第04天 java基础语法

今日内容介绍

* 数组介绍及动态初始化
* 数组内存结构及静态初始化
* 数组常见问题及练习

# 数组介绍及动态初始化

## 数组的概述和定义格式

### 数组的概述

需求：现在需要统计某公司员工的工资情况，例如计算平均工资、找到最高工资等。假设该公司有80名员工，用前面所学的知识，程序首先需要声明80个变量来分别记住每位员工的工资，然后在进行操作，这样做会显得很麻烦。为了解决这种问题，Java就提供了数组供我们使用。

那么数组到底是什么呢?有什么特点呢?通过上面的分析：我们可以得到如下两句话：

数组是存储多个变量(元素)的东西(容器)

这多个变量的数据类型要一致

### 数组的定义格式

### 数组概念

数组是存储同一种数据类型多个元素的容器。

数组既可以存储基本数据类型，也可以存储引用数据类型。

### 数组的定义格式

格式1：数据类型[] 数组名;

格式2：数据类型 数组名[];

注意：这两种定义做完了，数组中是没有元素值的。如何对数组的元素进行初始化呢?

## 数组初始化之动态初始化

### 数组初始化概述

Java中的数组必须先初始化,然后才能使用。

所谓初始化：就是为数组中的数组元素分配内存空间，并为每个数组元素赋值。

#### 数组初始化之动态初始化

* 动态初始化：初始化时只指定数组长度，由系统为数组分配初始值。

格式：数据类型[] 数组名 = new 数据类型[数组长度];

数组长度其实就是数组中元素的个数。

举例：

int[] arr = new int[3];

解释：定义了一个int类型的数组，这个数组中可以存放3个int类型的值。

#### 案例代码一

package com.itheima\_01;

/\*

\* 数组：存储同一种数据类型的多个元素的容器。

\*

\* 定义格式：

\* A:数据类型[] 数组名;

\* B:数据类型 数组名[];

\* 举例：

\* A:int[] a; 定义一个int类型的数组，数组名是a

\* B:int a[]; 定义一个int类型的变量，变量名是a数组

\*

\* 数组初始化：

\* A:所谓初始化，就是为数组开辟内存空间，并为数组中的每个元素赋予初始值

\* B:我们有两种方式对数组进行初始化

\* a:动态初始化 只指定长度，由系统给出初始化值

\* b:静态初始化 给出初始化值，由系统决定长度

\*

\* 动态初始化：

\* 数据类型[] 数组名 = new 数据类型[数组长度];

\*/

public class ArrayDemo {

public static void main(String[] args) {

//数据类型[] 数组名 = new 数据类型[数组长度];

int[] arr = new int[3];

/\*

\* 左边：

\* int:说明数组中的元素的数据类型是int类型

\* []:说明这是一个数组

\* arr:是数组的名称

\* 右边：

\* new:为数组分配内存空间

\* int:说明数组中的元素的数据类型是int类型

\* []:说明这是一个数组

\* 3:数组的长度，其实就是数组中的元素个数

\*/

}

}

## 数组操作之获取数组中的元素

### 案例代码一

**package** com.itheima\_01;

/\*

\* 数组：存储同一种数据类型的多个元素的容器。

\*

\* 定义格式：

\* A:数据类型[] 数组名;(推荐的使用方式)

\* B:数据类型 数组名[];

\*

\* 举例：

\* int[] arr; 定义了一个int类型的数组，数组名是arr

\* int arr[]; 定义了一个int类型的变量，变量名是arr数组

\*

\* 数组初始化：

\* A:所谓的初始化，其实就是为数组开辟内存空间，并为数组中的每个元素赋予初始值。

\* B:如何进行初始化呢?我们有两种方式对数组进行初始化

\* a:动态初始化 只给出长度，由系统给出初始化值

\* b:静态初始化 给出初始化值，由系统决定长度

\*

\* 动态初始化：

\* 数据类型[] 数组名 = new 数据类型[数组长度];

\*/

**public** **class** ArrayDemo {

**public** **static** **void** main(String[] args) {

//数据类型[] 数组名 = new 数据类型[数组长度];

