class
Inheritance	A subclass can implement many interfaces	A subclass can inherit only one class
	Can extend numerous interfaces	Can implement numerous interfaces
	Cannot extend a class	Extends one class

Interface vs. abstract class (cont)

	Interface	Abstract class
Root	none	Object (of all classes)
names	Adjective or Nouns	Nouns

`Comparable` interface

- This interface imposes a total ordering on the objects of each class that implements it. This ordering is referred to as the class's *natural ordering*, and the class's compareTo method is referred to as its *natural comparison method*.

int **compareTo** (Object o)
Compares this object with the specified object for order. Returns a negative integer, zero, or a positive integer as this object is less than, equal to, or greater than the specified object.

From java.sun.com documentation