

课程回顾

1. while do-while for循环执行流程

2. return break continue优化

3.扩展练习--菱形的输出方式

```
1  import java.util.Scanner;
2
3  /**
4   * @Author: lc
5   * @Date: 2022/3/12
6   * @Description: 打印菱形
7   * @Version: 1.0
8   */
9  public class Demo {
10     public static void main(String[] args) {
11         Scanner input = new Scanner(System.in);
12         int rows;
13         do {
14             do {
15                 System.out.println("请输入要打印的菱形行数：（总行数只能是奇数）");
16                 rows = input.nextInt();
17                 if(rows%2!=0){
18                     break;
19                 }
20             } while (true);
21
22             for(int i=-rows/2;i<=rows/2;i++){ //控制图形的总行数
23                 for(int k=0;k<Math.abs(i);k++){//控制空格的输出
24                     System.out.print(" ");
25                 }
26
27                 for(int j=0;j<rows-2*Math.abs(i);j++){//控制*的输出
28                     System.out.print("*");
29                 }
30                 System.out.println();//换行
31             }
32             } while (true);
33     }
34 }
```

课程目标

1 数组定义

2 数组元素存取

3 数组循环（遍历数组）

4 数组应用

4-1 求和、求平均值、求最值

4-2 数组查找、添加、修改、删除

课程实施

1 数组

数组概念

数组其实还是java的数据类型。和int、double的作用一样。用来在内存开辟空间，保存数据。

数组作用

数组一次性保存多个具有相同数据类型的数据。简化重复的同类型变量的定义和引用。

案例：保存一件商品名称、单价、库存、生成日期....

实际开发中，保存30件商品的信息。

如何实现？

使用传统的变量定义保存30件商品的信息，变量的个数是非常惊人的！！

一个变量可以一次性保存N个信息？数组 集合

需求：使用变量保存30件商品的名称

方案一：定义30个String的变量，保存

方案二：定义一个数组类型的变量。

数组定义

```
1 //动态赋值：jvm执行时候，由jvm存入默认值
2 数据类型[] 变量名=new 数据类型[实际存入数据个数];
3 解析：
4     数据类型数组保存数据的公共类型类型
5
6 //变形方式一：静态赋值
7 数据类型[] 变量名={数据1,数据2,.....,数据n};
8 细节：定义和赋值必须一行代码完成，不能先定义数组，再赋值。
9
10
11 //变形方式二：静态赋值
12 数据类型[] 变量名=new 数据类型[]{数据1,数据2,.....,数据n};
13 优点：可以先定义。再赋值，分两行写
14 注意实现：new 数据类型[长度]，此处长度一定不能写！！
```

课堂案例：演示三种数据定义并赋值的方式

```
1  /**
2   * @Author: lc
3   * @Date: 2022/3/12
4   * @Description: 数组定义
5   * @Version: 1.0
6   */
7  public class Demo3 {
8      public static void main(String[] args) {
9          //语法一：定义数组并动态赋值
10         int[] arr1_1=new int[3]; //定义数组，并赋值3个0
11         //先定义数组
12         int[] arr1_2; //定义数组
13         //再赋值
14         arr1_2=new int[3]; //数组存入3个0
15
16
17         //语法二：定义数组并静态赋值
18         int[] arr2_1={23,56,90}; //定义数组，并存入23 56 90，没有默认值0
19         //存在问题：语法不能拆分
20         int[] arr2_2;
21         arr2_2={23,56,90};
22
23         //语法三：定义数组并静态赋值
24         int[] arr3_1=new int[]{3,4,5,6,9};
25
26         int[] arr3_2;
27         arr3_2=new int[]{3,4,5,6,9};
28     }
29 }
30
```

细节：数组一旦定义，并指明长度后数组每一个下标位置都有默认值，默认值的规律如下所示

```
1  int byte short long    0
2  float double           0.0
3  char                   '\u0000' 其实就是一个空白
4  String                 null =====
5  boolean                false
```

