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】 Java实验三

【实验目的】

1. 掌握字符串基本使用方法
2. 学会使用数组方法
3. 练习内部类使用

【实验原理】

1. 字符串中split函数用法，注意特殊字符需要转义。
2. 数组使用方法，接收分割后数组。
3. 匿名类用于只使用一次的类。

【实验内容】

题目一：计算两点间距离

Point类：

public class Point {

public double x;

public double y;

public double z;

public Point(){}

public Point(double x,double y,double z){

this.x = x;

this.y = y;

this.z = z;

}

}

测试类：

public class PointDemo {

public static double distance(Point p1,Point p2){

double dis = 0;

dis = dis + ( p1.x - p2.x ) \* ( p1.x - p2.x );

dis = dis + ( p1.y - p2.y ) \* ( p1.y - p2.y );

dis = dis + ( p1.z - p2.z ) \* ( p1.z - p2.z );

dis = Math.sqrt(dis);

return dis;

}

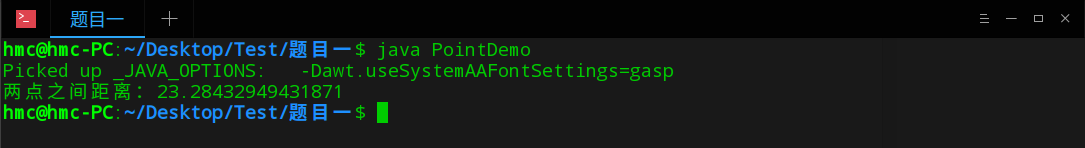
public static void main(String[] args) {

System.out.println("两点之间距离："+distance(new Point(3.5,1.7,9.4),new Point(6.7,23.1,18)));

}

}

实验截图：



题目二：员工类

员工类：

public class Employee {

public int id;

public String name;

public String sex;

public int workAge;

public double wage;

public double transportationAllowance;

public double lunchAllowance;

public double bonus;

public Employee(){}

public Employee(int id,String name,String sex,int workAge,double wage,double transportationAllowance,double lunchAllowance,double bonus){

this.id = id;

this.name = name;

this.sex = sex;

this.workAge = workAge;

this.wage = wage;

this.transportationAllowance = transportationAllowance;

this.lunchAllowance = lunchAllowance;

this.bonus = bonus;

}

public double calculate(){

double sum = 0;

sum = wage + transportationAllowance + lunchAllowance + bonus;

return sum;

}

public String toString(){

return "职工号:"+id+",姓名:"+name+",性别:"+sex+",工龄:"+workAge+",基本工资:"+wage+",交通补贴:"+transportationAllowance+",午餐补助:"+lunchAllowance+",奖金:"+bonus+"。";

}

}

测试类：略

实验截图：



题目三：复数类加法

复数类：

public class FuShu {

private double real;

private double image;

public FuShu(){}

public FuShu(double real,double image){

this.real = real;

this.image = image;

}

public FuShu add(FuShu s1){

double realSum = this.real + s1.real;

double imageSum = this.image + s1.image;

return new FuShu(realSum,imageSum);

}

public void show(){

System.out.println("实部为"+real+",虚部为"+image);

}

}

测试类：

public class Test {

public static void main(String[] args) {

FuShu test1 = new FuShu(12.4,4);

FuShu test2 = new FuShu(2.1,7.3);

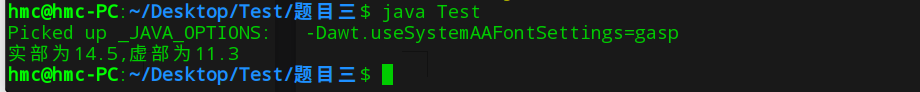
FuShu sum = test1.add(test2);

sum.show();

}

}

实验截图：



题目四：分数类

public class FuShu {

private double real;

private double image;

public FuShu(){}

public FuShu(double real,double image){

this.real = real;

this.image = image;

}

public FuShu add(FuShu s1){

double realSum = this.real + s1.real;

double imageSum = this.image + s1.image;

return new FuShu(realSum,imageSum);

}

public void show(){

System.out.println("实部为"+real+",虚部为"+image);

}

}

测试方法：

public class Test {

public static void main(String[] args) {

FuShu test1 = new FuShu(12.4,4);

FuShu test2 = new FuShu(2.1,7.3);

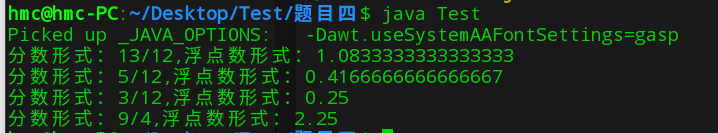
FuShu sum = test1.add(test2);

sum.show();

