

## Chp4 数组练习

### Key Point

- 一维数组
- 数组作为函数的参数、返回值
- 数组的基本操作（插入、删除、扩容）
- 数组的排序
- 二维数组的使用

### 练习

1. 下列数组的声明有哪些是对的？错的应该怎么修改？

- A. `int[] a;`
- B. `int a[] = new int[3];`
- C. `int[] a;`  
`a = {1, 2, 3, 4, 5};`
- D. `int[] a = new int[3] {1, 2, 3};`

2. 看下面的代码，写出输出的结果

```
public class Ex2 {
    public static void main(String[] args) {
        int[] a = {1, 2, 3, 4, 5};
        expand(a);
        changeArray(a);
        printArray(a);
    }
    public static void expand(int[] a) {
        int[] newArray = new int[a.length * 2];
        System.arraycopy(a, 0, newArray, 0, a.length);
        a = newArray;
    }
    public static void changeArray(int[] a) {
        a[0] = 10;
    }
    public static void printArray(int[] a) {
        for(int i = 0; i < a.length; i++) {
            System.out.print(a[i] + "\t");
        }
        System.out.println();
    }
}
```

请选择输出结果：

- A. 10 2 3 4 5
- B. 1 2 3 4 5
- C. 10 2 3 4 5 0 0 0 0 0
- D. 1 2 3 4 5 0 0 0 0 0

3. 写一个函数，计算一个整数数组的平均值
  4. 自定义一个整数数组a，读入一个整数n，如果n 在数组中存在，则输出n 的下标；如果不存在，则输出-1。
  5. 给定一个数组，输出数组中的最大值和最小值
  6. 已知一个二维数组A 表示一个矩阵，求AT。
- 其中，AT 表示矩阵的转置。矩阵转置的含义：表示把一个矩阵行列互换。例如，有下面的矩阵M 以及其转置MT：

$$M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

$$M^T = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$$

7. \*给定一个数组，把这个数组中所有元素顺序进行颠倒。
8. \*数组的扩容。  
给定一个数组，要求写一个expand 函数，把原有数组的长度扩容一倍，并保留原有数组原有的内容。

例如，给定一个数组int[] a = {1, 2, 3}，则扩容之后，a 数组为：{1, 2, 3, 0, 0, 0}

9. \*数组的插入和删除

写两个函数，一个函数为delete 函数，声明如下：

```
public static void delete(int pos)
```

该函数表示删除数组pos 位置上的元素。

第二个函数为insert 函数，声明如下：

```
public static void insert(int pos, int value)
```

该函数表示在数组pos 位置上插入value 值。

为了能在多个函数中使用同一个数组，需要把这个数组写在函数的外面，类的里面，并使用static 修饰。

为了方便，还应该定义一个index 变量，用来保存数组的有效元素的个数。例如下面的代码：

```
public class TestInsertDelete{
    static int[] a = {1, 3, 2, 5, 7}; //多个函数可以直接操作该数组
    static int index = 5;
    public static void main(String args[]){
        delete(2); //1 3 5 7
        insert(1, 4); //1 4 3 5 7
        insert(0, 6); //6 1 4 3 5 7
    }
    public static void delete(int pos){
        ...
    }
    public static void insert(int pos, int value){
```

```
...  
}  
}
```

实现方式:

1. delete 方法: 把数组pos 位后的所有元素向前移动1 位
2. insert 方法: 把数组pos 位以及之后的元素向后移动1 位, 然后设置value。要注意的是, insert 时有可能需要数组扩容。有效元素的个数的含义: 对于a 数组, 调用一次delete 之后, a 数组的长度不变, 长度依然为5, 然而有效元素的个数为4 个。

Tips:

insert 方法中, 如何判断是否需要扩容: 比较有效元素的个数和数组的长度, 如果这两者一致, 则需要扩容。

10. \*完成数组的冒泡排序算法: 给定一个数组: `int[] a = {1, 3, 2, 7, 5}`, 利用冒泡排序对其按照从小到大的顺序排序, 然后输出结果。

11. \*使用第二种算法对数组进行排序

12. \*输出杨辉三角

杨辉三角如下:

```
1  
1 1  
1 2 1  
1 3 3 1  
1 4 6 4 1  
.....
```

杨辉三角的特点:

