

JavaSE一维数组

1. 为什么需要数组

- 可以用一个变量，存储相同数据类型的多个值。

2. 有数组和没有数组的区别案例演示：

```
1 package org.gyun.array;
2
3 import java.util.Scanner;
4
5 /**
6  * @ClassName: ArrayTest1.java
7  * @Description: 没有使用数组前与使用数组后的比较及为什么要使用数组
8  * @Author: DongGaoYun
9  * @URL: www.gyun.org
10 * @Email: DongGaoYun@qq.com
11 * @QQ: 1050968899
12 * @Weixin: QingYunJiao
13 * @Date: 2019-9-9 下午2:14:32
14 * @Version: 1.0
15 */
16 public class ArrayTest1 {
17
18     /**
19      * 给一个班的3位同学算平均分。 1.动态的录入分数 Scanner input=new
Scanner(System.in); 三步走：引包
20      * 创建对象 使用数据 循环给它赋值。用for之前，看它有没有固定的次数，有，就用for
for(int i=0;i<3;i++){//打印需求};
21      * sum变量计算总分数，除以3位同学，得出平均分
22      *
23      */
24     public static void main(String[] args) {
25         // 调用方法
26         // 简单方便
27         // forTest();
28         // array1();
29         array2();
30
31     }
32     /**
33      * 求静态数组元素平均分
34      */
35     private static void array2() {
36         // 声明数组 几步走： 声明数组、开辟空间、赋值、使用
```

```

37 // 简化：声明数据并赋值、使用
38 // 静态的声明数组
39 int[] score1 = new int[] { 99, 98, 97 }; // 左右边中括号里不能指定数组的
长度
40
41 int sum = 0;
42 // 创建输入对象Scanner
43 Scanner input = new Scanner(System.in);
44 /*
45 * 普通for循环 for (int i = 0; i < score1.length; i++) {
46 * 等价于sum=sum+score1[i]; sum += score1[i]; }
47 */
48 // 增强for循环
49 for (int score : score1) {
50     // 等价于sum=sum+score;
51     sum += score;
52 }
53 //打印平均分
54 System.out.println(sum / score1.length);
55 }
56 /*
57 * 使用数组后，保存多个数据，仅需一个数组便可以搞定，优势明显
58 */
59 private static void array1() {
60     // 声明数组 几步走： 声明数组、开辟空间、赋值、使用
61     // 简化：声明数据并赋值、使用
62     // 动态的声明数组
63     int[] score = new int[7];
64
65     int sum = 0;
66     // 创建输入对象Scanner
67     Scanner input = new Scanner(System.in);
68     // 循环 //score.length是拿到数组的长度。
69     for (int i = 0; i < score.length; i++) {
70         System.out.println("请输入第" + (i + 1) + "位同学的成绩: ");
71         score[i] = input.nextInt();
72         // 等价于sum=sum+score[i];
73         sum += score[i];
74     }
75     System.out.println(sum / score.length);
76 }
77 /**
78 * 没有使用数组之前，想保存一些数据，需要定义多个相应的变量名，使用复杂
79 */
80 private static void forTest() {
81     // 声明变量
82     int num = 3;
83     String zs = "张三";
84     String ls = "李四";
85     String wmz = "王麻子";
86     int z = 0;
87     int l = 0;
88     int w = 0;
89     int sum = 0;
90     // 创建输入对象Scanner
91     Scanner input = new Scanner(System.in);
92     // 循环
93     for (int i = 0; i < num; i++) {

```

```

94     System.out.println("请输入第" + (i + 1) + "位同学的成绩: ");
95     int score = input.nextInt();
96     if (i == 0) {
97         z = score;
98     }
99     if (i == 1) {
100        l = score;
101    }
102    if (i == 2) {
103        w = score;
104    }
105    // 等价于sum=sum+score;
106    sum += score;
107 }
108 System.out.println(sum / num);
109 System.out.println(zs + "的成绩为:" + z);
110 System.out.println(ls + "的成绩为:" + l);
111 System.out.println(wmz + "的成绩为:" + w);
112 }
113 }
114

