Lesson 5.3 Interfaces

Monday, January 14, 2013
7:47 AM

The Class

Steps to implement an interface:

1. Implement the interface in the class headers of whichever classes you want to implement it. This now requires that the class must provide all the methods specified in the interface.
2. Write the methods required by the interface. Each class will require different details for the method bodies, but the headers must match the interface.

```
public class Person  implements Emotable  {
    private String name;
    private int age;


    public Person (String name, int age) {
         this.name = name;
         this.age = age;

    }

    public void changeName(String newName){
         name = newName;
    }

    public void changeAge (int newAge){
         age = newAge;
    }

    public String toString() {
         return name + " - Age " + age;
    }

\   public String getEmotion(){
      .....

    }

    public void changeEmotion(){
      .....
    }


}
```

```
public interface Emotable {

    public String getEmotion();

    public void changeEmotion();  // toggle emotion back and
                                  // forth between two values

}
```

Interfaces

- A collection of abstract methods (methods with headers but no bodies), and optionally some static variables.
- Used to promote code reuse through polymorphism.

The Client

```
public class Client {
    public static void main(String[] args) {

        Emotable e1 = new Person(. . .)
        Emotable e2 = new Smileyface(. . .)
        Emotable e3 = new Person(. . .)

    }
}
```

References are declared as interface type

Objects are instantiated from any class that implements Emotable

```
e1.changeEmotion();
e2.changeEmotion();
```

Only methods specified in the interface can be called on these references

Polymorphism: These refer to objects of various classes which all implement Emotable. Calls to .changeEmotion() behave differently depending on which kind of object is being referenced.

However, as far as the client is concerned, all the calls look the same.

```
((Smileyface) e2).frown();
```

casting object references. if you need to refer to the the object as a Smileyface.