**作业**

任务一：成员内部类

定义一个汽车类，汽车有：

* + - 1. 座位数量
      2. 车轮

颜色

个数

显示车轮信息

* + - 1. 显示汽车信息（）

要求：创建一辆车,展示车的信息。

package day11.homeWork;

class Car{

int seatNum;

class Wheel{

String color;

int num;

public void show() {

System.out.println("车轮是"+color+"的，有"+num+"个车轮");

}

}

public void show() {

System.out.println("车的座位有"+seatNum+"个");

}

}

public class ZuoYe1 {

public static void main(String[] args) {

Car.Wheel wheel = new Car().new Wheel();

wheel.color = "粉色";

wheel.num = 4;

Car car = new Car();

car.seatNum = 5;

car.show();

wheel.show();

}

}

任务二.

创建一个Lambda表达式，无参.该表达式将显示“This is a Lambda expression”字符串。

Interface

Inte i = ()->{System.out.println("This is a Lambda expression");};

任务三：

定义一个Commond接口，里面有一个抽象方法process(int[] target),再定义一个ProcessArray类，里面有一个process(int[] target,Commond commond)方法，方法里面用commond调用process方法，定义测试类，分别使用匿名类和lambda表达式来进行测试，在实现方法中计算数组的总和。

package day11.homeWork;

interface Commond{

int process(int [] target);

}

class ProcessArray{

public int process(int[] target,Commond commond) {

return commond.process(target);

}

}

public class ZuoYe3 {

public static void main(String[] args) {

int [] target ={1,2,3,4,5,6,7};

ProcessArray p = new ProcessArray();

Commond c1 = new Commond() {

@Override

public int process(int[] target) {

int sum = 0;

for (int i : target) {

sum+=i;

}

return sum;

}

};

Commond c2 = (a)->{int sum=0;for (int i : a) {sum+=i;}return sum;};

int sum1 = p.process(target, c1);

int sum2 = p.process(target, c2);

System.out.println("和1为："+sum1);

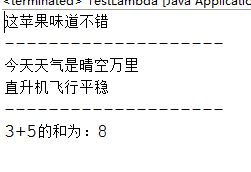
System.out.println("和2为："+sum2);

}

}

任务四：

定义三个接口，里面的方法分别是taste()，drive(String wheather),add(int a,int b)，定义一个类，将接口作为对象传入，在测试类中使用匿名类和lambda表达式来输出如下图所示的结果



package day11.homeWork;

interface Inte1 {

void taste();

}

interface Inte2 {

void drive(String wheather);

}

interface Inte3 {

void add(int a, int b);

}

class InteImpl{

void taste(Inte1 inte1){

inte1.taste();

}

void drive(String wheather,Inte2 inte2) {

inte2.drive(wheather);

}

void add(int a,int b,Inte3 inte3) {

inte3.add(a, b);

}

}

public class ZuoYe4 {

public static void main(String[] args) {

InteImpl impl = new InteImpl();

// 匿名类

Inte1 inte1 = new Inte1() {

@Override

public void taste() {

System.out.println("这苹果味道不错");

}

};

impl.taste(inte1);

// lambda表达式

Inte2 inte2 = (wheather) -> {

System.out.println("今天天气" + wheather);

};

impl.drive("晴空万里", inte2);

// lambda表达式

Inte3 inte3 = (a, b) -> {

System.out.println(a + "+" + b + "的和为：" + (a + b));

};

impl.add(2, 3, inte3);

}

}

任务五：

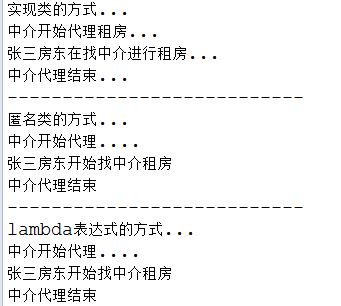
定义租房的接口Rent，里面有一个租房的方法rent()，要求：

1. 定义两个实现类，分别为房东和中介，其中中介为房东代理租房，在测试类中完成测

试

1. 将上面的步骤改写成匿名类的方式，完成测试
2. 将2中的匿名类改写成lambda表达式的方式，并完成测试

运行效果图：



package day11.homeWork;

interface Rent {

void rent();

}

class ZhongJie implements Rent {

@Override

public void rent() {

System.out.println("中介开始代理租房...");

class FangDong implements Rent {

@Override

public void rent() {

System.out.println("张三房东开始找中介租房...");

}

}

new FangDong().rent();

System.out.println("中介代理结束...");

}

}

interface ZhongJie1 extends Rent {

public abstract void rent();

}

interface FangDong1 extends Rent {

public abstract void rent();

}

public class ZuoYe5 {

public static void main(String[] args) {

System.out.println("实现类的方式...");

ZhongJie z1 = new ZhongJie();

z1.rent();

System.out.println("--------------------------");

System.out.println("匿名类的方式...");

FangDong1 f2 = new FangDong1() {

@Override

public void rent() {

System.out.println("张三房东开始找中介租房...");

}

};

ZhongJie1 z2 = new ZhongJie1() {

@Override

public void rent() {

System.out.println("中介开始代理租房...");

f2.rent();

System.out.println("中介代理结束...");

}

};

z2.rent();

System.out.println("--------------------------");

System.out.println("lambda 表达式的方式");

FangDong1 f3 = ()->{System.out.println("张三房东开始找中介租房...");};

ZhongJie1 z3 = () ->{

System.out.println("中介开始代理租房...");

f3.rent();

System.out.println("中介代理结束...");

};

z3.rent();

}

}