**《Java语言程序设计》课程实验报告12**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **专业名称** | 计算机科学与技术 | **年级** | 2017 | **班级** | 计2 |
| **学生姓名** | 王汝芸 | **指导老师** | 李焱 | **时间** | 2019.05.09 |

|  |  |
| --- | --- |
| 实验名称 | 面向对象 |
| 实验 目 的 及 要 求 | **目的**：  了解熟悉Java程序设计的形式，编写完整Java程序。  **要求**：   * 掌握类的设计； * 掌握类之间的关系； * 掌握用类解决问题； * 注意避免易犯的错误； * 会用Java编写完整的程序。 |
| 实 验 环 境 | Microsoft Windows 10 教育版64位  JDK 1.8.0\_201  IntelliJ IDEA Community Edition 2018.3.4 |
| 实 验 内 容 | 请按照要求编写出完整程序   * 第9章编程练习题(P308): 9.10, 9.12, 9.13 * 第10章编程练习题(P340):10.4-5 |
| 实 验 步 骤 或 实 验 方 案 | **课后题9.10**  01 **package** sndu.wry;  02  03 **public class** Main {  04 **public static void** **main**(String[] args) {  05 QuadraticEquation quadraticEquation = **new** **QuadraticEquation**(Math.**random**(),Math.**random**(),Math.**random**());  06 System.out.**println**("随机生成参数如下");  07 System.out.**println**("a="+quadraticEquation.**getA**()+",b="+quadraticEquation.**getB**()+",c="+quadraticEquation.**getC**());  08 **if**(quadraticEquation.**getDiscriminant**()>0){  09 System.out.**println**("判别式大于0，方程有两实数根：");  10 System.out.**println**(quadraticEquation.**getRoot1**());  11 System.out.**println**(quadraticEquation.**getRoot2**());  12 }  13 **else if**(quadraticEquation.**getDiscriminant**()==0){  14 System.out.**println**("判别式等于0，方程有两相等实数根");  15 System.out.**println**(quadraticEquation.**getRoot1**());  16 }  17 **else** {  18 System.out.**println**("判别式小于零，方程无实数根");  19 }  20  21  22 }  23  24 }  01 **package** sndu.wry;  02  03 **public class** QuadraticEquation {  04 **private double** a,b,c;  05 **QuadraticEquation**(**double** a,**double** b,**double** c){  06 **this**.a=a;  07 **this**.b=b;  08 **this**.c=c;  09 }  10  11 **public double** **getA**() {  12 **return** a;  13 }  14  15 **public double** **getB**() {  16 **return** b;  17 }  18  19 **public double** **getC**() {  20 **return** c;  21 }  22 **double** **getDiscriminant**(){  23 **return** b\*b-4\*a\*c;  24 }  25 **double** **getRoot1**(){  26 **return** (-b+Math.**sqrt**(**getDiscriminant**()))/(2\*a);  27 }  28 **double** **getRoot2**(){  29 **return** (-b-Math.**sqrt**(**getDiscriminant**())/(2\*a));  30 }  31  32  33 }  **课后题9.12**  01 **package** sndu.wry;  02  03 **public class** Main {  04 **public static void** **main**(String[] args) {  05 System.out.**println**("随机生成四个点");  06 LinearEquation linearEquation = **new** **LinearEquation**();  07 linearEquation.**setA**(**new** **Point**((**int**)(Math.**random**()\*10),(**int**)(Math.**random**()\*10)));  08 linearEquation.**setB**(**new** **Point**((**int**)(Math.**random**()\*10),(**int**)(Math.**random**()\*10)));  09 linearEquation.**setC**(**new** **Point**((**int**)(Math.**random**()\*10),(**int**)(Math.**random**()\*10)));  10 linearEquation.**setD**(**new** **Point**((**int**)(Math.**random**()\*10),(**int**)(Math.**random**()\*10)));  11 System.out.**println**("A("+linearEquation.**getA**().**getX**()+","+linearEquation.**getA**().**getY**()+")");  12 System.out.**println**("B("+linearEquation.**getB**().**getX**()+","+linearEquation.**getB**().**getY**()+")");  13 System.out.**println**("C("+linearEquation.**getC**().**getX**()+","+linearEquation.**getC**().**getY**()+")");  14 System.out.**println**("D("+linearEquation.**getD**().**getX**()+","+linearEquation.**getD**().**getY**()+")");  15  16 System.out.**println**("交点为：("+linearEquation.**getCrossPoint**().**getX**()+","+linearEquation.**getCrossPoint**().**getY**()+")");  17 }  18  19 }  01 **package** sndu.wry;  02  03 **public class** Point {  04 **private double** x,y;  05 **Point**(**double** x,**double** y){  06 **this**.x = x;  07 **this**.y = y;  08 }  09 **Point**(){  10  11 }  12  13 **public double** **getX**() {  14 **return** x;  15 }  16  17 **public void** **setX**(**double** x) {  18 **this**.x = x;  19 }  20  21 **public double** **getY**() {  22 **return** y;  23 }  24  25 **public void** **setY**(**double** y) {  26 **this**.y = y;  27 }  28 }  01 **package** sndu.wry;  02  03 **public class** LinearEquation {  04 **private** Point a,b;  05 **private** Point c,d;  06  07 **public void** **setA**(Point a) {  08 **this**.a = a;  09 }  10  11 **public void** **setB**(Point b) {  12 **this**.b = b;  13 }  14  15 **public void** **setC**(Point c) {  16 **this**.c = c;  17 }  18  19 **public void** **setD**(Point d) {  20 **this**.d = d;  21 }  22  23 **public** Point **getA**() {  24 **return** a;  25 }  26  27 **public** Point **getB**() {  28 **return** b;  29 }  30  31 **public** Point **getC**() {  32 **return** c;  33 }  34  35 **public** Point **getD**() {  36 **return** d;  37 }  38  39 Point **getCrossPoint**(){  40 Point point = **new** **Point**();  41 **double** k1 = (**getA**().