**int**[] arr = **new** **int**[3];

/\*

\* 左边：

\* int:说明的是数组中的元素类型是int类型

\* []:说明这是一个数组

\* arr:这是数组的名称

\* 右边：

\* new:为数组申请分配内存空间。

\* int:说明的是数组中的元素类型是int类型

\* []:说明这是一个数组

\* 3:数组长度，其实就是数组中的元素个数

\*/

//输出数组名

System.***out***.println("arr:"+arr); //[I@104c575

//通过输出数组名，我们得到了一个地址值，但是这个值对我们来说没有意义

//我要获取的是数组中的元素值，能不能获取到呢?能

//怎么获取呢?不同担心，Java已经帮你想好了

//其实数组中的每个元素是有编号的，编号从0开始，最大的编号是数组的长度-1

//通过数组名和编号的配合使用我们就可以获取数组中指定编号的元素值

//怎么配合的呢?编号的专业叫法：索引

//获取元素的格式：数组名[索引]

System.***out***.println("arr[0]:"+arr[0]);//0

System.***out***.println("arr[1]:"+arr[1]);//0

System.***out***.println("arr[2]:"+arr[2]);//0

}

}

# 数组内存结构及静态初始化

### JVM内存划分

Java 程序在运行时，需要在内存中的分配空间。为了提高运算效率，就对空间进行了不同区域的划分，因为每一片区域都有特定的处理数据方式和内存管理方式。

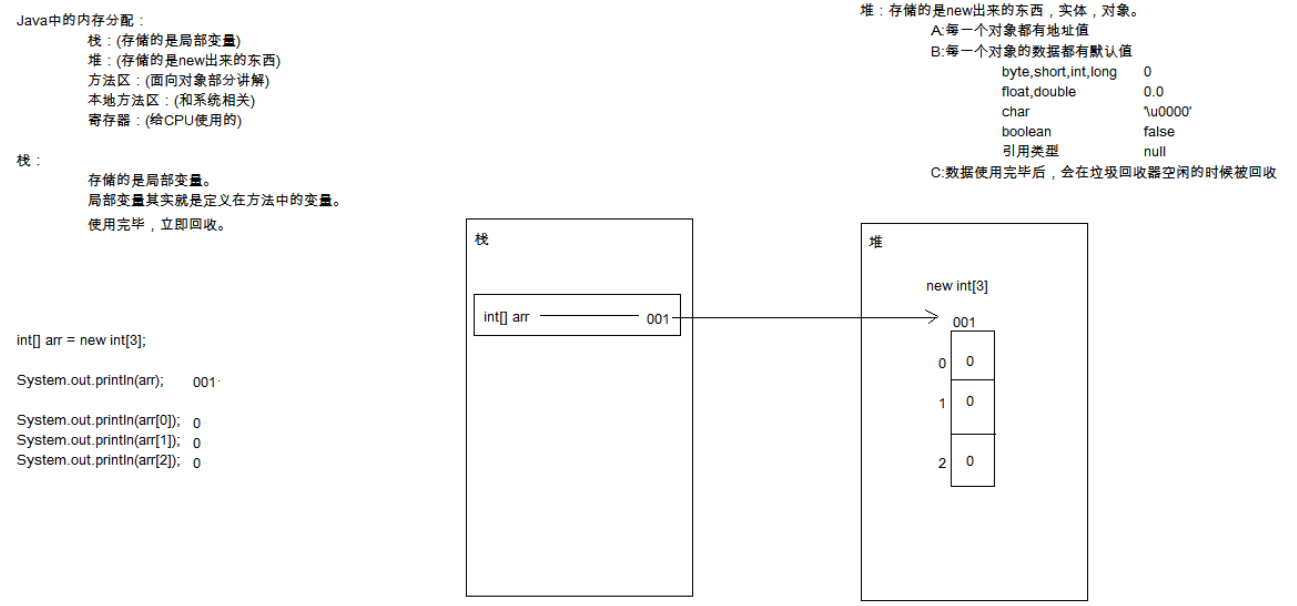
栈 存储局部变量

堆 存储new出来的东西

方法区 (面向对象进阶讲)

本地方法区 (和系统相关)

寄存器 (给CPU使用)



### 一个数组的内存图

定义一个数组，输出数组名及元素。然后给数组中的元素赋值，再次输出数组名及元素

#### 案例代码二

**package** com.itheima\_02;

/\*

\* 定义一个数组，输出数组名及元素。然后给数组中的元素赋值，再次输出数组名及元素。

\*/

**public** **class** ArrayTest {

**public** **static** **void** main(String[] args) {

//定义一个数组

**int**[] arr = **new** **int**[3];

//输出数组名及元素

System.***out***.println("arr:"+arr);

System.***out***.println("arr[0]:"+arr[0]);

System.***out***.println("arr[1]:"+arr[1]);

System.***out***.println("arr[2]:"+arr[2]);

System.***out***.println("--------------");

//然后给数组中的元素赋值

arr[0] = 100;

arr[2] = 200;

