学号 E21514033 专业 软件工程 姓名 何铭春

实验日期 **2018.5.3**  教师签字 成绩

实验报告

【实验名称】 适配器模式实验

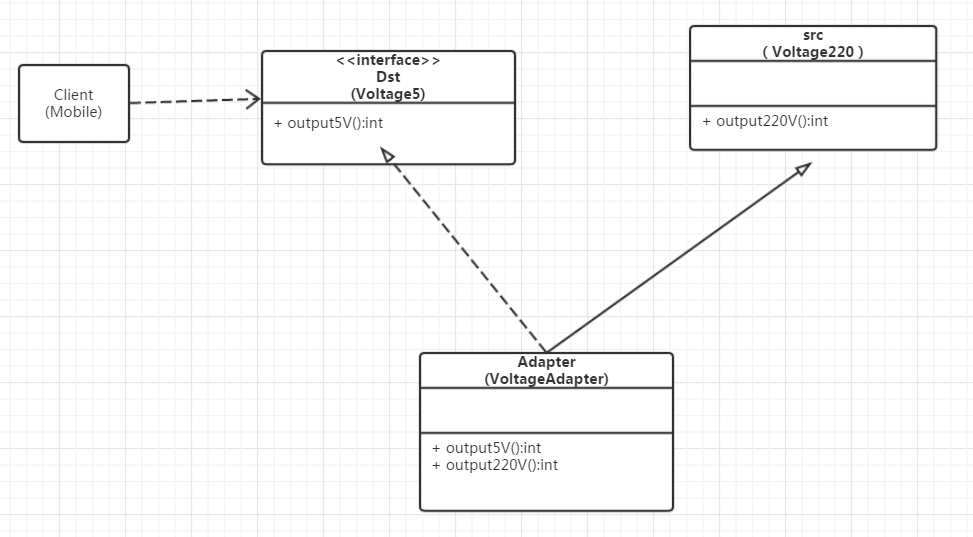
【实验目的】

1. 利用适配器模式将接口转化成所需要的接口。
2. 输出显示当前的接口。

【实验原理】

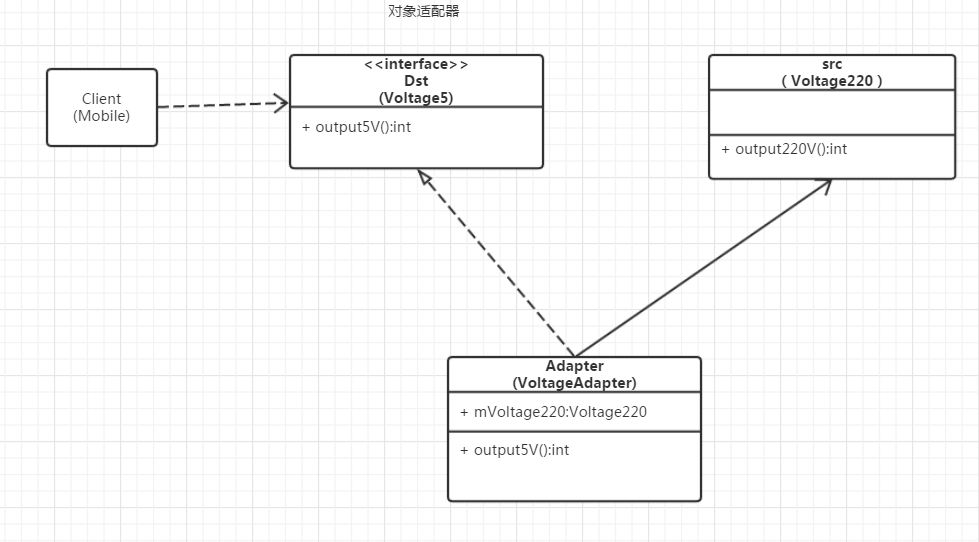
1. 类适配器，适配器继承旧的类，实现新的接口。利用旧的类的方法来实现新的接口。

UML图：



1. 与类适配器最大不同在于，使用旧类的一个对象来调用旧的方法。

UML图：



【实验内容】

USB类：

public class Usb {

public void isUsb(){

System.out.println("当前为USB接口");

}

}

PS2接口：

public interface Ps2 {

abstract void isPs2();

}

类适配器：

public class Adapter1 extends Usb implements Ps2 {

public void isPs2(){

System.out.println("转化前接口");

super.isUsb();

System.out.println("转化后");

System.out.println("适配为Ps2接口模式");

}

}

对象适配器：

public class Adapter2 implements Ps2 {

public Usb from;

public Adapter2(){}

public Adapter2(Usb from){

this.from = from;

}

public void isPs2(){

System.out.println("适配前：");

from.isUsb();

System.out.println("适配后：");

System.out.println("适配后的为USB接口");

}

}

类适配器测试：

public class Test1 {

public static void main(String[] args) {

Ps2 hmc = new Adapter1();

hmc.isPs2();

}

}

对象适配器测试：

public class Test2 {

public static void main(String[] args) {

Usb hmc = new Usb();

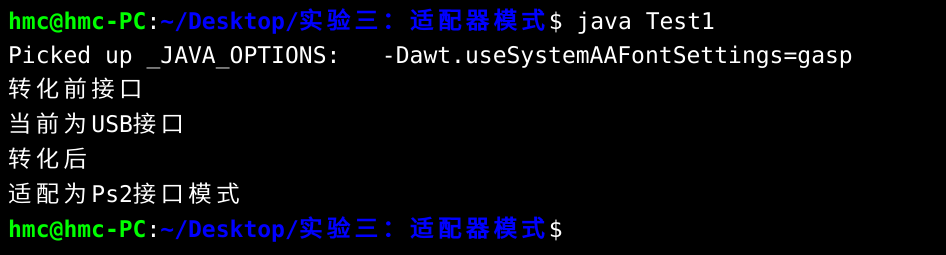
Ps2 spring = new Adapter2(hmc);

spring.isPs2();

}

}

实验截图：



【小结或讨论】

1. 适配器模式目的在于转换接口而外部调用不变。
2. 类适配器优势在于继承可以直接使用旧的方法。但是如果旧类方法臃肿，那么关系将很复杂。
3. 在面对臃肿旧类时，采用对象适配器可以简化关系。