```

3. 什么是数组

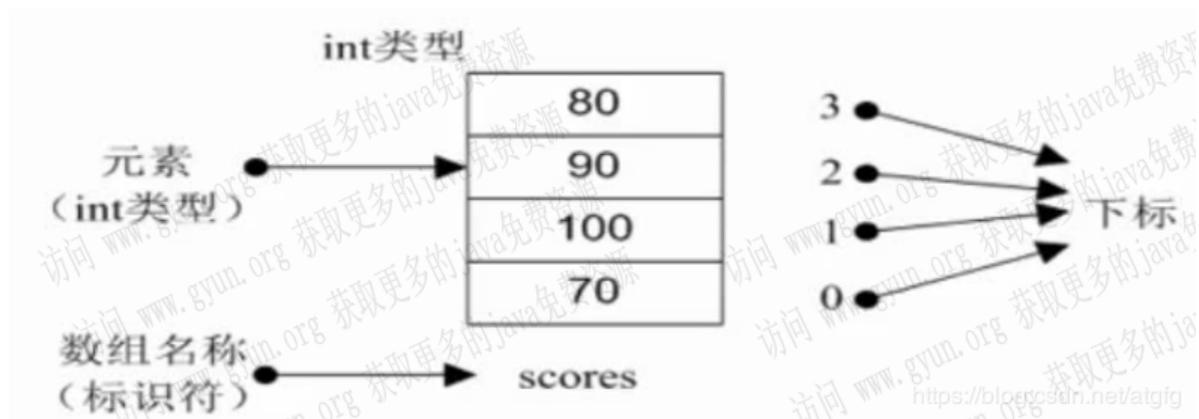
数组是存储相同数据类型多个元素的集合。

4. 数组和变量的小知识：

- 声明一个变量就是在内存空间中划出一块合适的空间
- 而声明一个数组就是在内存空间中划出一串相同数据类型且连续的空间

5. 数组基本要素：

- 标识符（数组名）
- 数组元素
- 元素下标：从0开始
- 元素类型



6. 数组声明方式：

6.1. 动态声明数组

1. 第一种**声明动态数组**分二步走：

- 声明数组：int[] score; //c++的成绩 告诉计算机数据类型是什么

```
1 | 语法1: 数据类型 数据名[];  
2 |  
3 |     int  score[];  
4 |  
5 | 语法2: 数据类型[] 数据名;  
6 |  
7 |     int[] score;
```

注意：声明数组时不规定数组长度

- 分配空间: 告诉计算机分配几个连续的空间

```
1 | 语法: 数组名 = new 数据类型[数组的长度];  
2 |  
3 |     score=new int[8]
```

附：**使用数组**分四步走：声明数组、分配空间、赋值、使用

- 赋值: score[0]=88; 向分配的格子里放数据
- 使用: score[0]++;

2. 第二种**声明动态数组**分一步走：

- 声明数组且分配空间(如下)

```
1 | 语法1: 数据类型[] 数组名 = new 数据类型[数组的长度]; (推荐使用)  
2 |  
3 |     int[] arr = new int[5];  
4 |  
5 | 语法2: 数据类型 数组名[] = new 数据类型[数组的长度];  
6 |  
7 |     int arr[] = new int[5];
```

注意：声明数组时左边无需指定数组长度 但右边需指定数组长度，系统会给出初始化值

附：**使用数组**分三步走：声明数组且分配空间、赋值、使用

- 赋值：arr[0]=1; arr[1]=2; 向分配的格子里放数据
- 使用: score[0]++;

6.2. 静态声明数组

- **边声明边赋值**的有两种声明方式：

```
1
2  语法1: 数据类型[] 数组名 = new 数据类型[] {元素1,元素2,...};
3
4  int[] score = new int[] { 99, 98, 97 };
```

注意：静态声明数组时，静态赋值不能指定数据长度

```
1  语法2: 简写方式: 数据类型[] 数组名 = {元素1,元素2,...};
2
3  int[] score = { 99, 98, 97 };
4
5  注意: 静态声明数组时, 要手工给出初始化值, 系统会自动计算数组长度
```