//再次输出数组名及元素

System.***out***.println("arr:"+arr);

System.***out***.println("arr[0]:"+arr[0]);

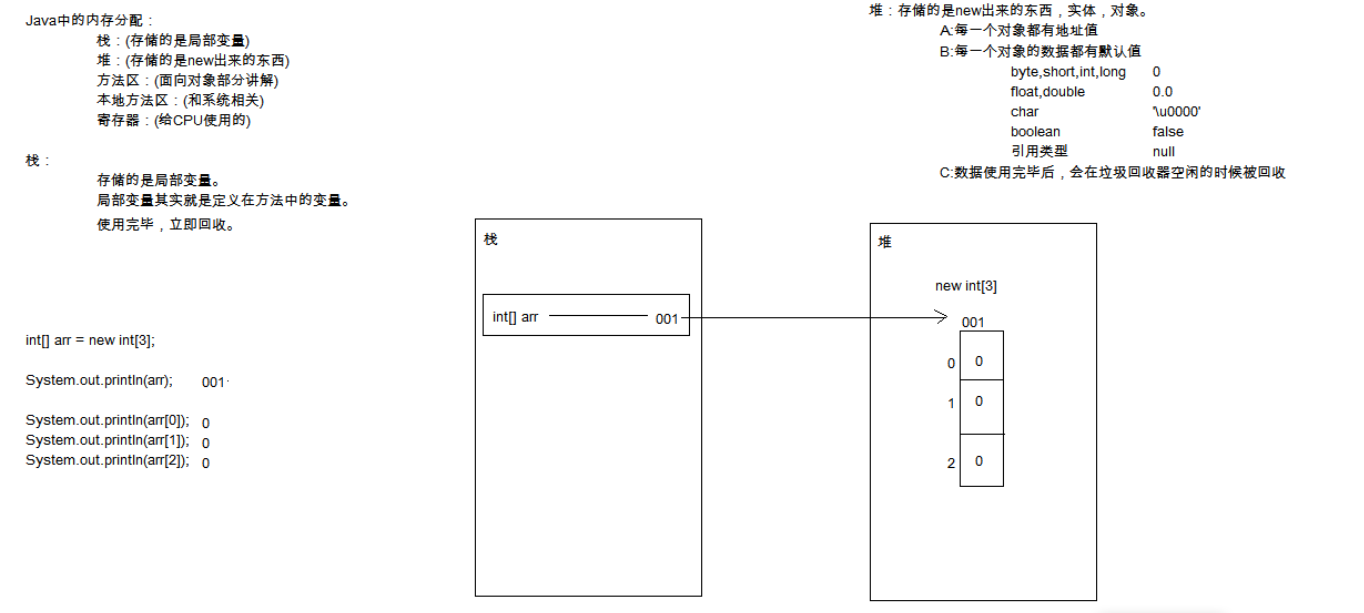
System.***out***.println("arr[1]:"+arr[1]);

System.***out***.println("arr[2]:"+arr[2]);

}

}

#### 代码内存图解



### 二个数组的内存图

定义两个数组，分别输出数组名及元素。然后分别给数组中的元素赋值，分别再次输出数组名及元素

#### 案例代码三

**package** com.itheima\_02;

/\*

\* 定义两个数组，分别输出数组名及元素。然后分别给数组中的元素赋值，分别再次输出数组名及元素。

\*/

**public** **class** ArrayTest2 {

**public** **static** **void** main(String[] args) {

//定义两个数组

**int**[] arr = **new** **int**[2];

**int**[] arr2 = **new** **int**[3];

//分别输出数组名及元素

System.***out***.println("arr:"+arr);

System.***out***.println("arr[0]:"+arr[0]);

System.***out***.println("arr[1]:"+arr[1]);

System.***out***.println("--------------");

System.***out***.println("arr2:"+arr2);

System.***out***.println("arr2[0]:"+arr2[0]);

System.***out***.println("arr2[1]:"+arr2[1]);

System.***out***.println("arr2[2]:"+arr2[2]);

System.***out***.println("--------------");

//然后分别给数组中的元素赋值

arr[1] = 100;

arr2[0] = 200;

arr2[1] = 300;

//分别再次输出数组名及元素

System.***out***.println("arr:"+arr);

System.***out***.println("arr[0]:"+arr[0]);

System.***out***.println("arr[1]:"+arr[1]);

System.***out***.println("--------------");

System.***out***.println("arr2:"+arr2);

System.***out***.println("arr2[0]:"+arr2[0]);

System.***out***.println("arr2[1]:"+arr2[1]);

