

数组使用

动态初始化

```
1 | int[] a = new int[9];
2 | a = {1,2,3,4,5,6,7,8,9};
```

静态初始化

```
1 | int[] a = {1,2,3,4,5,6,7,8,9};
```

数组的扩容和缩减

```
1 |
2 |
3 | //数组扩容
4 | import java.util.Scanner;
5 |
6 |
7 | public class ArrayAdd{
8 |     //main
9 |     public static void main(String[] args){
10 |         int addSize = 1;
11 |         int[] array1 = {1,2,3}; //定义初始数组
12 |         //定义循环让用户可以继续添加
13 |         while (true){
14 |             int[] newArray = new int[array1.length + addSize]; //定义扩容数组
15 |             int addNum; //需要扩容的值
16 |
17 |             Scanner addScanner = new Scanner(System.in);
18 |             //将初始数组的值复制到扩容数组中
19 |             for (int i = 0; i<array1.length ; i++){
20 |                 newArray[i] = array1[i];
21 |             }
22 |             //键盘输入需要添加的值
23 |             System.out.print("请输入你想添加的值: ");
24 |             addNum = addScanner.nextInt();
25 |             //将输入的值加入到扩容模型的最后一个，应为最后一个数组就是为"0"
26 |             newArray[newArray.length - 1] = addNum;
27 |
28 |             //将初始数组指向扩容数组，初始数组的值就被销毁
29 |             array1 = newArray;
30 |
31 |             for (int i = 0; i<array1.length ; i++){
32 |                 System.out.print(array1[i]+"\\t");
33 |             }
34 |             System.out.println("\\n是否继续添加?(y/n):");
35 |             char judge = addScanner.next().charAt(0);
36 |             if (judge == 'y'){
```

```

37         continue;
38     }else if (judge == 'n') {
39         break;
40     }else{
41         System.out.println("输入错误");
42         break;
43     }
44 }
45 }
46 }

```

```

1  //数组缩减
2  import java.util.Scanner;
3
4  public class ArrayReduce{
5      //main
6      public static void main(String[] args){
7          int reduceSize = 1;
8          int[] array1 = {1,2,3,4,5};
9          while (true) {
10             int[] newArray = new int[array1.length - reduceSize];
11             Scanner reduceScanner = new Scanner(System.in);
12             //将初始数组的值复制到扩容数组中
13             for (int i = 0; i<newArray.length ; i++){
14                 newArray[i] = array1[i];
15             }
16             array1 = newArray;
17             for (int i = 0; i<newArray.length ; i++){
18                 System.out.print(array1[i]+"\\t");
19             }
20             System.out.println("\\n是否继续缩减?(y/n):");
21             char judge = reduceScanner.next().charAt(0); //接受char类型
22             if (judge == 'y'){
23                 if (array1.length == 1 ){
24                     System.out.println("不能在缩减");
25                     break;
26                 }else {
27                     continue;
28                 }
29             }else if (judge == 'n') {
30                 break;
31             }else{
32                 System.out.println("输入错误");
33                 break;
34             }
35         }
36     }
37 }

```

⚠ 数组的使用和注意事项

- 数组是多个相同数据的组合，实现对这些数据的统一管理
- 数组中的元素可以是任何元素类型，包括基本类型和引用类型，但是不能混用
- 数组创建后，如果没有赋值，是有默认值的

```
1  int 0    short 0    byte 0    long 0
2  float 0.0  double 0.0    char \u0000
3  boolean false    String null
```

- 使用数组的步骤为：1.声明数组并且开辟空间 2.给数组各个元素赋值 3.使用数组
- 数组属于引用类型，数组型数据是对象(object)

二维数组

遍历二维数组中的值并求和

```
1  //遍历二维数组并计算他们的和
2
3  public class TwoDimensionalArray01{
4      //main
5      public static void main(String[] args){
6          int arr[][] = {{4,6},{1,4,5,7},{-2}};
7          int sum = 0;
8          for (int i = 0; i<arr.length; i++){
9              for (int j = 0; j<arr[i].length; j++){
10                 sum += arr[i][j];
11             }
12         }
13         System.out.print("二维数组的总数为:"+sum);
14     }
15 }
```

杨辉三角

```
1  //杨辉三角
2
3  public class YangHui {
4      //main
5      public static void main(String[] args){
6          int arr[][] = new int[10][];
7          for (int i = 0 ; i<arr.length; i++){
8              arr[i] = new int[i];
9              for (int j = 0 ; j<arr[i].length; j++){
10                 if (i <= 2){
11                     arr[i][j]=1;
12                 }else{
13                     if (j==0 || j == i-1){
14                         arr[i][j] = 1;
```

```

15         }else{
16             arr[i][j] = arr[i-1][j] + arr[i-1][j-1];
17         }
18     }
19 }
20 }
21 for(int i = 0; i<arr.length; i++){
22     for(int j=0; j<arr[i].length; j++){
23         System.out.print(arr[i][j]+"\\t");
24     }
25     System.out.println();
26 }
27 }
28 }