6.3. 数组存储类型小知识：

- 数组既可以存储基本数据类型，也可以存储引用数据类型。
- 数组元素根据类型不同，有不同的初始值。

int类型初始化为：0

String类型初始化为：null

7. 动态数组的演示案例：

```
1 package org.gyun.array;
2
3 /**
4  * @ClassName: ArrayTest2.java
5  * @Description: 声明数组并化内存图
6  * @Author: DongGaoYun
7  * @URL: www.gyun.org
8  * @Email: DongGaoYun@qq.com
9  * @QQ: 1050968899
10 * @Weixin: QingYunJiao
11 * @Date: 2019-9-9 下午2:56:08
12 * @Version: 1.0
13 */
14 public class ArrayTest2 {
15
16     /**
17     * 完成需求: 画一个数组的内存图
18     */
19     public static void main(String[] args) {
20         // 动态声明数组 使用动态数组可以分四步走: 声明数组、开辟空间、赋值、使用,
21         //使用动态数组也可以分三步走: : 声明数组并开辟空间、赋值及使用
22         //1.声明数组并开辟空间
23         int[] num = new int[5]; //推荐使用
24         //int num[] = new int[5];
25         //2.赋值
26         num[0] = 1;
27         num[4] = 4;
28         //3.使用
29         System.out.println(num[0]); //1
```

```

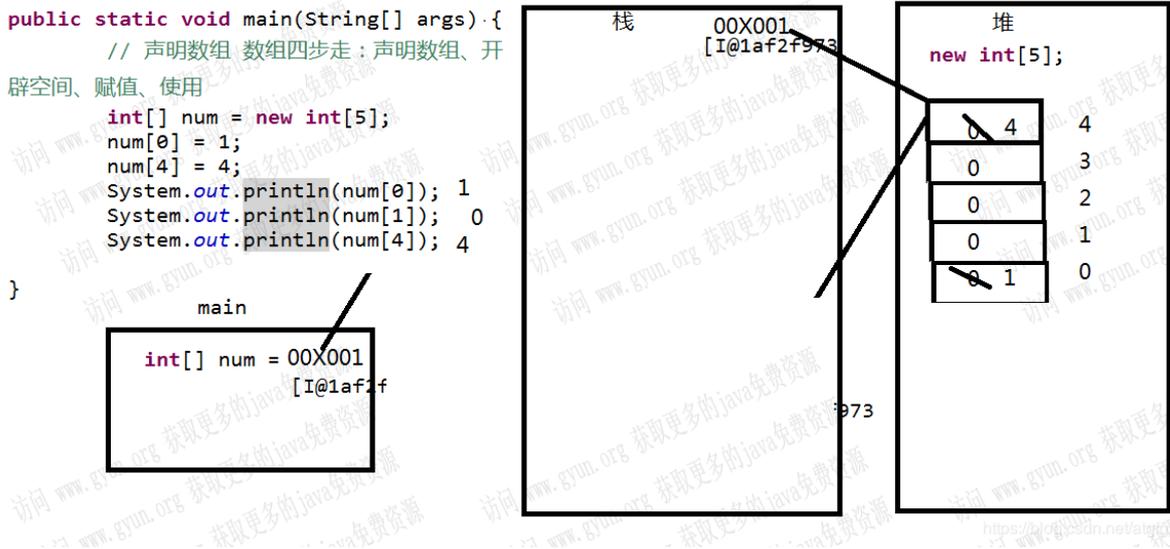
30     System.out.println(num[1]); //0
31     System.out.println(num[4]); //4
32 }
33
34 }
35

```

8. Java中的内存分配以及栈和堆的小知识：

- 栈(掌握)
存放局部变量
- 堆(掌握)
存放new出来的数组或对象
- 方法区
后面面向对象部分讲解

9. 内存图：



10. 数组猜数字演示案例：

```

1 package org.gyun.array;
2
3 import java.util.Scanner;
4
5 /**
6  * @ClassName: ArrayTest3.java
7  * @Description: 猜数字
8  * @Author: DongGaoYun
9  * @URL: www.gyun.org
10 * @Email: DongGaoYun@qq.com
11 * @QQ: 1050968899
12 * @Weixin: QingYunJiao
13 * @Date: 2019-9-9 下午3:18:48
14 * @version: 1.0
15 */
16 public class ArrayTest3 {