1. 第i 行有i 个元素
2. 每一行的第一个元素和最后一个元素都为1
3. 除了1 之外, 每个元素的值, 都等于上一行同位置的元素以及前一个元素的和。

例如:

1 4 6 4 1 的下一行

1 a1 a2 a3 a4 1

$a1 = 4 + 1 = 5$

$a2 = 6 + 4 = 10$

$a3 = 4 + 6 = 10$

$a4 = 1 + 4 = 5$

依次类推。

要求: 读入一个整数n, 输出杨辉三角的前n 行

13. \*数学黑洞6174

已知: 一个任意的四位正整数。将数字重新组合成一个最大的数和最小的数相减, 重复这个过程, 最多七步, 必得6174。即:  $7641 - 1467 = 6174$ 。将永远出不来。

求证: 所有四位数数字 (全相同的除外), 均能得到6174。输出掉进黑洞的步数。

14. \*筛选法求质数: 输入一个整数n, 求小于这个整数的所有质数。

算法：定义一个长度为n 的boolean 数组，true 表示是质数，false 表示不是质数。初始均为true。之后从2 开始循环：

1. 找到第一个值为true 的下标i
2. 把所有下标为i 的倍数的值置为false。

直到扫描完数组中的所有数值。

最后遍历数组，如果下标i 的值为true，则说明i 为质数。

15. \*\*定义一个奇数阶二维数组, 把每个元素顺序填入不同的自然数, 要求行列和对角线元素相加的结果相等

算法：

1. 第一个数字填在第一行正中间
2. 如果可以填在斜上方，则数字尽量填在斜上方
3. 如果斜上方出了上边界，则数字填入同一列最下端
4. 如果斜上方出了右边界，则数字填入同一行最左端
5. 如果既出了右边界，又出了上边界，则数字填入上一个数字的下方
6. 如果斜上方已经被填过，则数字填入上一个数字的下方。

以3\*3 的为例：

$$\begin{aligned}
 & \text{(规则1)} \begin{bmatrix} 1 & & \\ & & \\ & & \end{bmatrix} \rightarrow \text{(规则3)} \begin{bmatrix} 1 & & \\ & & \\ & 2 & \end{bmatrix} \rightarrow \text{(规则4)} \begin{bmatrix} 1 & & \\ 3 & & \\ & 2 & \end{bmatrix} \\
 & \rightarrow \text{(规则6)} \begin{bmatrix} 1 & & \\ 3 & & \\ 4 & & 2 \end{bmatrix} \rightarrow \text{(规则2)} \begin{bmatrix} 1 & & \\ 3 & 5 & \\ 4 & & 2 \end{bmatrix} \rightarrow \text{(规则2)} \begin{bmatrix} 1 & 6 & \\ 3 & 5 & \\ 4 & & 2 \end{bmatrix} \\
 & \rightarrow \text{(规则5)} \begin{bmatrix} 1 & 6 & \\ 3 & 5 & 7 \\ 4 & & 2 \end{bmatrix} \rightarrow \text{(规则4)} \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & & 2 \end{bmatrix} \rightarrow \text{(规则3)} \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}
 \end{aligned}$$

要求：读入一个奇数n，按照上述规则，输出n\*n 的方阵。

16. \*\*十五个猴子围成一圈选大王，依次1-7 循环报数，报到7 的猴子被淘汰，直到最后一只猴子成为大王。问，哪只猴子最后能成为大王？

17. \*\*螺旋填数

读入两个整数m，n，输出一个m 行n 列的矩阵，这个矩阵是1~m\*n 这些自然数按照右、下、左、上螺旋填入的结果。

例如：读入4， 5，

则输出

```

1 2 3 4 5
14 15 16 17 6
13 20 19 18 7
12 11 10 9 8

```

18. \*\*\*八皇后问题

在8×8 的国际象棋盘上，放置八个皇后，使任何一个皇后都不能吃掉另一个。国际象棋规则中，皇后可以吃到任何一个与他在同一行、同一列或者同一斜

线上的敌方棋子，所以八皇后问题的所有解满足：8 个皇后都不在同一行、同一列，或者同一斜线上。

输出所有的解。

提示：使用递归。