

# Java第七天练习题

## 翻转数组

设存在以下数组:

```
1 char[][] chunxiao = {  
2     {'春', '眠', '不', '觉', '晓'},  
3     {'处', '处', '闻', '啼', '鸟'},  
4     {'夜', '来', '风', '雨', '声'},  
5     {'花', '落', '知', '多', '少'}  
6     };
```

- 1 将其输出成如下形式
- 2 尝试设计一组循环语句，将以上形式的输出结果翻转为以下形式(旋转 90 度):

```
1 public static void main(String[] args) {  
2     // 翻转数组  
3     char[][] chunxiao = {  
4         {'春', '眠', '不', '觉', '晓'},  
5         {'处', '处', '闻', '啼', '鸟'},  
6         {'夜', '来', '风', '雨', '声'},  
7         {'花', '落', '知', '多', '少'}  
8     };  
9     for (int i = 0; i < chunxiao.length; i++) {  
10        System.out.println(Arrays.toString(chunxiao[i]));  
11    }  
12    for (int i = 0; i < chunxiao.length + 1; i++) {  
13        for (int j = chunxiao.length - 1; j ≥ 0; j--) {  
14            System.out.print(chunxiao[j][i] + "\t");  
15        }  
16        System.out.println();  
17    }  
18 }
```

思考：如果是一首词该如何旋转输出：

十年生死两茫茫，不思量，自难忘。

千里孤坟，无处话凄凉。

i

纵使相逢应不识，尘满面，鬓如霜。

夜来幽梦忽还乡，小轩窗，正梳妆。

相顾无言，惟有泪千行。

料得年年肠断处，明月夜，短松冈。

```

1 char[][] poetry = {
2     {'十', '年', '生', '死', '两', '茫', '茫', ',', ', ',
   '不', '思', '量', ',', ', ', '自', '难', '忘', '。'},
3     {'千', '里', '孤', '坟', ',', ', ', '无', '处', '话',
   '凄', '凉', '。'},
4     {'纵', '使', '相', '逢', '应', '不', '识', ',', ', ',
   '尘', '满', '面', ',', ', ', '鬓', '如', '霜', '。'},
5     {'夜', '来', '幽', '梦', '忽', '还', '乡', ',', ', ',
   '小', '轩', '窗', ',', ', ', '正', '梳', '妆', '。'},
6     {'相', '顾', '无', '言', ',', ', ', '惟', '有', '泪',
   '千', '行', '。'},
7     {'料', '得', '年', '年', '肠', '断', '处', ',', ', ',
   '明', '月', '夜', ',', ', ', '短', '松', '冈', '。'}
8 };
9 // 取最长的数组长度
10 int maxLength = poetry[0].length;
11 // 假设第一个最长，从第二个开始查找最长的
12 for (int i = 1; i < poetry.length; i++) {
13     if (poetry[i].length > maxLength){
14         maxLength = poetry[i].length;
15     }
16 }
17 // 对数组进行反转90度
18 for (int i = 0; i < maxLength; i++) {
19     for (int j = poetry.length - 1; j ≥ 0 ; j--) {
20         // 定义一维数组来存储二维数组里拿出来的值
21         char[] arr = poetry[j];
22         if (i > arr.length - 1){
23             // 判断下标是否越界
24             System.out.print("\t");
25         }else {
26             System.out.print(arr[i] + "\t");
27         }
28     }
29     System.out.println();

```

## 数组排序

设存在以下数组:

```

1 // 设 names 数组中存放学生姓名
2 String[] names = { "安琪拉" , "王昭君" , "蔡文姬" , "妲己" , "张良" };
3 // 设 courses 数组中依次存放三门课程的名称
4 String[] courses = { "C++" , "Java" , "Python" };
5 // 设 scores 数组中依次存储的是 names 数组中各个学生的 C++ 、 Java 、
  Python 课程的成绩
6 int[][] scores = {
7     { 90 , 89 , 75 } ,
8     { 59 , 40 , 100 } ,
9     { 100 , 99 , 80 } ,
10    { 80 , 61 , 61 } ,
11    { 60 , 100 , 99 } ,
12 };

```

采用以下程序可以输出每个学生的姓名以及该学生各门课程的成绩:

```

1 for( int i = 0 ; i < scores.length ; i++ ){
2     System.out.print( names[ i ] + " ⇒ " ); // 输出学生姓名
3     for( int j = 0 ; j < scores[ i ] .length ; j++ ){
4         System.out.print( courses[ j ] + ":" ); // 输出课程名称
5         System.out.print( scores[ i ][ j ] ); // 输出课程成绩
6         if( j < scores[ i ] .length - 1 ) {
7             System.out.print( " , " );
8         }
9     }
10    System.out.println();
11 }

```

- ① 设计程序按照各个学生的 Java 成绩进行排序 ( 降序 )
- ② 设计程序, 根据学生总成绩进行排序(降序排列), 并输出学生姓名、每门课程 的名称和该学生的成绩、该学生的总成绩

```

1 public static void main(String[] args) {
2     // 数字排序
3     // 设 names 数组中存放学生姓名
4     String[] names = { "安琪拉" , "王昭君" , "蔡文姬" , "妲己" ,
  "张良" };
5     // 设 courses 数组中依次存放三门课程的名称

