《Java语言程序设计》

课程实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 专业名称 | 计算机科学与技术 | 年级 | 17级 | 班级 | 计1 |
| 学生姓名 | 任志慷 | **指导老师** | 李焱 | 时间 | 5-10 |

|  |  |
| --- | --- |
| 实验名称 | 数组2 |
| 实  验  目  的  及  要  求 | 目的：  了解熟悉顺序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 | WIN 10 64位  JDK 1.9  Eclipse 2018 |
| 实  验  内  容 | 第8章编程练习题: 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 | **8.1**  **import** java.util.\*;  **public** **class** test {  **public** **static** **double**[] SumColumn(**double**[][]m,**int** columnIndex) {  **for**(**int** j=0;j<columnIndex;j++) {  **double** sum=0;  **for**(**int** i=0;i<m.length;i++) {  sum+=m[i][j];  }  System.***out***.println("Sum of the elements at coiumn"+" "+j+" "+"is"+" "+sum);  }  **return** m[0];  }  **public** **static** **void** main(String[]arges){  Scanner in=**new** Scanner(System.***in***);  **int** a=in.nextInt();  **int** b=in.nextInt();  **double**[][]m=**new** **double**[a][b];  **for**(**int** x=0;x<m.length;x++) {  **for**(**int** y=0;y<m[x].length;y++) {  m[x][y]=in.nextDouble();  }  }  *SumColumn*(m,b);  }  }  **8.2**  **import** java.util.\*;  **public** **class** test {  **public** **static** **double** sumMajorDiagonal(**double**[][]m) {  **double** sum=0;  **for**(**int** j=0;j<m.length;j++) {  sum+=m[j][j];  }  System.***out***.println("Sum of the elements in the major diagonal is"+" "+sum);  **return** 0;  }  **public** **static** **void** main(String[]arges){  Scanner in=**new** Scanner(System.***in***);  **int** a=in.nextInt();  **int** b=in.nextInt();  **double**[][]m=**new** **double**[a][b];  **for**(**int** x=0;x<m.length;x++) {  **for**(**int** y=0;y<m[x].length;y++) {  m[x][y]=in.nextDouble();  }  }  *sumMajorDiagonal*(m);  }  }  **8.4**  **import** java.util.\*;  **public** **class** test {  **public** **static** **double**[] SumColumn(**double**[][]m,**int** columnIndex) {  **double**[] a=**new** **double** [m.length];  **for**(**int** j=0;j<m.length;j++) {  **double** sum=0;  **for**(**int** i=0;i<columnIndex;i++) {  sum+=m[j][i];  a[j]=sum;  }  }  **return** a;  }  **public** **static** String[][] ArraySort(String[]a,**double**[]b) {  String[]m=**new** String[b.length];  String[][]n=**new** String[b.length][2];  **for**(**int** i=0;i<b.length;i++) {  m[i]=a[i]+":"+""+b[i];  n[i]=m[i].split("\\:");  }  **for**(**int** k=0;k<b.length-1;k++) {  **for**(**int** i=0;i<b.length-1-k;i++) {  **if**(n[i][1].compareTo(n[i+1][1])<0) {  **for**(**int** j=0;j<2;j++) {  String text=n[i][j];  n[i][j]=n[i+1][j];  n[i+1][j]=text;}  }  }  }  **return** n;  }  **public** **static** **void** main(String[]arges){  Scanner in=**new** Scanner(System.***in***);  System.***out***.println("请输入数组大小：");  **int** a=in.nextInt();  **int** b=in.nextInt();  **double**[][]m=**new** **double**[a][b];  System.***out***.println("请输入员工姓名：");  String[]n=**new** String[a];  **for**(**int** i=0;i<n.length;++i) {  n[i]=in.next();  }  System.***out***.println("请输入工作时长：");  **for**(**int** x=0;x<m.length;x++) {  **for**(**int** y=0;y<m[x].length;y++) {  m[x][y]=in.nextDouble();  }  }  **double**[]x=*SumColumn*(m,b);  String[][]y=*ArraySort*(n,x);  **for**(**int** i=0;i<a;i++) {  System.***out***.println(Arrays.*toString*(y[i]));}  }  }  **8.6**  **import** java.util.\*;  **public** **class** test {  **public** **static** **double**[][] multiplyMatrix(**double**[][]a,**double**[][]b) {  **double**[] x=**new** **double** [a.length];  **double**[] y=**new** **double** [b.length];  **for**(**int** j=0;j<a.length;j++) {  **double** sum=0;  **for**(**int** i=0;i<a[j].length;i++) {  sum+=a[j][i];  x[j]=sum;  }  }  **for**(**int** i=0;i<b.length;i++) {  **double** sum=0;  **for**(**int** j=0;i<b.length;i++) {  sum+=a[j][i];  y[i]=sum;  }  }  **double**[][]z=**new** **double**[a.length][b.length];  **for**(**int** i=0;i<a.length;i++) {  **for**(**int** j=0;j<b.length;j++){  z[i][j]=x[i]\*y[j];  }  }  **return** z;  }    **public** **static** **void** main(String[]arges){  Scanner in=**new** Scanner(System.