**河南工业大学 Java程序设计实验 实验报告**

专业班级： 计科2101 学号： 211040100127 姓名： 翟向阳 指导老师： 赵晨阳 评分：

1. **实验题目**：实验（三）：Java类与对象
2. **实验目的**：1. 理解Java的类；2. 理解Java的对象。
3. **实验要求：1. 每个题目，写出实验思路；2. 代码实现；3. 运行结果截图。**
4. **实验内容**：
5. 求解一元二次方程ax2+bx+c=0的根，要求根据一元二次方程的判别式，分情况显示方程的根。
6. 模拟一个银行的用户行为，包含银行类和账户类：（建议使用带有权限的数据类型，且该题目是开放题目，功能可自定义）
7. 账户类数据成员包括：用户账户名称、用户个人信息、用户的账户余额等。
8. 账户类方法包括：开户、查询余额、取款、存款等。
9. 银行类包含多个账户，建议使用对象数组。
10. 实现测试类进行上述功能测试。
11. 使用面向对象的概念表示出下面的生活场景：小明去超市买东西，所有买到的东西都放在购物车之中，最后到收银台一起结账。（开放题目，功能可自定义）
12. 设计一个名为Point的类，表示一个带x坐标和y坐标的点。要求包括：（1）创建一个默认值为x=0,y=0的无参构造方法；（2）创建一个特定坐标点的构造方法；（3）设置和返回数据x和y的方法；（4）返回Point类型的两个点之间的距离；（5）实现测试类进行上述功能测试。
13. 定义二维形状类（如矩形、三角形、圆形等），这些类具有方法area和perimeter，分别用来计算形状的面积和周长。试定义一个Student类，利用方法重载实现学生求面积和周长（实现多态），并编写测试类验证。
14. 定义Circle2D类，包括：（1）一个无参构造方法，该方法创建一个中心点（x，y），其值为（0,0）且半径为1的默认圆；（2）一个构造方法，创建特定中心点和半径的圆；（3）返回圆面积和周长；（4）contains（Circle2D circle）判定一个圆是否在圆内，并返回布尔型；（5）判定两个圆是否重叠overlaps（Circle2D circle）；（6）测试类测试上述功能。
15. 定义一个表示学生信息的类，要求如下：（1）学生的属性包括：学号、姓名、性别、年龄和课程成绩（5门课，其中包括Java）；（2）创建带参数的构造方法；（3）获取学生的属性信息；（4）根据学生类，创建五个该类的对象，要求对象数组，并打印五个学生的Java课程成绩的平均值，并输出成绩的最大值（分数和学生名）和最小值（分数和学生名）。
16. **实验执行：**

1. 求解一元二次方程ax2+bx+c=0的根，要求根据一元二次方程的判别式，分情况显示方程的根。

（1）思路

根据一元二次方程的判别式（b2-4ac）>0,<0,=0三种情况分别讨论求解

（2）代码实现

System.out.println("请输入a,b,c三个参数的值：");

Scanner reader = new Scanner(System.in);

double a = reader.nextDouble();

double b = reader.nextDouble();

double c = reader.nextDouble();

double dealt=0,x=0;

dealt = b\*b-4\*a\*c;

if(dealt<0){

System.out.println("该方程无解");

}

else if(dealt==0){

System.out.println("有一个实根");

x = (-b+Math.sqrt(dealt))/(2\*a);

System.out.println("x = "+x);

}

else {

System.out.println("有两个实根");

double x1,x2;

x1 = (-b+Math.sqrt(dealt))/(2\*a);

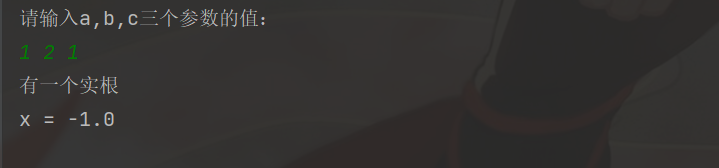
x2 = (-b-Math.sqrt(dealt))/(2\*a);