数组数据的存和取

```
1  //存
2  数组名[下标]=值;
3
4  //取
5  数组名[下标]
```

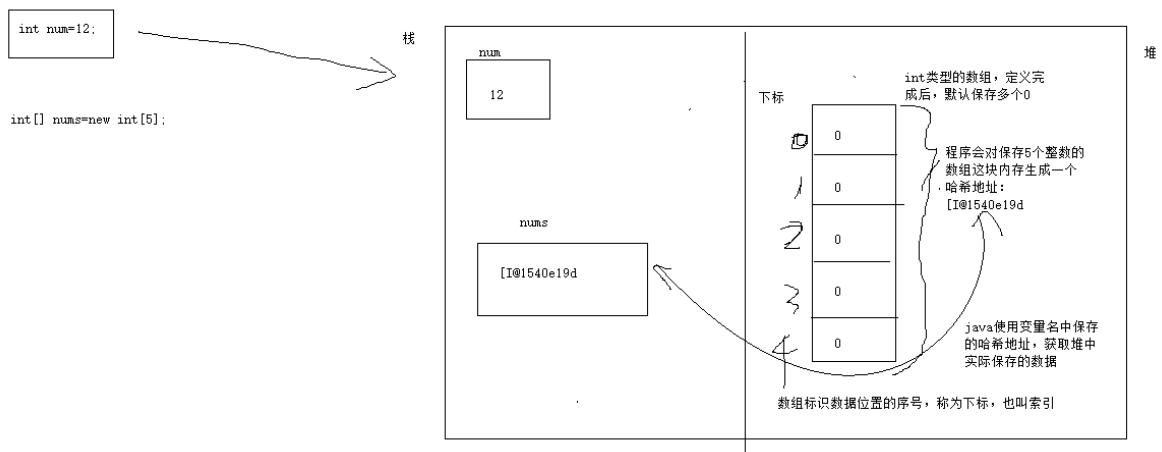
如何获取数组保存的数据个数

```
1 数组名.length //获取数组存储数据个数
```

遍历数组语法

```
1 for(int i=0;i<数组名.length;i++){
2     数组名[下标]=值;
3     sout(数组名[下标]);
4 }
```

数组在内存的存储示意图



课堂案例:使用数组保存5个学生的身高

```
1 /**
2  * @Author: lc
3  * @Date: 2022/3/12
4  * @Description: 数组的循环
5  * @Version: 1.0
6  */
7 public class Demo2 {
8     public static void main(String[] args) {
9         //1.数组 数组一旦定义，并且，内存每个空间都有一个默认值 0.0
10        float[] heights=new float[5]; //保存5个浮点类型的值
11
12        //1-2 分别对5个内存赋值
13        heights[0]=1.81F;
14        heights[1]=1.65F;
15        heights[2]=1.78F;
16        heights[3]=1.88F;
17        heights[4]=1.90F;
18        //heights[5]=2.12F;下标越界，运行时会出现异常
19
20        //2.输出看看
21        System.out.println(heights); //变量名中存地址
22
23        System.out.println(heights[0]); //下标: [0,5) 正常输出
24        System.out.println(heights[1]); //下标: [0,5) 正常输出
25        System.out.println(heights[2]); //下标: [0,5) 正常输出
```

```

26         System.out.println(heights[3]); //下标: [0,5) 正常输出
27         System.out.println(heights[4]); //下标: [0,5) 正常输出
28         //System.out.println(heights[5]); //下标: [0,5) 下标越界, 运行时会出现异常
29     }
30 }

```

2 数组的使用

2-1 求和求平均值

```

1  /**
2   * @Author: lc
3   * @Date: 2022/3/12
4   * @Description: 数组找最值
5   * @Version: 1.0
6   */
7  public class Demo3 {
8      public static void main(String[] args) {
9          int[] arr;
10         arr=new int[]{3,4,5,6,9};
11         double sum=0; //保存和
12         for(int i=0;i<arr.length;i++){
13             sum+=arr[i];
14             System.out.println(arr[i]);
15         }
16         //求平均值 求和/个数
17         System.out.println(sum/arr.length);
18     }
19 }
20

```

2-2 输出数组数据, 格式形如: 123,456,89,34

```

1  import java.util.Scanner;
2
3  /**
4   * @Author: lc
5   * @Date: 2022/3/12
6   * @Description: 求最值 N个学生的身高, 平均身高 最高 最矮
7   * @Version: 1.0
8   */
9  public class GetMaxAndMin {
10     public static void main(String[] args) {
11         Scanner input = new Scanner(System.in);
12
13         //保存N个学生身高
14         System.out.println("请输入要统计的身高学生人数: ");
15         int n=input.nextInt(); //人数
16         //动态: jvm给默认值 保存N个0.0
17         double[] heights=new double[n];
18
19         //提示输入每个学生的身高且存入数组中。
20         //经验: 一个for循环只做一件事 for赋值数据获取 for求和 for 求最值
21         for(int i=0;i<heights.length;i++){
22             //heights[下标]=实际录入的学生的身高;