***in***);  **int** a=in.nextInt();  **int** b=in.nextInt();  **double**[][]m=**new** **double**[a][b];  **double**[][]n=**new** **double**[a][b];  **for**(**int** x=0;x<m.length;x++) {  **for**(**int** y=0;y<m[x].length;y++) {  m[x][y]=in.nextDouble();  }  }  **for**(**int** x=0;x<n.length;x++) {  **for**(**int** y=0;y<n[x].length;y++) {  n[x][y]=in.nextDouble();  }  }  **double**[][]c=*multiplyMatrix*(m,n);  **for**(**int** i=0;i<m.length;i++) {  System.***out***.println(Arrays.*toString*(c[i]));  }  }  }  **8.8**  **import** java.util.Scanner;  **public** **class** test {  **public** **static** **void** main(String[] args) {  Scanner input = **new** Scanner(System.***in***);  System.***out***.print("Enter the number of points:");  **int** number = input.nextInt();  System.***out***.print("Enter " + number + " points:");  **int**[][] array = **new** **int**[number][2];  **for** (**int** i = 0; i < array.length; i++)  **for** (**int** j = 0; j < array[i].length; j++)  array[i][j] = input.nextInt();  **int**[][] arrays = *point*(array);  **for** (**int** i = 0; i < arrays.length; i++) {  **if** (arrays[i][0] != 0 || arrays[i][1] != 0) {  System.***out***  .println("The closest two points are (" + array[arrays[i][0]][0] + "," + array[arrays[i][0]][1]  + ") and (" + array[arrays[i][1]][0] + "," + array[arrays[i][1]][1] + ")");  }  }  System.***out***.println("The distance is " + *smalldistance*(array));  **for** (**int** i = 0; i < arrays.length; i++)  System.***out***.println(arrays[i][0] + "," + arrays[i][1]);  }  **public** **static** **double** distance(**int** x1, **int** y1, **int** x2, **int** y2) { // 求两点之间的距离  **return** Math.*sqrt*((x2 - x1) \* (x2 - x1) + (y2 - y1) \* (y2 - y1));  }  **public** **static** **double** smalldistance(**int**[][] m) { // 求最小距离  **double** smalldistance = *distance*(m[0][0], m[0][1], m[1][0], m[1][1]);  **for** (**int** i = 0; i < m.length; i++) {  **for** (**int** j = i + 1; j < m[i].length; j++) {  **if** (*distance*(m[i][0], m[i][1], m[j][0], m[j][1]) < smalldistance) {  smalldistance = *distance*(m[i][0], m[i][1], m[j][0], m[j][1]);  }  }  }  **return** smalldistance;  }  **public** **static** **int**[][] point(**int**[][] m) { // 求最小距离点对  **double** smalldistance = *smalldistance*(m);  **int** total = 0;  **int**[][] array = **new** **int**[m.length \* (m.length - 1) / 2][2];  **for** (**int** i = 0; i < m.length; i++) {  **for** (**int** j = i + 1; j < m.length; j++) {  **if** (*distance*(m[i][0], m[i][1], m[j][0], m[j][1]) == smalldistance) {  array[total][0] = i;  array[total][1] = j;  total++;  }  }  }  **return** array;  }  }  **8.11**  **package** demo;  **import** java.util.\*;  **public** **class** test {  **public** **static** **void** main(String[] args) {  Scanner input = **new** Scanner(System.***in***);  System.***out***.print("Enter a number between 0 and 551:");  **int** number = input.nextInt();  **int**[][] array = *matrix*(number);  **for** (**int** i = 0; i < array.length; i++) {  **for** (**int** j = 0; j < array[i].length; j++) {  **if** (array[i][j] != 1)  System.***out***.print("H ");  **else**  System.***out***.print("T ");  **if** (j == 2)  System.***out***.println();  }  }  }  **public** **static** **int**[][] matrix(**int** n) {  **int**[][] array = **new** **int**[3][3];  **for** (**int** i = 2; i >= 0; i--) {  **for** (**int** j = 2; j >= 0; j--) {  **if** (n != 0) {  array[i][j] = n % 2;  n /= 2;  }  **if** (n == 0)  **break**;  }  }  **return** array;  }  }  **8.15**  **package** demo;  **import** java.util.\*;  **public** **class** test {  **public** **static** **void** main(String[] args) {  Scanner input = **new** Scanner(System.***in***);  System.***out***.print("Enter five points:");  **double**[][] array = **new** **double**[5][2];  **for** (**int** i = 0; i < array.length; i++)  **for** (**int** j = 0; j < array[i].length; j++)  array[i][j] = input.nextDouble();  input.close();  **if** (*sameLine*(array))  System.***out***.println("The five points are on the same line");  **else**  System.