System.out.println("x1 = "+x1);

System.out.println("x2 = "+x2);

}

（3）运行结果截图



2. 模拟一个银行的用户行为，包含银行类和账户类：（建议使用带有权限的数据类型，且该题目是开放题目，功能可自定义）

（1）账户类数据成员包括：用户账户名称、用户个人信息、用户的账户余额等。

（2）账户类方法包括：开户、查询余额、取款、存款等。

（3）银行类包含多个账户，建议使用对象数组。

（4）实现测试类进行上述功能测试。

1. 先创建一个基本的客户类，客户类里边包含客户的各种信息（电话，姓名…）,包含取钱和存钱的操作等，接着创建一个银行类，银行类里有一个客户的对象数组，在银行类里调用客户类里的接口函数
2. 代码实现

import java.util.Scanner;

import static java.lang.System.exit;

public class test2 {

public static void main(String[] args){

Account a =new Account();

Scanner sc =new Scanner(System.in);

while (true){

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

System.out.println("1.开户");

System.out.println("2.存钱");

System.out.println("3.取钱");

System.out.println("4.显示信息");

System.out.println("0.退出");

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

System.out.println("输入你的选择：");

int p = sc.nextInt();

switch (p){

case 1:

a.creat();

break;

case 2:

a.save();

break;

case 3:

a.get();

break;

case 4:

a.showall();

break;

case 0:

exit(0);

}

}

}

}

class Account {

static int MAX = 10100;//最大用户容量

static int i = 10001;

private Custumer[] custumers = new Custumer[10100];//所属客户

Scanner sc = new Scanner(System.in);

public void creat() {

if (i >= MAX)

System.out.println("用户已经达到上限，禁止开户");

else {

custumers[i] = new Custumer();

custumers[i].ID = i;

System.out.println("姓名：");

String m = sc.next();

custumers[i].setName(m);

System.out.println("电话：");

String tel = sc.next();

custumers[i].setTel(tel);

System.out.println("密码：");

int password = sc.nextInt();

custumers[i].setPassword(password);

custumers[i].setBalance(0);

i++;

}

}

public void save(){

System.out.println("输入你的银行卡号：");

int i = sc.nextInt();

if(i<=custumers.length+10001)

custumers[i].saveMoney();

else {

System.out.println("所输入的用户不存在，请重新输入");

return;

}

}

public void get(){

System.out.println("请输入你的银行卡号：");

int i = sc.nextInt();

if(i<=custumers.length+10001)

custumers[i].GetMoney();

else {

System.out.println("所输入的用户不存在，请重新输入");

return;

}

}

public void showall(){

System.out.println("请输入你的银行卡号：");

int id = sc.nextInt();

System.out.println("请输入你的密码：");

int passward2 = sc.nextInt();

if(passward2!=custumers[id].GetPassword()){

System.out.println("密码输入错误，请重新输入：");

}

else {

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

System.out.println("银行卡号："+custumers[id].GetID());

System.out.println("姓名："+custumers[id].GetName());

System.out.println("电话号码："+custumers[id].GetTel());

System.out.println("余额："+custumers[id].GetBalance());

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

}

}

}

class Custumer{

private String name;

private String tel;

public int ID;//账户

private double balance;//余额

private int password;

public Custumer(){

}

public Custumer(String name,String tel,int ID){

this.ID=ID;

this.tel=tel;

this.name=name;

}

public void setName(String m){

name = m;

}

public void setTel(String t){

tel = t;

}

public void setPassword(int p){

password=p;

}

public void setBalance(double b){

balance=b;

}

public double GetBalance(){

return this.balance;

}

public int GetID(){

return this.ID;

}

public int GetPassword(){

return this.password;

}

public String GetName()

{

return this.name;

}

public String GetTel(){

return this.tel;

