

## 第5章-接口与多态

接口的数据成员一定要赋初值，且此值将不能再更改，允许省略final关键字

接口中的方法必须是“抽象方法”，不能有方法体，允许省略public及abstract关键字

接口的实现，必须实现接口中的所有方法，来自接口的方法必须声明成public

Java的设计以简单实用为导向，不允许一个类有多个父类；但允许一个类可以实现多个接口，通过这种机制可实现多重继承。

接口的扩展（类似于类的继承）

塑型（type-casting）又称为类型转换，方式有：隐式（自动）的类型转换，显式（强制）的类型转换。

隐式（自动）的类型转换

基本数据类型：相同类型之间存储容量低的自动向存储容量高的类型转换。

引用变量：被塑型为更一般的类；被塑型为对象所属类实现的接口属性。

显式（强制）的类型转换

基本数据类型

引用变量：还原为本来的类型。（我想这种机制应该是Java中引用机制实现的）

塑型的本质就是用不同引用指向同一个对象。

引用类型本身，决定了可以访问的成员，这是编译时决定的。

多态是指不同类型的对象可以响应相同的消息。

绑定指将一个方法调用同一个方法主题连接到一起。

```
1  abstract class Driver{
2      public Driver(){}
3      public abstract void drives();
4  }
5
6  class FemaleDriver extends Driver{
7      public FemaleDriver(){}
8      public void drives(){
9          System.out.println("a female");
10     }
11 }
12
13 class MaleDriver extends Driver{
14     public MaleDriver(){}
15     public void drives(){
16         System.out.println("a male");
17     }
18 }
19 public class Test1{
20     public static void main(String[] args){
21         Driver a = new MaleDriver();
22         a.drives();
23         Driver b = new FemaleDriver();
```

```
24         b.drives();
25     }
26 }
```

二次分发 (double dispatching) , 即对输出消息的请求被分发两次。

构造方法并不具有多态性, 但仍然非常有必要理解构造方法如何在复杂的分级结构中随同多态性一同使用的情况。