```

```

17
18     /**
19     * 需求： 有一个数列： 8， 4， 2， 1， 23， 344， 12 循环输出数列的值 有固定次数，我们就用
for 普通for 增加for
20     *
21     * 求数列中所有数值的和 在用for遍历时累加
22     *
23     * 猜数游戏 从键盘中任意输入一个数据，判断数列中是否包含此数 用Scanner录入数字 数字
用==最好 对象用equals最好
24     */
25     public static void main(String[] args) {
26         // 声明数组 四步走：声明数组、开辟空间、赋值、使用
27         // int[] num=new int[]{8,4,2,1,23,344,12};
28         int[] num = { 8, 4, 2, 1, 23, 344, 12 };
29         int count = 0;
30         // 普通for
31         for (int i = 0; i < num.length; i++) {
32             System.out.print(num[i] + "\t");
33             count += num[i];
34         }
35         // 打印之和
36         System.out.println(count);
37         // 三步走：引包、创建对象、使用数据
38         Scanner input = new Scanner(System.in);
39         System.out.print("请输入一个数字：");
40         int nums = input.nextInt();
41         boolean flag = false;
42         for (int i = 0; i < num.length; i++) {
43             //遍历元素时，只要是输入的值与数组元素的值相同，就把标识变量的值变为true,然
后就跳出。
44                 if (nums == num[i]) {
45                     flag = true;
46                     break;
47                 }
48             }
49             // 判断
50             if (flag) {
51                 System.out.println("恭喜你，猜对了！");
52             } else {
53                 System.out.println("继续加油！");
54             }
55         }
56     }
57

```

11. 求最大值演示案例：

```

1 package org.gyun.array;
2
3 import java.util.Scanner;
4
5 /**
6  * @ClassName: ArrayTest3.java
7  * @Description: 求出最大值
8  * @Author: DongGaoYun
9  * @URL: www.gyun.org

```

```

10  * @Email: DongGaoYun@qq.com
11  * @QQ: 1050968899
12  * @Weixin: QingYunJiao
13  * @Date: 2019-9-9 下午3:18:48
14  * @Version: 1.0
15  */
16  public class ArrayTest4 {
17
18      /**
19       * 需求： 用键盘输入五位学员成绩，并求出最大值
20       *
21       * 求数列中所有数值的和 在用for遍历时累加
22       *
23       * 猜数游戏 从键盘中任意输入一个数据，判断数列中是否包含此数 用Scanner录入数字 数字
        用==最好 对象用equals最好
24       */
25      public static void main(String[] args) {
26          // 三步走： 引包、创建对象、使用数据
27          Scanner input = new Scanner(System.in);
28          // 声明数组 四步走： 声明数组、开辟空间、赋值、使用
29          int[] score = new int[5];
30          for (int i = 0; i < score.length; i++) {
31              System.out.print("请输入第" + (i + 1) + "学员的java成绩: ");
32              score[i] = input.nextInt();
33          }
34          int max = score[0]; // 注意一定是在赋值后再定义max变量，否则max的值是0
35          // 普通for循环
36          for (int i = 1; i < score.length; i++) {
37              if (score[i] > max) {
38                  max = score[i];
39              }
40          }
41          // 打印最大数
42          System.out.println(max);
43      }
44  }

```

12. 求最小值演示案例：

```

1  package org.gyun.array;
2
3  import java.util.Scanner;
4
5  /**
6   * @ClassName: ArrayTest3.java
7   * @Description: 求出最小值
8   * @Author: DongGaoYun
9   * @URL: www.gyun.org
10  * @Email: DongGaoYun@qq.com
11  * @QQ: 1050968899
12  * @Weixin: QingYunJiao
13  * @Date: 2019-9-9 下午3:18:48
14  * @Version: 1.0
15  */