}

//存钱

public void saveMoney(){

System.out.println("请输入您存的钱：");

Scanner sc = new Scanner(System.in);

double money = sc.nextDouble();

setBalance(GetBalance()+money);

System.out.println(GetID()+"存入:"+money);

System.out.println("现有余额:"+GetBalance());

}

//取钱

public void GetMoney( ){

Scanner sc = new Scanner(System.in);

System.out.println("请输入你的密码：");

int password1 = sc.nextInt();

System.out.println("请输入你的取钱数额：");

double money = sc.nextDouble();

if(GetPassword()!=password1){

System.out.println("您输入的密码错误请重新输入");

}

else {

if(money>GetBalance()){

System.out.println("余额不足");

}

else {

setBalance(GetBalance()-money);

System.out.println("取钱后："+GetBalance());

}

}

}

}

（3）

--------------------------------

1.开户

2.存钱

3.取钱

4.显示信息

0.退出

--------------------------------

输入你的选择：

1

姓名：

zxy

电话：

123456

密码：

123456

--------------------------------

1.开户

2.存钱

3.取钱

4.显示信息

0.退出

--------------------------------

输入你的选择：

4

请输入你的银行卡号：

10001

请输入你的密码：

123456

--------------------------------

银行卡号：10001

姓名：zxy

电话号码：123456

余额：0.0

--------------------------------

--------------------------------

1.开户

2.存钱

3.取钱

4.显示信息

0.退出

--------------------------------

输入你的选择：

2

输入你的银行卡号：

10001

请输入您存的钱：

200

10001存入:200.0

现有余额:200.0

--------------------------------

1.开户

2.存钱

3.取钱

4.显示信息

0.退出

--------------------------------

输入你的选择：

3

请输入你的银行卡号：

10001

请输入你的密码：

123456

请输入你的取钱数额：

100

取钱后：100.0

--------------------------------

1.开户

2.存钱

3.取钱

4.显示信息

0.退出

--------------------------------

输入你的选择：

4

请输入你的银行卡号：

10001

请输入你的密码：

123456

--------------------------------

银行卡号：10001

姓名：zxy

电话号码：123456

余额：100.0

--------------------------------

--------------------------------

1.开户

2.存钱

3.取钱

4.显示信息

0.退出

--------------------------------

输入你的选择：

0

进程已结束,退出代码0）

1. 使用面向对象的概念表示出下面的生活场景：小明去超市买东西，所有买到的东西都放在购物车之中，最后到收银台一起结账。（开放题目，功能可自定义）

（1）

（2）

public class test3 {

public static void main(String[] args) {

gouwuche g1 = new gouwuche();

g1.add(new clothes("羽绒服",66.9));

g1.add(new drink("可乐",3.0));

g1.add(new food("口水鸡",13));

System.out.println("您购买的数量是："+g1.getNumber());

g1.GetMoney();

}

}

abstract class Goods{

abstract public String getName();

abstract public double getPrice();

}

class clothes extends Goods{//衣服

private String name;

private double price;

clothes(String name,double price) {

this.name = name;

this.price=price;

}

public String getName(){

return this.name;

}

public double getPrice() {

return this.price;

}

}

class drink extends Goods{

private String name;

private double price;

drink(String name,double price) {

this.name = name;

this.price=price;

}

public String getName(){

return this.name;

}

public double getPrice() {

return this.price;

}

}

class food extends Goods{

private String name;

private double price;

food(String name,double price) {

this.name = name;

this.price=price;

}

public String getName(){

return this.name;

}

public double getPrice() {

return this.price;

}

}

class gouwuche{

Goods [] goods;

private int number;//物品数量

public static int max=10;//最大的购物数量

gouwuche( ){

this.goods=new Goods[max];

}

public void add(Goods goods){

if(this.goods.length>max){

System.out.println("所拿物品数量已达到最大，装不下辣！");

}

else {

this.goods[number] = goods;

number++;

}

}

public void GetMoney(){

double sum=0;

