CS 112 / Section 02

Yasemin Türedi

Osman Berat Okutan

Lecture notes of April 10, 2008;

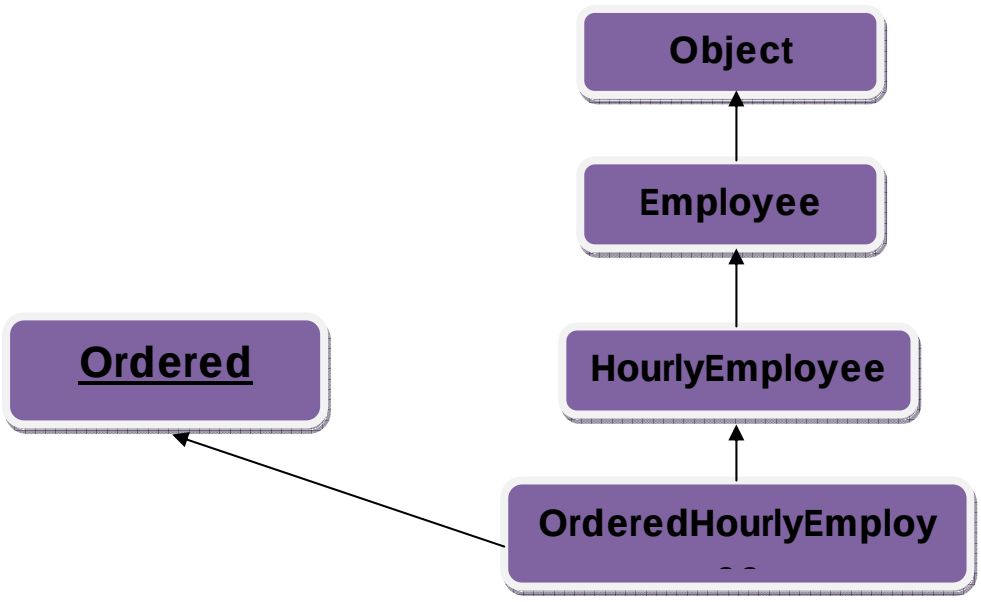
INTERFACES:

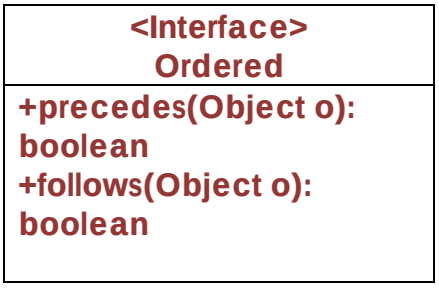
An interface is not a class.

An interface is a type. ← **Comparable**

An interface is a property of a class that says what methods it must have.

```
public interface Ordered {
   public boolean precedes (Object other);
   public boolean follows (Object other);
  /* Object1.follows (Object2) == Object2.precedes (Object1) */
}
```






```
public class OrderedHourlyEmployee extends HourlyEmployee implements Ordered {

    public boolean precedes (Object other) {

            if (other==null)

            return false;

        else if ( !(other instanceof OrderedHourlyEmployee) )

            /* else if (other.getClass() != getClass())*/

                return false;

        else {

            OrderedHourlyEmployee otherEmployee ==
(OrderedHourlyEmployee) other;

            return (getPay() < otherEmployee.getPay());

        }

    }

}
```

ABSTRACT CLASSES IMPLEMENTING INTERFACES:

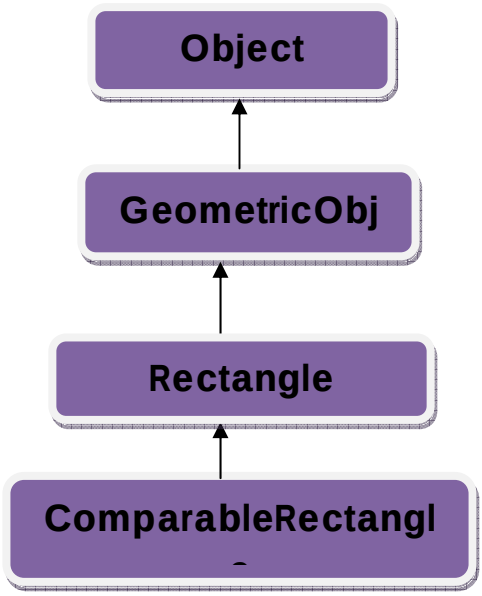
```
public abstract class MyAbstractClass implements Ordered {
    private int number;

    private char grade;

    public boolean precedes (Object other) {
      if (other==null)

            return false;

        else if ( !(other instanceof MyAbstractClass) )

            /* else if (other.getClass() != getClass())*/

                return false;

        else {

            MyAbstarctClass otherAbstract == (MyAbstractClass) other;

            return ( this.number < otherAbstract.number );

       }

    }

    public abstract boolean follows (Object other) ;

}  //end of class
```

THE COMPARABLE INTERFACE:

Available in java.lang package and so automatically available to your program.