```

```

23         System.out.println("请输入第" + (i + 1) + "个学生的身高: ");
24         heights[i] = input.nextDouble();
25     }
26
27     //身高表打印,使用一行输出所有的数据,格式: 1.78,1.91,1,56
28     for (int i = 0; i < heights.length; i++) {
29         System.out.print(heights[i]);
30         if (i == heights.length - 1) {
31             break;
32         }
33         System.out.print(", ");
34     }
35 }
36 }
37

```

2-3 求最大值和最小值

```

1  import java.util.Scanner;
2
3  /**
4   * @Author: lc
5   * @Date: 2022/3/12
6   * @Description: 求最大值 N个学生的身高, 平均身高 最高 最矮
7   * @Version: 1.0
8   */
9  public class GetMaxAndMin {
10     public static void main(String[] args) {
11         Scanner input = new Scanner(System.in);
12
13         //保存N个学生身高
14         System.out.println("请输入要统计的身高学生人数: ");
15         int n = input.nextInt(); //人数
16         //动态: jvm给默认值 保存N个0.0
17         double[] heights = new double[n];
18
19         //提示输入每个学生的身高且存入数组中。
20         //经验: 一个for循环只做一件事 for赋值数据获取 for求和 for求最值
21         for (int i = 0; i < heights.length; i++) {
22             //heights[下标]=实际录入的学生的身高;
23             System.out.println("请输入第" + (i + 1) + "个学生的身高: ");
24             heights[i] = input.nextDouble();
25         }
26         //找最大值和最小值
27         double max = heights[0]; //不能给0, 那给多少? 真正的Double.MIN_VALUE 数组
任意跳一个数据给他
28         double min = heights[0]; //假设
29         //max和数组每一个数据进行一次比赛, 赢了就是最厉害
30         for (int i = 1; i < heights.length; i++) {
31             if (max < heights[i]) { //数组每一个数据
32                 max = heights[i];
33             }
34             if (min > heights[i]) {
35                 min = heights[i];
36             }
37         }
38         System.out.println("最高的身高是: " + max);

```

```
39     }
40 }
41
```

2-4 从数组查找指定的元素

需求：使用变量保存一个班级的学生姓名。

scanner提示用户输入要查找的学生姓名，找到的话，输出学生姓名对应的下标。找不到：查无此人

```
1  /**
2   * @Author: lc
3   * @Date: 2022/3/12
4   * @Description: 数组元素的查找
5   * @Version: 1.0
6   */
7  public class Demo4 {
8      public static void main(String[] args) {
9          //1.静态赋值，数组定义
10         String[] names={"张三丰","赵敏","谢逊","张无忌"};
11         //2.定义要找的学生姓名
12         String findName="张无忌";
13         //3.查找的功能
14         int findIndex=-1;//不能给0
15         for(int i=0;i<names.length;i++){
16             if(findName.equals(names[i])){//找到了/没找到
17                 //找到了
18                 findIndex=i;//[0,N)
19                 break;
20             }
21             //System.out.println(findName+"查无此人");
22         }
23
24         //4.输出执行结果
25         if(findIndex>=0) {
26             System.out.println(findName + "存在，位置是：" +findIndex);
27         }else {
28             System.out.println(findName + "查无此人");
29         }
30         //基于查找的数据，实现删除或修改
31     }
32 }
```

常见的数组应用的异常

1 下标越界

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Create breakpoint : 5
    at Demo2.main(Demo2.java:18)          数组索引越界异常
```

Process finished with exit code 1

课程总结

1 数组是一种用来保存多个具有相同数据类型的数据，引用数据

2 数组三种定义方式、数据获取单个数据、数组循环 基础！！！！

3 数组求和、求最值、查找指定的数据

课后作业

迷你DVD系统可以练习写写，体会数组的定义、存入数据、遍历显示数据以及数组中查找数据、操作数据。

预习安排

数组：

排序

查找（二分查找法）

二维数组 Arrays工具类常用功能