System.out.println("以下是您的详细购买信息：");

for(int i=0;i<this.number;i++){

sum+=this.goods[i].getPrice();

System.out.println("name:"+this.goods[i].getName()+"\t\tprice:"+this.goods[i].getPrice());

}

System.out.println("总金额："+sum);

}

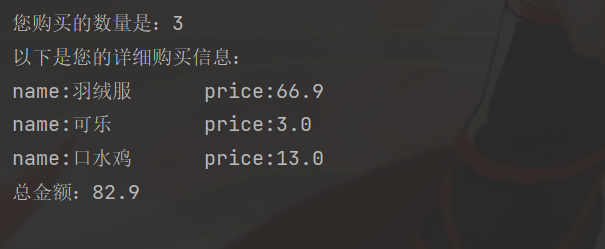
public int getNumber(){

return this.number;

}

}

（3）



4. 设计一个名为Point的类，表示一个带x坐标和y坐标的点。要求包括：（1）创建一个默认值为x=0,y=0的无参构造方法；（2）创建一个特定坐标点的构造方法；（3）设置和返回数据x和y的方法；（4）返回Point类型的两个点之间的距离；（5）实现测试类进行上述功能测试

（1）按照上述要求操作就好

（2）

import java.util.Scanner;

import static java.lang.Math.sqrt;

public class test4 {

public static void main(String[] args) {

System.out.println("输入两个点的坐标：a,b");

Scanner sc = new Scanner(System.in);

double x1 = sc.nextDouble();

double y1 = sc.nextDouble();

double x2 = sc.nextDouble();

double y2 = sc.nextDouble();

Point a = new Point(x1,y1);

Point b = new Point(x2,y2);

double dis = b.distance(a);

System.out.print("两个点的距离：");

System.out.println(dis);

}

}

class Point{

private double x;

private double y;

Point(){

this.x=0;

this.y=0;

}

Point(double x,double y){

this.x=x;

this.y=y;

}

public double getX() {

return x;

}

public double getY(){

return y;

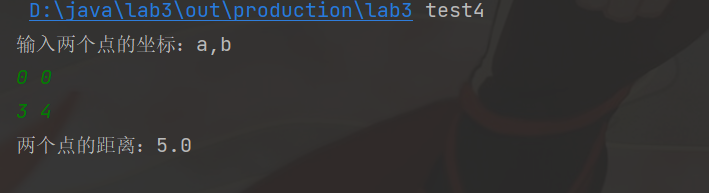
}

public double distance(Point a){

return sqrt((this.getX()- a.getX())\*(this.getX() - a.getX())+(this.getY() - a.getY())\*(this.getY() - a.getY()));

}

}



5. 定义二维形状类（如矩形、三角形、圆形等），这些类具有方法area和perimeter，分别用来计算形状的面积和周长。试定义一个Student类，利用方法重载实现学生求面积和周长（实现多态），并编写测试类验证。

（1）

利用重载实现多态，通过传的实参不同调用不同的同名函数来实现多态

（2）

public class test5 {

public static void main(String[] args) {

student stu = new student();

Circle c =new Circle(2);

sanjiao s = new sanjiao(3,4,5);

juxin j = new juxin(2,3);

stu.area(c);

stu.area(s);

stu.area(j);

stu.peirmeter(c);

stu.peirmeter(s);

stu.peirmeter(j);

}

}

class student{

public void area(Circle c){

System.out.println("计算出圆的面积："+c.getArea());

}

public void peirmeter(Circle c){

System.out.println("计算出圆的周长："+c.getperimeter());

}

public void area(sanjiao c){

System.out.println("计算出三角形的面积："+c.getArea());

}

public void peirmeter(sanjiao c){

System.out.println("计算出三角形的周长："+c.getperimeter());

}

public void area(juxin c){

System.out.println("计算出矩形的面积："+c.getArea());

