学号 E21514033 专业 软件工程 姓名

实验日期 **2018.5.10**  教师签字 成绩

实验报告

【实验名称】 装饰模式编程实现

【实验目的】

1. 装饰模式学习
2. 掌握装饰模式的原理

【实验原理】

1. 利用装饰模式增加新功能。
2. 能够方便的新增类。

【实验内容】

Car类：所有类的总父类

abstract class Car {

public String name;

public abstract void dotheThing();

}

丰田汽车类：继承于Car类，输出丰田

public class FengTian extends Car {

public FengTian(){

name = "丰田";

}

public void dotheThing(){

System.out.print(name);

}

}

沃尔沃类：同丰田类，但是输出沃尔沃

装饰器父类：继承于Car类，拥有Car的一个对象

public class Decorator extends Car {

public Car hmc;

public void dotheThing(){

System.out.println("装饰父类");

}

}

导航装饰类：继承于装饰器类，增加输出导航

public class NavigationDecorator extends Decorator {

public NavigationDecorator(){}

public NavigationDecorator(Car in){

hmc = in;

}

public void dotheThing(){

hmc.dotheThing();

System.out.print("+导航");

}

}

自动驾驶类和语音识别类同导航类，都增加输出语句。

测试类：输出丰田

public class Test{

public static void main(String[] args) {

Car spring = new AutoDecorator(new NavigationDecorator(new FengTian()));

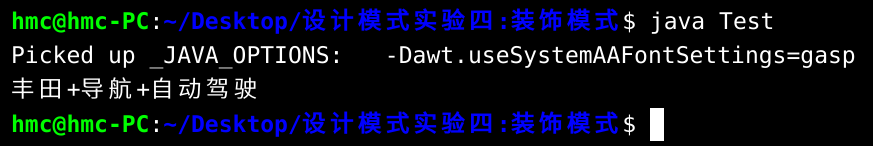
spring.dotheThing();

System.out.println("");

}

}

实验截图：



【小结或讨论】

1. 装饰器模式和对象继承于同一个类，使得装饰器之间能够相互嵌套。
2. 利用该模式能够动态的增加原有类的功能。