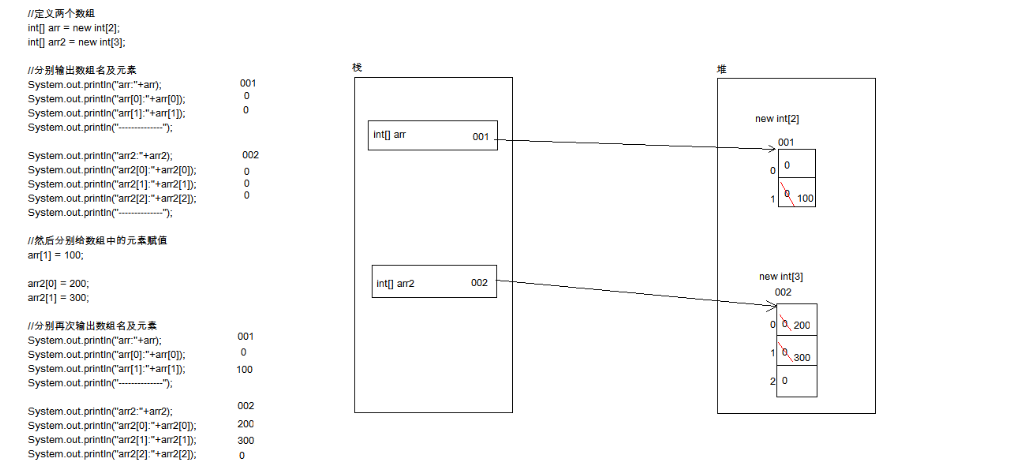
System.***out***.println("arr2[2]:"+arr2[2]);

System.***out***.println("--------------");

}

}

#### 代码内存图解



### 二个数组指向同一个堆内存的内存图

定义两个数组，先定义一个数组，赋值，输出。然后定义第二个数组的时候把第一个数组的地址赋值给第二个数组。然后给第二个数组赋值，再次输出两个数组的名及元素

#### 案例代码四

**package** com.itheima\_02;

/\*

\* 定义两个数组，先定义一个数组，赋值，输出。然后定义第二个数组的时候把第一个数组的地址赋值给第二个数组。

\* 然后给第二个数组赋值，再次输出两个数组的名及元素。

\*/

**public** **class** ArrayTest3 {

**public** **static** **void** main(String[] args) {

//先定义一个数组

**int**[] arr = **new** **int**[3];

//赋值

arr[0] = 100;

arr[1] = 200;

arr[2] = 300;

//输出

System.***out***.println("arr:"+arr);

System.***out***.println("arr[0]:"+arr[0]);

System.***out***.println("arr[1]:"+arr[1]);

System.***out***.println("arr[2]:"+arr[2]);

System.***out***.println("--------------");

//然后定义第二个数组的时候把第一个数组的地址赋值给第二个数组

**int**[] arr2 = arr;

//然后给第二个数组赋值

arr2[0] = 111;

arr2[1] = 222;

arr2[2] = 333;

//再次输出两个数组的名及元素

System.***out***.println("arr:"+arr);

System.***out***.println("arr[0]:"+arr[0]);

System.***out***.println("arr[1]:"+arr[1]);

System.***out***.println("arr[2]:"+arr[2]);

System.***out***.println("--------------");

System.***out***.println("arr2:"+arr2);

System.***out***.println("arr2[0]:"+arr2[0]);

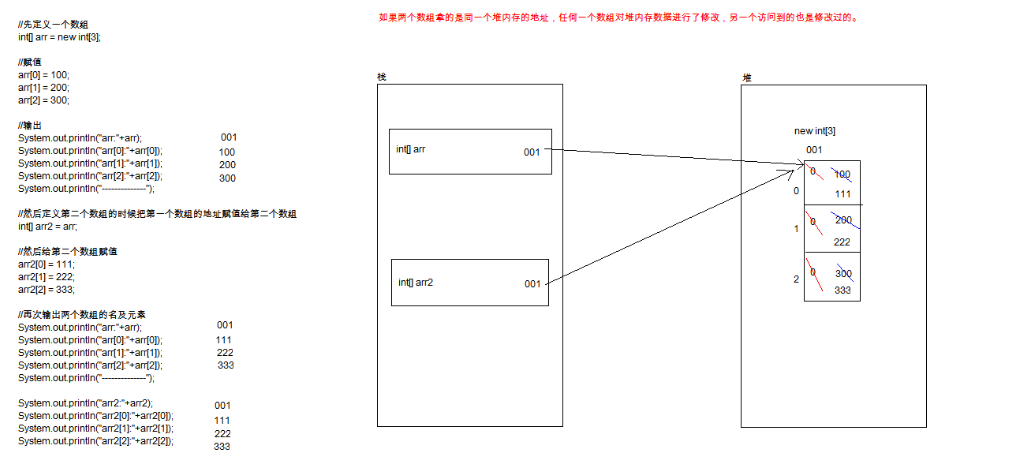
System.***out***.println("arr2[1]:"+arr2[1]);