```

⚠️二维数组的细节和注意事项

- 一维数组的声明方式有 `int[] x` 或者 `int x[]`
- 二维数组的声明方式有 `int[][] y`、`int[] y[]`、`int y[][]`
- 二维数组是多个一维数组构成，，各个一维数组长度可以一样也可以不一样

可变长参数

可变长参数就是可以传入多个对应类型的 `实参`，实现方法其实就是数组

```

1 public class Person {
2     String name;
3     int age;
4     String sex;
5
6     public void test(String... str){ //String... str 表示可以传递多个String类型的参
数
7         for (String s : str) { //加强For循环，使用这个就是可以遍历这个数组
8             System.out.println(s);
9         }
10    }
11 }
12

```

可以直接传入多个参数，也可以不传参数

```

1 public class Main {
2
3     public static void main(String[] args) {
4         Person per = new Person();
5         per.test("1","2","3"); //直接传入
6         Person per2 = new Person();
7         per.test();
8
9     }

```

```
10 }
11
12 //输出:
13 1
14 2
15 3
```

★ 如果存在其他参数，则可变长参数必须是在后面

```
1 public void test(int a,String... str){ //String... str 表示可以传递多个String类
    型的参数
2     for (String s : str) {
3         System.out.println(s);
4     }
5 }
```

⚠ 这里只是拿String类型做个例子，可变长数组不仅限于String类型

```
1 public void test2(int... a){
2     for (int i : a) {
3         System.out.println(i);
4     }
5 }
6
7 public void test2(long... a){
8
9 }
```

之前一直很好奇Main函数中的 `String[] args` 是什么，它其实就是为我们在用命令行来执行 `JAVA` 程序时所接受的参数，其实类似于 `shell` 的 `$1`, `$2`

```
1 public static void main(String[] args) {
2     for (String arg : args) {
3         System.out.println(arg); //本质上它也是一个数组，就可以使用遍历列表的方式来读
    取
4     }
5 }
```

如果是在代码编辑器中执行，输出应该是空，如果我们在命令行中使用后面在随便接入几个参数,我这里不知道为什么一直报错，如果正常应该就是会输出这后面接入的这几个参数

```
1 PS D:\learnCode\Java\src> java com/company/zy01/Main bx bxb bx
2 错误：找不到或无法加载主类 com.company.zy01.Main
3
```

练习

```
1 //已知有个升序的数组，要求插入个元素，该数组顺序依然是升序，比如:[10, 12, 45, 90]，添加23
  后，数组为 [10, 12, 23, 45, 90]
2 import java.util.Scanner;
3
4 public class Homework04 {
5     //main
6     public static void main(String[] args){
7         //自己做的版本
8         // int arr[] = {10,12,45,90};
9         // int newArr[] = new int[arr.length+1];
10        // Scanner addScanner = new Scanner(System.in);
11        // for(int i = 0; i < newArr.length; i++){
12        //     if (i < arr.length){
13        //         newArr[i] = arr[i];
14        //     }else{
15        //         System.out.print("请输入你需要添加的值：");
16        //         newArr[i] = addScanner.nextInt();
17        //     }
18        // }
19
20        // arr = newArr;
21
22        // for(int i = 0; i<arr.length; i++){
23        //     for(int j = i+1; j<arr.length; j++){
24        //         if(arr[i] > arr[j]){
25        //             int temp = arr[i];
26        //             arr[i] = arr[j];
27        //             arr[j] = temp;
28        //         }
29        //     }
30        // }
31
32        // for (int i=0 ; i<arr.length ; i++ ) {
33        //     System.out.print(arr[i]+ " ");
34        // }
35
36        //老师的思路
37        int arr[] = {10,12,45,90};
38        int newArr[] = new int[arr.length+1];
39        Scanner addScanner = new Scanner(System.in);
40        System.out.print("请输入你需要添加的值：");
41        int addNum = addScanner.nextInt();
42        int index = -1;
43        for(int i = 0; i < arr.length; i++){
44            if(arr[i] >= addNum){
45                index = i;
46                break;
47            }
48        }
```

```
49     System.out.println(index);
50     for (int i = 0,j=0 ; i< newArr.length ; i++){
51         if (i != index){
52             newArr[i] = arr[j];
53             j++;
54         }else{
55             newArr[i] = addNum;
56         }
57     }
58
59     arr = newArr;
60
61     for (int i=0 ; i<arr.length ; i++ ) {
62         System.out.print(arr[i]+ " ");
63     }
64
65 }
66 }
```