}

public void peirmeter(juxin c){

System.out.println("计算出矩形的周长："+c.getperimeter());

}

}

class Circle{

private double r;

Circle(int r){

this.r=r;

}

Circle(){}

public void setR(double r) {

this.r = r;

}

double getArea(){

return 3.14\*r\*r;

}

double getperimeter(){

return 2\*3.14\*r;

}

}

class sanjiao{

double a,b,c;

sanjiao(double a,double b,double c){

this.a=a;

this.b=b;

this.c=c;

}

double getArea(){

double p = getperimeter()/2;

return Math.sqrt(p\*(p-a)\*(p-b)\*(p-c));

}

double getperimeter(){

return a+b+c;

}

}

class juxin{

double a,b;

juxin(double a,double b){

this.a=a;

this.b=b;

}

double getArea(){

return a\*b;

}

double getperimeter(){

return (a+b)\*2;

}

}

（3）



6. 定义Circle2D类，包括：（1）一个无参构造方法，该方法创建一个中心点（x，y），其值为（0,0）且半径为1的默认圆；（2）一个构造方法，创建特定中心点和半径的圆；（3）返回圆面积和周长；（4）contains（Circle2D circle）判定一个圆是否在圆内，并返回布尔型；（5）判定两个圆是否重叠overlaps（Circle2D circle）；（6）测试类测试上述功能。

（1）

根据题目要求做就好了

（2）

import static java.lang.Math.sqrt;

public class test6 {

public static void main(String[] args) {

Circle2D a = new Circle2D(0,0,2);

Circle2D b =new Circle2D(0,0,1);

System.out.println("a的面积："+a.getArea());

System.out.println("b的面积："+b.getArea());

System.out.println("a的周长："+a.getPerimeter());

System.out.println("b的周长："+b.getPerimeter());

System.out.println("a是否在b内："+a.contains(b));

System.out.println("a是否与b重合："+a.overlaps(b));

}

}

class Circle2D{

int x,y;

int r;

Circle2D(){

x=0;

y=0;

r=1;

}

Circle2D(int x,int y,int r){

this.r=r;

this.y=y;

this.x=x;

}

public int getX() {

return x;

}

public int getY() {

return y;

}

public double getArea(){

return 3.14\*r\*r;

}

public double getPerimeter(){

return 2\*r\*3.14;

}

public boolean contains(Circle2D circle) {

double distance = sqrt((this.getX() - circle.getX()) \* (this.getX() - circle.getX()) + (this.getY() - circle.getY()) \* (this.getY() - circle.getY()));

return distance + circle.r < this.r;

}

public boolean overlaps(Circle2D circle) {

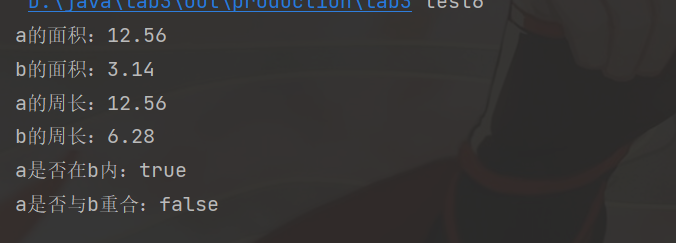
double distance = sqrt((this.getX() - circle.getX()) \* (this.getX() - circle.getX()) + (this.getY() - circle.getY()) \* (this.getY() - circle.getY()));

return distance == 0 && this.r == circle.r;

}

}

（3）



7. 定义一个表示学生信息的类，要求如下：（1）学生的属性包括：学号、姓名、性别、年龄和课程成绩（5门课，其中包括Java）；（2）创建带参数的构造方法；（3）获取学生的属性信息；（4）根据学生类，创建五个该类的对象，要求对象数组，并打印五个学生的Java课程成绩的平均值，并输出成绩的最大值（分数和学生名）和最小值（分数和学生名）。