}

}

实验截图：



题目五：圆锥类中有圆形对象

源代码：

public class Main {

public static void main(String[] args) {

Cone hmc = new Cone(5,4);

System.out.println("圆锥的体积："+hmc.calculate());

}

}

class Circle {

private double r;

public Circle(){}

public Circle(double r){

this.r = r;

}

public double calculate(){

return Math.PI\*r\*r;

}

}

class Cone {

private Circle cc;

private double h;

public Cone(){}

public Cone(double r,double h){

cc = new Circle(r);

this.h = h;

}

public Cone(Circle hmc,double h){

cc = hmc;

this.h = h;

}

public double calculate(){

double s = cc.calculate();

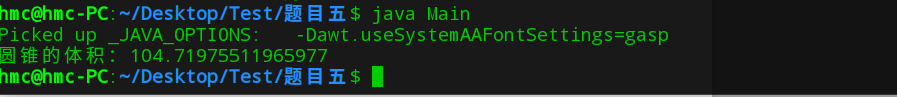
double v = 1.0/3 \* s \* h;

return v;

}

}

实验截图：



题目六：有理数计算器

源代码：

public class Ex {

public static void main(String[] args) {

Fraction f1;

Fraction f2;

Fraction test;

if (args[0].indexOf("+") >= 0) {

String[] hmc = args[0].split("[+]");

String[] spring = hmc[0].split("\\/");

f1 = new Fraction(Integer.parseInt(spring[0]), Integer.parseInt(spring[1]));

spring = hmc[1].split("\\/");

f2 = new Fraction(Integer.parseInt(spring[0]), Integer.parseInt(spring[1]));

test = f1.add(f2);

System.out.println(args[0]+"结果为");

test.show();

}

else if (args[0].indexOf("-") >= 0) {

String[] hmc = args[0].split("[-]");

String[] spring = hmc[0].split("\\/");

f1 = new Fraction(Integer.parseInt(spring[0]), Integer.parseInt(spring[1]));

spring = hmc[1].split("\\/");

f2 = new Fraction(Integer.parseInt(spring[0]), Integer.parseInt(spring[1]));

test = f1.sub(f2);

System.out.println(args[0]+"结果为");

test.show();

}

else if (args[0].indexOf("\*") >= 0) {

String[] hmc = args[0].split("[\*]");

String[] spring = hmc[0].split("\\/");

f1 = new Fraction(Integer.parseInt(spring[0]), Integer.parseInt(spring[1]));

spring = hmc[1].split("\\/");

f2 = new Fraction(Integer.parseInt(spring[0]), Integer.parseInt(spring[1]));

test = f1.mul(f2);

System.out.println(args[0]+"结果为");

test.show();

}

else {

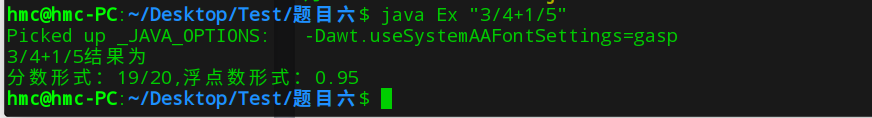
System.out.println("输入有误");

}

}

}

实验截图：



题目七：

Geometry类：

public abstract class Geometry {

public abstract double getArea();

public abstract double getPerimeter();

}

Square类：

public class Square extends Geometry {

private double border;

public Square(){}

public Square(double border){

this.border = border;

}

public double getArea(){

return border \* border;

}

public double getPerimeter(){

return 4 \* border;

}

}

测试类：

public class Test {

public static void main(String[] args) {

Geometry[] hmc = {new Square(6.6),new Geometry(){

private double r = 9.9;

@Override

public double getPerimeter() {

return Math.PI \* 2 \* r;

}

@Override

public double getArea() {

return Math.PI\*r\*r;

}

}};

System.out.println("第一个对象面积："+hmc[0].getArea());

System.out.println("第一个对象面积："+hmc[0].getPerimeter());

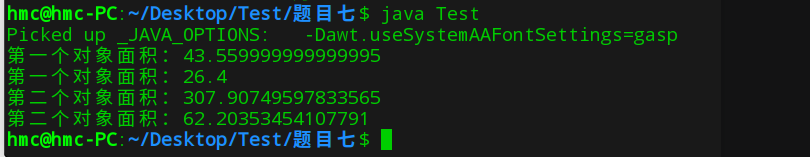
System.out.println("第二个对象面积："+hmc[1].getArea());

System.out.println("第二个对象面积："+hmc[1].getPerimeter());

}

}

实验截图：



【小结或讨论】

1. 不同于Python, java判断字符是否在字符串中使用indexof函数。
2. toString()方法能够直接打印调用。
3. 使用匿名类能够定义只使用一次的类。