**getY**()-**getB**().**getY**())/(**getA**().**getX**()-**getB**().**getX**());  42 **double** k2 = (**getC**().**getY**()-**getD**().**getY**())/(**getC**().**getX**()-**getD**().**getX**());  43 *// x=(k1\*x0-k2\*x2+y2-y0)/(k1-k2);*  44 point.**setX**((k1\***getA**().**getX**()-k2\***getC**().**getX**()+**getC**().**getY**()-**getA**().**getY**())/(k1-k2));  45 point.**setY**(**getA**().**getY**()+(point.**getX**()-**getA**().**getX**())\*k1);  46  47 *// y=y0+(x-x0)\*k1;*  48 **return** point;  49 }  50 }  **课后题9.13**  01 **package** sndu.wry;  02  03 **public class** Main {  04 **public static void** **main**(String[] args) {  05 **double** [][] a = **new double**[3][4];  06 System.out.**println**("Ëæ»úÉú³ÉÈçÏÂÊý×é");  07 **for** (**int** i = 0;i<3;i++){  08 **for**(**int** j=0;j<4;j++){  09 a[i][j] = Math.**random**();  10 System.out.**printf**("%.2f\t",a[i][j]);  11 }  12 System.out.**print**("\n");  13 }  14 System.out.**printf**("Êý×é×î´óÖµ£º%.2f,Î»ÖÃÎª:(%d,%d)\n",Location.**locateLargest**(a).maxValue,Location.**locateLargest**(a).row,Location.**locateLargest**(a).column);  15 }  16 }  01 **package** sndu.wry;  02  03 **public class** Location {  04 **public int** row=0,column=0;  05 **public double** maxValue;  06  07 **public static** Location **locateLargest**(**double**[][] a){  08 Location location = **new** **Location**();  09 location.maxValue = a[0][0];  10  11 **for**(**int** i=0;i<3;i++){  12 **for**(**int** j=0;j<4;j++){  13 **if**(location.maxValue<=a[i][j]){  14 location.maxValue = a[i][j];  15 location.row = i;  16 location.column = j;  17 }  18 }  19 }  20 **return** location;  21 }  22 }  **课后题10.4**  01 **package** sndu.wry;  02 **public class** Main {  03 **public static void** **main**(String[] args) {  04 MyPoint p0 = **new** **MyPoint**();  05 MyPoint p1 = **new** **MyPoint**(10,30.5);  06 System.out.**println**("MyPoint调用："+p1.**distance**(p0));  07 System.out.**println**("传值调用："+p1.**distance**(0,0));  08 }  09 }  01 **package** sndu.wry;  02  03 **public class** MyPoint {  04 **private double** x,y;  05  06 **MyPoint**(){  07 **this**.x = 0;  08 **this**.y = 0;  09 }  10 **MyPoint**(**double** x,**double** y){  11 **this**.x = x;  12 **this**.y = y;  13 }  14  15 **double** **distance**(MyPoint t){  16 **return** Math.**sqrt**(Math.**pow**((t.**getX**()-**this**.x),2)+Math.**pow**((t.**getY**()-**this**.y),2));  17 }  18  19 **double** **distance**(**double** x,**double** y){  20 **return** Math.**sqrt**(Math.**pow**((x-**this**.x),2)+Math.**pow**((y-**this**.y),2));  21 }  22  23 **public double** **getX**() {  24 **return** x;  25 }  26  27 **public double** **getY**() {  28 **return** y;  29 }  30  31  32 }  **课后题10.5**  01 **package** sndu.wry;  02 **public class** Main {  03 **public static void** **main**(String[] args) {  04 StackOfIntegers stackOfIntegers = **new** **StackOfIntegers**();  05 System.out.**println**(stackOfIntegers.**findElement**(120));  06 }  07 }  01 **package** sndu.wry;  02  03 **import** java.util.ArrayList;  04 **import** java.util.List;  05  06 **public class** StackOfIntegers {  07 **private** List<Integer> element = **new** ArrayList<>();  08  09 **public** List **findElement**(**int** num){  10 **for**(**int** i=2;;i++){  11 **if**(num%i==0){  12 **this**.element.**add**(i);  13 num=num/i;  14 i=1;  15 }  16 **if**(num==1){  17 **break**;  18 }  19 }  20 **return this**.element;  21 }  22 } |
| 调 试 过 程 及 实 验 结  果 | **课后题9.10**    **课后题9.12**    **课后题9.13**    **课后题10.4**    **课后题10.5** |
| 总 结 | 无 |
| 附 录 | Github源码地址： |