|  |
| --- |
| 专业班级 信息173班 学号 201711010228 姓名 苏栋 |

**实验代码：**

package com.hellojava.abstrsct;

//定义抽象类Contents

public abstract class Contents

{

//定义抽象方法value()

abstract public int value();

}

package com.hellojava.abstrsct;

//定义接口Destination

public interface Destination

{

//定义抽象方法readLabel，返回值类型为String，访问修饰符为public

String readLabel();

}

package com.hellojava.abstrsct;

//定义抽象类Wrapping

public abstract class Wrapping

{

//定义抽象方法value

public abstract int value();

}

package 练习;

public class Parcel6 //创建一个名为Pancel6的类

{

public Contents cont()

{

//创建一个cont()方法，返回值为Contents

return new Contents() //实现Contents接口，产生并返回一个引用(匿名内部类)

{

private int i = 11;//给私有的i赋值(类中所包含的属性)

public int value()//创建一个value()方法，返回值为int(类中所包含的方法)

{

return i;

}

};

}

public static void main(String[] args) //主方法

{

Pancel6 p = new Parcel6();//创建一个类型为Parcel6的对象

Contents c = p.cont();//创建一个类型为Contents的对象，值为对象p调用的cont()无参方法所返回的值

}

}

package 练习;

public class Parcel7 //创建一个名为Parcel7的类

{

public Wrapping wrap(int x)//创建一个返回值为Wrapping类型的有参方法

{

return new Wrapping(x)//实现Wrapping接口，产生并返回一个引用的(内部类)

{

public int value()//创建一个value()方法，返回值为int(类中所包含的方法)

{

return super.value()\*47;//返回最终的数值

}

}

}

public static void main(String[] args)

{

Parcel7 p = new Parcel7();//创建一个类型为Parcel7的对象

Wrapping w = p.wrap(10);//创建一个类型为Wrapping的对象，值为对象p调用的wrap()方法所返回的值，传入的参数为7

}

}

package 练习;

public class Parcel8 //创建一个名为Parcel8的类

{

public Destination dest(final String dest)//创建一个返回值为Destination类型的有参方法，参数的值不能改变

{

return new Destination()//实现Destination接口，产生并返回一个引用(匿名内部类)

{

private String label = dest;//给私有的lable复制(将dest的值赋给label)

public String readLabel()//创建一个readLabel()方法，返回值为String类(类中所包含的方法)

{

return label;

}

};

}

public static void main(String[] args)

{

Parcel8 p = new Parcel8();//创建一个类型为Parcel8的对象

Destination d = p.dest("Tanzania");//创建一个类型为Destination的对象，值为对象p调用的dest()方法所返回的值，传入的参数为字符串“Tanzania”

}

}

package 练习;

public class Parcel9 //创建一个名为Parcel9的类

{

public Destination dest(final String dest, final float price)//创建一个返回值为Destination类型的有参方法，参数的值必须被final修饰

{

return new Destination()//实现Destination接口，产生并返回一个引用(匿名内部类)

{

private int cost;//类中所包含的属性

{

cost= Math.round(price);//给属性cost赋值，使用Java库中的函数Math.round(price)四舍五入求价格

if(cost>100)//如果大于100

{

System.out.println("Over budget!");//输出“Over budget！”

}//不能重载

}

private String label = dest;//给私有的label赋值，(将dest的值赋给label)

public String readLabel()//创建一个readLabel()方法，返回值为String类(类中所包含的方法)

{

return label;

}

};

}

public static void main(String[] args)

{

Parcel9 p = new Parcel9();//创建一个类型为Parcel9的对象

Destination d = p.dest("Tanzania", 101.395F);//创建一个类为Destination的对象，值为对象p调用的dest()方法所返回的值，传入的参数为字符串“Tanzania”，和float类型的数字101.395F

}

}