```

```

16 public class ArrayTest6 {
17
18     /**
19      * 需求： 用键盘输入五位学员成绩，并求出最小值
20      *
21      * 求数列中所有数值的和 在用for遍历时累加
22      *
23      * 猜数游戏 从键盘中任意输入一个数据，判断数列中是否包含此数 用Scanner录入数字 数字
        用==最好 对象用equals最好
24      */
25     public static void main(String[] args) {
26         // 三步走： 引包、创建对象、使用数据
27         Scanner input = new Scanner(System.in);
28         // 声明数组 四步走： 声明数组、开辟空间、赋值、使用
29         int[] score = new int[2];
30         for (int i = 0; i < score.length; i++) {
31             System.out.print("请输入第" + (i + 1) + "学员的java成绩: ");
32             score[i] = input.nextInt();
33         }
34         int min = score[0]; // 注意一定是在赋值后再定义min变量，否则min的值是0
35         // 普通for循环
36         /* for (int i = 1; i < score.length; i++) {
37             if (score[i] <= min) {
38                 min = score[i];
39             }
40         } */
41         // 增强for循环
42         for (int s : score) {
43             if (s < min) {
44                 min = s;
45             }
46         }
47         // 打印最小数
48         System.out.println(min);
49     }
50 }
51

```

13. 插入数字的案例演示：

```

1 package org.gyun.array;
2
3 import java.util.Scanner;
4
5 /**
6  * @ClassName: ArrayTest3.java
7  * @Description: 插入一个数字，并降序的排列 数组的灵活使用
8  * @Author: DongGaoYun
9  * @URL: www.gyun.org
10 * @Email: DongGaoYun@qq.com
11 * @QQ: 1050968899
12 * @Weixin: QingYunJiao
13 * @Date: 2019-9-9 下午3:18:48
14 * @version: 1.0
15 */
16 public class ArrayTest5 {

```

```

17
18     /**
19     * 需求： 插入一个数字，并降序的排列 排序前的效果： 99 85 82 63 60 比如： 98 排序
    前的效果： 99 98 85 82 63 60
20     * int[] score = new int[6]; score[0]=99; score[1]=85; score[2]=82;
21     * score[3]=63; score[4]=60;
22     *
23     * Scanner input=new Scanner(System.in); int scores=input.nextInt();
24     *
25     *
26     *
27     * //int index=score.length-1; //比较 //遍历一下 if(scores>score[i]){
    index=i; }
28     *
29     * for( int j=5;j>index;j--)
30     *
31     * 1 score[5]=score[5-1-0]; 4 j=5 4 3 2 score[5-1]=score[5-1-1]; 3 j=4
    3 2 3
32     * score[5-1-1]=score[5-1-2]; 2 j=3 2 1 4 score[5-1-2]=score[5-1-3] j=2
33     */
34     public static void main(String[] args) {
35         // 三步走： 引包、创建对象、使用数据
36         Scanner input = new Scanner(System.in);
37         // 声明数组 四步走： 声明数组、开辟空间、赋值、使用
38         int[] score = new int[6];
39         score[0] = 99;
40         score[1] = 85;
41         score[2] = 82;
42         score[3] = 63;
43         score[4] = 60;
44         System.out.print("请输入需要插入的值： ");
45         int scores = input.nextInt();
46         int index = score.length - 1;
47
48         // int index=socre.length-1;
49         // 比较
50         // 遍历一下
51         for (int i = 0; i < score.length; i++) {
52             if (scores > score[i]) {
53                 index = i;
54                 //break 不加break会一直保存在index变量
55                 break;
56             }
57         }
58         //这个遍历给输入的值腾空间
59         for (int j = score.length - 1; j > index; j--) {
60             score[j] = score[j - 1];
61         }
62         //赋值：
63         score[index]=scores;
64         for (int i = 0; i < score.length; i++) {
65             System.out.println(score[i]);
66         }
67     }
68 }
69

```

14. Java数组的常见错误小知识：

- `ArrayIndexOutOfBoundsException`: 数组索引越界异常
原因：数组索引越界，访问了不存在的下标（索引）。
- `NullPointerException`: 空指针异常
原因：数组不再指向堆内存了，如果还用数组名去访问元素就会报错。