```

```

6      String[] courses = { "C++" , "Java" , "Python" };
7      // 设 scores 数组中依次存储的是 names 数组中各个学生的 C++ 、
      Java 、 Python 课程的成绩
8      int[][] scores = {
9          { 90 , 89 , 75 } ,
10         { 59 , 40 , 100 } ,
11         { 100 , 99 , 80 } ,
12         { 80 , 61 , 61 } ,
13         { 60 , 100 , 99 } ,
14     };
15     // 设计程序按照各个学生的 Java 成绩进行排序 ( 降序 )
16     for (int i = 0; i < scores.length; i++) {
17         for (int j = i; j < scores.length; j++) {
18             if (scores[i][1] < scores[j][1]){
19                 // 输出所有科目
20                 for (int k = 0; k < 3; k++) {
21                     scores[i][k] = scores[i][k] ^ scores[j]
22 [k];
23                     scores[j][k] = scores[i][k] ^ scores[j]
24 [k];
25                     scores[i][k] = scores[i][k] ^ scores[j]
26 [k];
27                 }
28             }
29         }
30     }
31
32     System.out.println("=====");
33     // 设计程序, 根据学生总成绩进行排序(降序排列), 并输出学生姓名、每门
      课程的名称和该学生的成绩、该学生的总成绩
34     int[] sums = new int[scores.length];
35     for (int i = 0; i < sums.length; i++) {
36         for (int i1 : scores[i]) {
37             sums[i] += i1;
38         }
39     }
40     // 先把sums冒泡排序
41     for (int i = 0; i < sums.length - 1; i++) {
42         for (int j = 0; j < sums.length - 1 - i; j++) {
43             if (sums[j] < sums[j + 1]) {
44                 // 交换总成绩数组
45                 sums[j] = sums[j + 1] ^ sums[j];

```

```

45         sums[j + 1] = sums[j + 1] ^ sums[j];
46         sums[j] = sums[j + 1] ^ sums[j];
47         // 交换单科成绩的数组
48         int[] temp = scores[j];
49         scores[j] = scores[j + 1];
50         scores[j + 1] = temp;
51         // 交换姓名的数组
52         String name = names[j];
53         names[j] = names[j + 1];
54         names[j + 1] = name;
55     }
56 }
57 }
58 System.out.print("姓名\t\t\t课程名称以及各科成绩\t\t总分");
59 System.out.println();
60 for( int i = 0 ; i < scores.length ; i++ ){
61     System.out.print( names[ i ] + "\t" ); // 输出学生姓名
62     for( int j = 0 ; j < scores[ i ].length ; j++ ){
63         System.out.print( courses[ j ] + ":" ); // 输出课程名
64         System.out.print( scores[ i ][ j ] ); // 输出课程成
65         System.out.print( " " );
66     }
67     if( j < scores[ i ].length - 1 ) {
68         System.out.print( " , " );
69     }
70     System.out.print("\t");
71 }
72 System.out.print(sums[i]);
73 System.out.println();
74 }

```

## 杨辉三角

编写一个程序，生成一个给定行数的杨辉三角

```

1 public static void main(String[] args) {
2     // 杨辉三角
3     // 需求：打印杨辉三角形（行数可以键盘录入）
4     // 分析：看这种图像的规律
5     // A:任何一行的第一列和最后一列都是1
6     // B:从第三行开始，每一个数据是它上一行的前一列和它上一行的本列之和。
7     // 步骤：

```

```

8      //A: 首先定义一个二维数组。行数如果是n,我们把列数也先定义为n。这个n
      的数据来自于键盘录入。
9      //B: 给这个二维数组任何一行的第一列和最后一列赋值为1
10     //C: 按照规律给其他元素赋值从第三行开始，每一个数据是它上一行的前一列
      和它上一行的本列之和。
11     //D:遍历这个二维数组。
12     int[][] tree = new int[6][6];
13     for (int i = 0; i < tree.length; i++) {
14         for (int j = 0; j ≤ i; j++) {
15             // 第一个和最后一个都是1
16             if (j == 0 || j == i) {
17                 tree[i][j] = 1;
18             } else {
19                 tree[i][j] = tree[i - 1][j] + tree[i - 1][j -
20 1];
21             }
22         }
23     }
24     for (int[] ints : tree) {
25         for (int ints2 : ints) {
26             if (ints2 ≠ 0){
27                 System.out.print(ints2 + "\t");
28             }
29         }
30         System.out.println();
31     }

```

## 复制数组【拓展】

有如下数组 `nums`，请通过删除掉其中的 `33`，得到一个新的数组 `newNums`

```

1  int[] nums = {2, 11, 33, 44, 55, 66, 277}
2  //newNums = {2, 11, 44,55, 66, 277}

```

使用多种方式实现：

- ① for 循环（自己写方法实现）
- ② System 类中的 `arraycopy` 方法实现

```

1  public static void main(String[] args) {
2      // 复制数组
3      int[] nums = {2, 11, 33, 44, 55, 66, 277};
4      int[] newNums = new int[nums.length - 1];
5      int key = 33;
6      int keyIndex = -1;

```

```

7      // 找到下标
8      for (int i = 0; i < nums.length; i++) {
9          if (nums[i] == key){
10             keyIndex = i;
11         }
12     }
13     // for循环
14     if (keyIndex < 0){
15         System.out.println("数字不存在");
16     }else {
17         for (int i = 0; i < newNums.length; i++) {
18             if (i < keyIndex){
19                 // 复制删除之前的值到新数组里面
20                 newNums[i] = nums[i];
21             }else {
22                 // 复制删除值后面的到新数组
23                 newNums[i] = nums[i + 1];
24             }
25         }
26         System.out.println(Arrays.toString(newNums));
27     }
28
29     System.out.println("=====");
30     ;
31     // System 类中的 arraycopy 方法实现 删除元素
32     if (keyIndex < 0) {
33         System.out.println("数字不存在");
34     } else {
35         System.arraycopy(nums, 0, newNums, 0, keyIndex);
36         System.arraycopy(nums, keyIndex + 1, newNums,
37             keyIndex, newNums.length - keyIndex);
38         System.out.println(Arrays.toString(newNums));
39     }
40 }

```