让java实现多重继承

在java中，一个类可以多重实现，但不能多重继承，也就是说，一个类可以实现多个接口，但不能同时继承多个类。但有时候我们确实需要继承多个类。比如，希望拥有两个类的行为功能，这就很难使用单继承来解决问题了。但是java又很婉转曲折的提供了两种方法让我们实现多重继承的功能：接口和内部类。

1. 接口

这是我在学习《java语言程序设计----基础篇》中学到的实现多重继承的第一种方法。虽然子类只能继承一个父类，但是却可以实现多个接口，这就为实现多重继承铺好了路。

例如下面的例子：

interface Sleep{

void sleep();

}

interface Eat{

void eat();

}

public class Father{

public void play();

}

Class Son extends Father implements Sleep, Eat{

Public void sleep(){

}

Public void eat(){

}

}

从上面的例子就可以看出，Son继承了父类可以play，又实现了接口可以sleep和eat，因此就实现了多重继承的功能。

1. 内部类

这是我在《编写高质量java代码》一书中学到的第二种方法。我们可以来看一个案例：

public class Father {

public int strong(){

return 9;

}

}

public class Mother {

public int kind(){

return 8;

}

}

public class Son {

/\*\*

\* 内部类继承Father类

\*/

class Father\_1 extends Father{

public int strong(){

return super.strong() + 1;

}

}

class Mother\_1 extends Mother{

public int kind(){

return super.kind() - 2;

}

}

public int getStrong(){

return new Father\_1().Strong();

}

public int getKind(){

return new Mother\_1().kind();

}

}

public class Test1 {

public static void main(String[] args) {

Son son = new Son();

System.out.println("Son 的Strong：" + son.getStrong());

System.out.println("Son 的kind：" + son.getKind());

}

}

测试的结果是：Son 的Strong：10

Son 的kind：6

儿子继承了父亲，变得比父亲更加强壮，同时也继承了母亲，只不过温柔指数下降了。这里定义了两个内部类，他们分别继承父亲Father类、母亲类Mother类，且都可以非常自然地获取各自父类的行为，这是内部类一个重要的特性：内部类可以继承一个与外部类无关的类，保证了内部类的独立性，正是基于这一点，多重继承才会成为可能。