***out***.println("The five points are not on the same line");  }  **public** **static** **boolean** sameLine(**double**[][] points) {  **boolean** bool = **true**;  **double** slope = (points[1][1] - points[0][1]) / (points[1][0] - points[0][0]);  **for** (**int** i = 3; i < points.length; i++) {  **if** ((points[i][1] - points[0][1]) / (points[i][0] - points[0][0]) != slope) {  bool = **false**;  **break**;  }  }  System.***out***.println(bool);  **return** bool;  }  }  **8.35**  **import** java.util.\*;  **public** **class** test {  **public** **static** **void** main(String[] args) {  Scanner input = **new** Scanner(System.***in***);  System.***out***.print("Enter the number of rows in the matrix:");  **int** row = input.nextInt();  System.***out***.println("Enter the matrix row by row:");  **int**[][] array = **new** **int**[row][row];  **for** (**int** i = 0; i < array.length; i++)  **for** (**int** j = 0; j < array[i].length; j++)  array[i][j] = input.nextInt();  input.close();  **int**[] arrays = *findLargestBlock*(array);  System.***out***.println("The maximum square submatrix is at ("+arrays[0]+","  +arrays[1]+") with size "+arrays[2]);  }  **public** **static** **int**[] findLargestBlock(**int**[][] m) {  **int**[] a = **new** **int**[3];  **boolean** bool = **true**;  **int** l, i, j;  **for** (l = m.length; l > 0; l--) {  **for** (i = 0; i <= m.length - l; i++) {  **for** (j = 0; j <= m.length - l; j++) {  bool = **true**;  **for** (**int** x = i; x < i + l; x++) {  **for** (**int** y = j; y < j + l; y++) {  **if** (m[x][y] != 1) {  bool = **false**;  **break**;  }  }  **if** (bool == **false**)  **break**;  }  **if** (bool == **true**)  **break**;  }  a[0] = i;  a[1] = j;  a[2] = l;  **if** (bool == **true**)  **break**;  }  **if** (bool == **true**)  **break**;  }  **return** a;  }  }  **8.36**  **import** java.util.\*;  **public** **class** test {  **public** **static** **void** main(String[] args) {  Scanner input = **new** Scanner(System.***in***);  System.***out***.print("Enter number n:");  **int** n = input.nextInt();  System.***out***.println("Enter " + n + " rows of letters separated by spaces:");  **char**[][] array = **new** **char**[n][n];  **boolean** bool = **true**;  String str = input.nextLine();  **for** (**int** i = 0; i < array.length; i++) {  str = input.nextLine();  **for** (**int** j = 0; j < array[i].length; j++) {  array[i][j] = str.charAt(2 \* j);  **if** (*isLetter*(array, i)) {  bool = **false**;  **break**;  }  }  **if** (bool == **false**)  **break**;  }  **if** (bool == **false**)  System.***out***.println("Wrong input: the letters must be from A to " + (**char**) (array.length - 1 + 'A'));  **if** (bool == **true**) {  **if** (*isLatinSquare*(array))  System.***out***.println("The input array is a Latin square");  **else**  System.***out***.println("The input array is not a Latin square");  }  }  **public** **static** **boolean** isLetter(**char**[][] m, **int** ii) {  **boolean** bool = **false**;  **for** (**int** i = 0; i < m.length; i++) {  **if** (m[ii][i] > (**char**) m.length + 'A') {  bool = **true**;  **break**;  }  }  **return** bool;  }  **public** **static** **boolean** isLatinSquare(**char**[][] m) {  **boolean** bool = **true**;  **for** (**int** i = 0; i < m.length; i++) {  **for** (**int** j = 0; j < m[i].length; j++) {  **for** (**int** l = j + 1; l < m[i].length; l++) {  **if** (m[i][j] == m[i][l]) {  bool = **false**;  **break**;  }  }  **if** (bool == **false**)  **break**;  }  **if** (bool == **false**)  **break**;  }  **for** (**int** i = 0; i < m.length; i++) {  **for** (**int** j = 0; j < m.length; j++) {  **for** (**int** l = j + 1; l < m.length; l++) {  **if** (m[j][i] == m[l][i]) {  bool = **false**;  **break**;  }  }  **if** (bool == **false**)  **break**;  }  **if** (bool == **false**)  **break**;  }  **return** bool;  }  } |
| 调  试  过  程  及  实  验  结  果 | **8.1**  **8.2**  **8.4**    **8.6**    **8.8**    **8.11**    **8.15**  **8.35**  **8.36** |
| 总  结 | 嫌疑人X的献身 |
| 附  录 | 《JAVA语言程序设计》（基础篇）  P258-268 |