**《Java语言程序设计》课程实验报告10**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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** | 2019.04.25 |

|  |  |
| --- | --- |
| 实验名称 | 数组2 |
| 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**：   * 掌握数组的存储形式； * 掌握数组的定义与声明； * 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 家庭中文版（简体中文）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8章编程练习题(P258): 8.1-2, 8.4, 8.6, 8.8，8.11，8.15, 8.35-36 |
| 实 验 步 骤 或 实 验 方 案 | **课后题8.1**  01 **package** sdnu.wry.demo;  02 **import** java.util.\*;  03  04 **public class** Homework{  05 **public static void** **main**(String[] args) {  06 **int** [][] num = **new int**[3][4];  07 *//初始化矩阵，随机元素赋值*  08 **for**(**int** i=0;i<3;i++){  09 **for**(**int** j=0;j<4;j++){  10 num[i][j]=(**int**)(Math.**random**()\*10+1);  11 }  12 }  13 *//打印矩阵*  14 **for**(**int** i=0;i<3;i++){  15 **for**(**int** j=0;j<4;j++){  16 System.out.**print**(num[i][j]+"\t");  17 }  18 System.out.**print**("\n");  19 }  20  21 **for**(**int** i=0;i<4;i++){  22 System.out.**println**("第"+(i+1)+"列数求和为："+**sumColumn**(num,i));  23 }  24  25 }  26  27 **public static int** **sumColumn**(**int**[][] num,**int** columnIndex){  28 **int** sum=0;  29 **for**(**int** row=0;row<3;row++){  30 sum+=num[row][columnIndex];  31  32 }  33 **return** sum;  34 }  35  36 }  **课后题8.2**  01 **package** sdnu.wry.demo;  02 **import** java.text.DecimalFormat;  03 **import** java.util.\*;  04  05 **public class** Homework{  06 **public static void** **main**(String[] args) {  07 DecimalFormat df = **new** **DecimalFormat**("#.00");  08 **double**[][] num = **new double**[4][4];  09  10 *//随机生成数组并打印*  11 **for**(**int** i=0;i<4;i++){  12 **for**(**int** j=0;j<4;j++){  13 num[i][j]=(Double.**valueOf**(df.**format**(Math.**random**()\*2)));  14 System.out.**print**(num[i][j]+"\t");  15 }  16 System.out.**println**();  17 }  18 System.out.**printf**("主对角求和为：%.2f",**sumMajorDiagonal**(num));  19  20  21  22  23 }  24 **public static double** **sumMajorDiagonal**(**double**[][]num){  25 **double** sum=0;  26 **for**(**int** i=0;i<4;i++){  27 sum+=num[i][i];  28 }  29 **return** sum;  30 }  31  32 }  **课后题8.4**  01 **package** sdnu.wry.demo;  02 **import** java.util.\*;  03  04 **public class** Homework{  05 **public static void** **main**(String[] args) {  06 **int**[][] num = **new int**[7][7];  07  08 System.out.**println**("随机生成数组");  09 *//随机生成数组并打印*  10 **for**(**int** i=0;i<7;i++){  11 **for**(**int** j=0;j<7;j++){  12 num[i][j]=(**int**)(Math.**random**()\*9+1);  13 System.out.**print**(num[i][j]+"\t");  14 }  15 System.out.**println**();  16 }  17  18 **int**[] sumTime = **sortRow**(num);*//每行总工时数*  19  20 System.out.**println**();  21 **int** max=sumTime[0],max\_index=0;  22 **for**(**int** i=0;i<7;i++){  23 *//找数组最大值*  24 **for**(**int** j=0;j<7;j++){  25 **if**(max<sumTime[j]){  26 max=sumTime[j];  27 max\_index=j;  28 }  29 }  30 **printRow**(num,max\_index);  31 System.out.**println**("总工时："+sumTime[max\_index]);  32 sumTime[max\_index]=0;  33 max=0;max\_index=0;  34 }  35 }  36  37 *//打印指定行*  38 **public static void** **printRow**(**int**[][]num,**int** row){  39 **for**(**int** i=0;i<7;i++){  40 System.out.**print**(num[row][i]+"\t");  41 }  42 }  43  44 *//返回一个总工时数组*  45 **public static int**[] **sortRow**(**int**[][]num){  46 **int** [] sumTime = **new int**[7];  47 **for**(**int** i=0;i<7;i++){  48 sumTime[i]=**sumRow**(num,i);  49 }  50 **return** sumTime;  51 }  52  53 *//计算每行总工时数*  54 **public static int** **sumRow**(**int** [][]num,**int** row){  55 **int** sum=0;  56 **for**(**int** i=0;i<7;i++){  57 sum+=num[row][i];  58 }  59 **return** sum;  60 }  61 }  **课后题8.6**  01 **package** sdnu.wry.demo;  02 **import** java.util.