Public int compareTo(Object other);

<Interface> java.lang comparable
+compareTo(Object other) : int




```
public class ComparableRectangle extends Rectangle implements
Comparable {
    public ComparableRectangle ( double width, double height )
      { super( width, height ) }
    public int compareTo ( Object other ) {
      if (getArea() > ((ComparableRectangle) other).getArea())
            return 1;
      else if (getArea() < ((ComparableRectangle) other).getArea())
                  return -1;
        else {
             return 0;
        }
```

```
    }
}
```

QUESTIONS:

1) How does the concept of interface help programmer in Java?

2) List the similiarities and differences between an abstract class and interface.

3) Fix the errors in the following program:

```
public interface Factorial {
   private int ZEROFACTORIAL=1;
   public NaturalNumber factorial(){}
} //end of the interface
public class NaturalNumber implements Comparable,Factorial{
   public int n;
   public NaturalNumber(int num){
      n=Math.abs(num);
   }
   public int compareTo(NaturalNumber otherNumber){
      int result=0;
      if( n>otherNumber.n )
         result=1;
      if( n<otherNumber.n )
         result=-1;
      return result;
   }
   public NaturalNumber factorial(){
```

```
        int result=0;
        if( n==0 )
            result=ZEROFACTORIAL;
        else
            for(int i=1;i<=n;i++)
                result*=i;
        return result;
    }
}
```

4) Write a Hotel class implementing Comparable interfaces with local variables (int)star and (String)name. Create a constructor taking enough parameters. Add a compare() method.

5) Does an abstract class implementing an interface have to have declaration of methods in the interface? What is the case for non-abstract child classes of abstract classes?

ANSWERS:

1) Interfaces give the ability of encapsulating some methods related with each other insome aspects, without declaring them. This is adventagous because some methods (e.g compareTo()) can be applied to many object but have different decleration. While a programmer writing a long program, to not forget decleration of some methods, from beggining he/she can implement one ore more interfaces containing desired methods.

2)

SIMILARITIES
1.) Can not be initilaized
2.) It is possible to have methods without declarition
3) When they applied to a non-abstract class, that class has to declare methods in interface/abstract class.

DIFFERENCES	
INTERFACE	ABSTRACT CLASS
1) No consturctor.	1) Can have constructor.
2) Multi interfaces can be implemented.	2) Only one abstract class can be extended
3) No method declaration.	3) Methods can be declared.
4) No local variables.	4) Can have local variables.

3)

```
public interface Factorial {
   int ZEROFACTORIAL=1;
   public NaturalNumber factorial();
} //end of the interface
public class NaturalNumber implements Comparable,Factorial{
   public int n;
   public NaturalNumber(int num){
      n=Math.abs(num);
   }
   public int compareTo(Object o){
      NaturalNumber otherNumber=(NaturalNumber)o;
      int result=0;
      if( n>otherNumber.n )
         result=1;
      if( n<otherNumber.n )
         result=-1;
      return result;
   }
   public NaturalNumber factorial(){
      int result=0;
      if( n==0 )
         result=ZEROFACTORIAL;
      else
         for(int i=1;i<=n;i++)
            result*=i;
```

```
        return new NaturalNumber(result);
    }
}
```

4)

```
public class Hotel implements Comparable {
    private int star;
    private String name;
    public Hotel(int s,String n){
        star=(s<1 | |s>5)?1:s;
        name=n;
    }

    public int compareTo(Object o){
        int result=-1;
        Hotel otherHotel=(Hotel)o;
        if( star>otherHotel.star )
            result=1;
        else{
            if( star==otherHotel.star )
                result=0;
        }
        return result;
    }
    public class Hotel implements Comparable {
       public Hotel(int s,String n){
    }
}
```

5) An abstract class implementing an interface does not have to contain decleration of methods in the interface. However, non-abstract classes extending the abstract class have to contain declerations.