（1）

根据要求写即可

（2）

public class test7 {

public static void main(String[] args) {

calculate c = new calculate();

for(int i=0;i<5;i++){

c.students[i].showall();

}

System.out.println("这五个学生的java平均值："+c.javaAverage());

System.out.println("java成绩的最大值最小值：");

c.showMaxMin();

}

}

class calculate{

Student[] students=new Student[5];

calculate(){

students[0] = new Student(1,"a","男",16,80,90,60,75,50);

students[1] = new Student(2,"b","女",15,60,80,70,88,80);

students[2] = new Student(3,"c","男",17,90,80,58,79,89);

students[3] = new Student(4,"d","男",18,90,58,64,76,84);

students[4] = new Student(5,"e","男",19,65,78,75,67,50);

}

public double javaAverage(){

return (students[0].getJava()+students[1].getJava()+students[2].getJava()+students[3].getJava()+students[4].getJava())/5;

}

public void showMaxMin(){

double Max,Min;

int a = 0,b=0;//a表示max的位置，b表示min的位置

Max=0;

Min = students[0].getJava();

for(int i=0;i<5;i++){

if(students[i].getJava()>Max){

Max=students[i].getJava();

a=i;

}

if(students[i].getJava()<Min){

Min=students[i].getJava();

b=i;

}

}

System.out.println("最大值："+students[a].getName()+" "+Max);

System.out.println("最小值："+students[b].getName()+" "+Min);

}

}

class Student{

private int ID;

private String name;

private String sex;

private int age;

private double math,chinese,english,physics,java;

Student(int ID,String name,String sex,int age,double math,double chinese,double english,double java,double physics){

this.age=age;

this.ID=ID;

this.name=name;

this.sex=sex;

this.math=math;

this.english=english;

this.chinese=chinese;

this.java=java;

this.physics=physics;

}

Student(){}

public String getName() {

return name;

}

public int getID() {

return ID;

}

public String getSex() {

return sex;

}

public int getAge() {

return age;

}

public double getJava() {

return java;

}

public void showall(){

System.out.println("输出所有信息：");

System.out.println("学号："+this.getID());

System.out.println("姓名："+this.getName());

System.out.println("性别："+this.getSex());

System.out.println("年龄："+this.getAge());

System.out.println("math\tchinese\tenglish\tphysics\tjava");

System.out.println(this.math+"\t"+this.chinese+"\t"+this.english+"\t"+this.physics+"\t"+this.java);

}

}

（3）

输出所有信息：

学号：1

姓名：a

性别：男

年龄：16

math chinese english physics java

80.0 90.0 60.0 50.0 75.0

输出所有信息：

学号：2

姓名：b

性别：女

年龄：15

math chinese english physics java

60.0 80.0 70.0 80.0 88.0

输出所有信息：

学号：3

姓名：c

性别：男

年龄：17

math chinese english physics java

90.0 80.0 58.0 89.0 79.0

输出所有信息：

学号：4

姓名：d

性别：男

年龄：18

math chinese english physics java

90.0 58.0 64.0 84.0 76.0

输出所有信息：

学号：5

姓名：e

性别：男

年龄：19

math chinese english physics java

65.0 78.0 75.0 50.0 67.0

这五个学生的java平均值：77.0

java成绩的最大值最小值：

最大值：b 88.0

最小值：e 67.0

1. **实验测试中遇到的问题和自己的解决方案**

在实验二的时候，刚开始实在是不知所措，有点不知道如何了，后来分布去实现，一点一点实现部分，最后实现出来了，有点像一个小的课设了感觉，中间的几个倒是没有什么报错的问题，按步骤来就能实现出来了，但是实验7一直报错，报的错还看不懂，弄了好长时间，最后发现原来是student类的命名跟之前一个实验中的一个类名重名了，类名好像不区分大小写，导致一直无法运行，我还以为是我的对象数组创建错误了，并没有分配内存空间，后来看了看报错信息 wrong name …… ,才试一下更改类名，试了一下运行成功。