\*;  03  04 **public class** Homework{  05 **public static void** **main**(String[] args) {  06 **int**[][] a = **new int**[3][3];  07 **int**[][] b = **new int**[3][3];  08  09 System.out.**println**("随机生成数组");  10 *//随机生成矩阵a,b*  11 **for**(**int** i=0;i<3;i++){  12 **for**(**int** j=0;j<3;j++){  13 a[i][j]=(**int**)(Math.**random**()\*9+1);  14 b[i][j]=(**int**)(Math.**random**()\*9+1);  15 }  16 }  17  18 System.out.**println**("矩阵a为：");  19 **for**(**int** i=0;i<3;i++){  20 **for**(**int** j=0;j<3;j++){  21 System.out.**print**(a[i][j]+"\t");  22 }  23 System.out.**println**();  24 }  25 System.out.**println**("矩阵b为：");  26 **for**(**int** i=0;i<3;i++){  27 **for**(**int** j=0;j<3;j++){  28 System.out.**print**(b[i][j]+"\t");  29 }  30 System.out.**println**();  31 }  32  33 *//打印结果数组*  34 System.out.**println**("相乘后结果为：");  35 **int**[][]c=**multiplyMatrix**(a,b);  36 **for**(**int** i=0;i<3;i++){  37 **for**(**int** j=0;j<3;j++){  38 System.out.**print**(c[i][j]+"\t");  39 }  40 System.out.**println**();  41 }  42  43 }  44 *//矩阵相乘*  45 **public static int**[][] **multiplyMatrix** (**int**[][] a,**int**[][] b){  46 **int**[][] c=**new int**[3][3];  47 **for**(**int** i=0;i<3;i++){  48 **for**(**int** j=0;j<3;j++){  49 c[i][j]=a[i][0]\*b[0][j]+a[i][1]\*b[1][j]+a[i][2]\*b[2][j];  50 }  51 }  52 **return** c;  53 }  54 }  **课后题8.8**  01 **package** sdnu.wry.demo;  02 **import** java.util.\*;  03  04 **public class** Homework{  05 **public static void** **main**(String[] args) {  06 **int** [] x = {0,1,-1,2,-2,-3,-4,5};  07 **int** [] y = {0,1,-1,2,-2,-3,-4,5};  08  09 **double** [][] distance = **new double**[8][8];  10 **for**(**int** i=0;i<8;i++){  11 **for**(**int** j=i+1;j<8;j++){  12 distance[i][j] = Math.**sqrt**(Math.**pow**((x[i]-x[j]),2)+Math.**pow**((y[i]-y[j]),2));  13 System.out.**printf**("%.2f\t",distance[i][j]);  14 }  15 System.out.**println**();  16 }  17  18 **double** min = distance[0][1];  19 List min\_index = **new** **ArrayList**();  20 List min\_x = **new** **ArrayList**();  21 List min\_y = **new** **ArrayList**();  22 *// int min\_index\_x = 0,min\_index\_y = 0;*  23 **for**(**int** i=0;i<8;i++){  24 **for**(**int** j=0;j<8;j++){  25 **if**(distance[i][j]<=min&&distance[i][j]!=0){  26 min=distance[i][j];  27 *// min\_index\_x=i;*  28 *// min\_index\_y=j;*  29 }  30 }  31 }  32  33 **for**(**int** i =0;i<8;i++){  34 **for**(**int** j=0;j<8;j++){  35 **if**(distance[i][j]==min){  36 min\_x.**add**(i);  37 min\_y.**add**(j);  38 }  39 }  40 }  41 **for**(**int** i=0;i<min\_x.**size**();i++){  42 System.out.**println**("点("+x[(**int**)min\_x.**get**(i)]+","+y[(**int**)min\_x.**get**(i)]+")与点("+x[(**int**)min\_y.**get**(i)]+","+y[(**int**)min\_y.**get**(i)]+")");  43 };  44 System.out.**println**("最短距离为"+min);  45 }  46  47 }  **课后题8.11**  01 **package** sdnu.wry.demo;  02 **import** java.math.BigInteger;  03 **import** java.util.\*;  04  05 **public class** Homework{  06 **public static void** **main**(String[] args) {  07 System.out.**println**("随机生成如下数字：");  08 **long** coin = (**int**)(Math.**random**()\*512);  09 System.out.**println**(coin);  10 BigInteger coin\_bi = **new** **BigInteger**(String.**valueOf**(coin));  11 String coin\_bin\_str = **bin**(coin\_bi);  12 String add = "";  13 **if**(coin\_bin\_str.**length**()<9){  14 **for**(**int** i =0;i<9-coin\_bin\_str.**length**();i++){  15 add+="0";  16 }  17 coin\_bin\_str = add+coin\_bin\_str;  18 }  19 System.out.**println**(coin\_bin\_str);  20  21 **for**(**int** i=0;i<9;i++){  22 **if**(i%3==0){  23 System.out.**println**();  24 }  25 System.out.**print**((coin\_bin\_str.**charAt**(i)=='1'?"T":"H")+"\t");  26 }  27 }  28 **public static** String **bin**(BigInteger coin){  29 **return** coin.**toString**(2);  30 }  31 }  **课后题8.15**  01 **package** sdnu.wry.demo;  02 **import** java.math.BigInteger;  03 **import** java.util.\*;  04  05 **public class** Homework{  06 **public static void** **main**(String[] args) {  07 Point [] points1 = {  08 **new** **Point**(1,1),  09 **new** **Point**(2,2),  10 **new** **Point**(3,3),  11 **new** **Point**(4,4),  12 **new** **Point**(5,5)  13 };  14 Point [] points2 = {  15 **new** **Point**(3.4,2),  16 **new** **Point**(6.5,9.5),  17 **new** **Point**(2.3,2.3),  18 **new** **Point**(5.5,5),  19 **new** **Point**(-5,4)  20 };  21 **for**(**int** i=0;i<points1.length;i++){  22 System.out.**print**("("+points1[i].**getX**()+","+points1[i].**getY**()+")\t");  23 }  24 System.out.**println**(**sameLine**(points1));  25 **for**(**int** i=0;i<points2.length;i++){  26 System.out.**print**("("+points2[i].**getX**()+","+points2[i].**getY**()+")\t");  27 }  28 System.out.**println**(**sameLine**(points2));  29 }  30 **public static boolean** **sameLine**(Point[] points){  31 *// double k = (points[0].getY()-points[1].getY())/(points[0].getX()-points[1].getX());*  32 *// double [] k = new double[points.length-1];*  33 List<Double> list = **new** ArrayList<>();  34 **double** k=0,count=0;  35 Map map = **new** **HashMap**();  36 **for**(**int** i=0;i<points.length-1;i++){  37 k = (points[i].**getY**()-points[i+1].**getY**())/(points[i].**getX**()-points[i+1].**getX**());  38 **if**(map.**get**(k)==**null**){  39 map.**put**(k,1);  40 }  41 **else**{  42 map.**put**(k,(**int**)map.**get**(k)+1);  43 }  44 }  45 **return** map.**size**()==1;  46 }  47 }  **课后题8.35**  01 **package** sdnu.wry.demo;  02 **import** java.math.BigInteger;  03 **import** java.util.\*;  04  05 **public class** Homework{  06 **public static void** **main**(String[] args) {  07 **int** [][]matrix = **new int**[5][5];  08 *//随机生成5\*5的0-1矩阵并打印*  09 **for**(**int** i=0;i<5;i++){  10 **for**(**int** j=0;j<5;j++){  11 matrix[i][j] = (**int**)(Math.**random**()+0.5);  12 System.out.**print**(matrix[i][j]+"\t");  13 }  14 System.out.**println**();  15 }  16 **maxSubMatrix**(matrix);  17 }  18 **public static void** **maxSubMatrix**(**int**[][] matrix){  19 **int** index\_i=0,index\_j=0;  20 **int** maxSize = 1;  21 **int** max\_dim=1;  22 **int** dim\_i=0,dim\_j=0;  23 **int** []count = **new int**[5];  24 **boolean**[][]flag = **new boolean**[5][5];  25 **for**(**int** row=0;row<5;row++){  26 **for**(**int** col=0;col<5;col++){  27 **for**(**int** dim=5-Math.**max**(row,col);dim>0;dim--){  28 **if**(**sumIndex**(matrix,row,col,dim)==0||**sumIndex**(matrix,row,col,dim)==(dim\*dim)){*//如果全为0或全为1*  29 max\_dim = dim;  30 dim\_i = row;dim\_j=col;  31 **break**;*//找到当前节点最大矩阵，跳出缩小循环*  32 }  33 }  34 **if**(max\_dim>maxSize){*//如果比记录值大，则更新记录*  35 maxSize=max\_dim;  36 index\_i = dim\_i;  37 index\_j = dim\_j;  38 }  39  40 }  41  42 }  43 System.out.**println**("最大值矩阵起始地址：("+index\_i+","+index\_j+"),大小为："+maxSize);  44 }  45  46 **public static int** **sumIndex**(**int**[][]matrix,**int** i,**int** j,**int** dim){  47 **int** sum=0;  48 **for**(**int** p = 0;p<dim;p++){  49 **for**(**int** q=0;q<dim;q++){  50 sum+=matrix[i+p][j+q];  51 }  52 }  53 **return** sum;  54 }  55 }  **课后题8.36** |
| 调 试 过 程 及 实 验 结  果 | **课后题8.1**    **课后题8.2**    **课后题8.4**    **课后题8.6**    **课后题8.8**    **课后题8.11**    **课后题8.15**    **课后题8.35**    **课后题8.36** |
| 总 结 | 无 |
| 附 录 | Github源码地址：<https://github.com/RuYunW/JavaHomework/tree/master/实验10> |