System.***out***.println("arr2[2]:"+arr2[2]);

}

}

#### 代码内存图解



### 数组初始化之静态初始化

* 静态初始化：初始化时指定每个数组元素的初始值，由系统决定数组长度。
  + 格式：
    - 数据类型[] 数组名 = new 数据类型[]{元素1,元素2,…};
  + 举例：
    - int[] arr = new int[]{1,2,3};
    - 解释：定义了一个int类型的数组，这个数组中可以存放3个int类型的值，并且值分别是1,2,3。
    - 其实这种写法还有一个简化的写法
      * int[] arr = {1,2,3};

#### 案例代码五

**package** com.itheima\_03;

/\*

\* 静态初始化：给出数组中的元素值，由系统决定数组的长度。

\*

\* 格式：

\* 数据类型[] 数组名 = new 数据类型[]{元素1,元素2,元素3,...};

\*

\* 举例：

\* int[] arr = new int[]{1,2,3};

\*

\* 简化的格式：

\* 数据类型[] 数组名 = {元素1,元素2,元素3,...};

\* 简化格式的代码：

\* int[] arr = {1,2,3};

\*/

**public** **class** ArrayDemo {

**public** **static** **void** main(String[] args) {

//定义数组

**int**[] arr = {1,2,3};

//输出数组名和元素值

System.***out***.println("arr:"+arr);

System.***out***.println("arr[0]:"+arr[0]);

System.***out***.println("arr[1]:"+arr[1]);

System.***out***.println("arr[2]:"+arr[2]);

}

}

# 数组常见问题及练习

### 数组的元素访问

#### 案例代码八:

package com.itheima\_01;

/\*

\* 数组：存储同一种数据类型的多个元素的容器。

\*

\* 定义格式：

\* A:数据类型[] 数组名;

\* B:数据类型 数组名[];

\* 举例：

\* A:int[] a; 定义一个int类型的数组，数组名是a

\* B:int a[]; 定义一个int类型的变量，变量名是a数组

\*

\* 数组初始化：

\* A:所谓初始化，就是为数组开辟内存空间，并为数组中的每个元素赋予初始值

\* B:我们有两种方式对数组进行初始化

\* a:动态初始化 只指定长度，由系统给出初始化值

\* b:静态初始化 给出初始化值，由系统决定长度

\*

\* 动态初始化：

\* 数据类型[] 数组名 = new 数据类型[数组长度];

\*/

public class ArrayDemo {

public static void main(String[] args) {

//数据类型[] 数组名 = new 数据类型[数组长度];

int[] arr = new int[3];

/\*

\* 左边：

\* int:说明数组中的元素的数据类型是int类型

\* []:说明这是一个数组

\* arr:是数组的名称

\* 右边：

\* new:为数组分配内存空间

\* int:说明数组中的元素的数据类型是int类型

\* []:说明这是一个数组

\* 3:数组的长度，其实就是数组中的元素个数

\*/`

System.out.println(arr); //[I@3fa5ac,地址值

//我们获取到地址值没有意义，我要的是数组中的数据值，该怎么办呢?

//不用担心，java已经帮你想好了

//其实数组中的每个元素都是有编号的，编号是从0开始的，最大的编号就是数组的长度-1

//用数组名和编号的配合我们就可以获取数组中的指定编号的元素

//这个编号的专业叫法：索引

//格式：数组名[编号] -- 数组名[索引]

System.out.println(arr[0]);

System.out.println(arr[1]);

System.out.println(arr[2]);

}

}

### 数组操作的两个常见小问题

#### 案例代码六

**package** com.itheima\_04;

/\*

\* 两个常见小问题：

\* ArrayIndexOutOfBoundsException:数组索引越界异常

\* 如何产生的呢?我们访问了不存在的索引。

\*

\* NullPointerException:空指针异常

\* 如何产生的呢?null是指不再指向堆内存的数据，而我们还在访问堆内存的数据

\*/

**public** **class** ArrayDemo {

**public** **static** **void** main(String[] args) {

//定义数组

**int**[] arr = {1,2,3};

//访问数组中的元素

//System.out.println(arr[3]);

//引用数据类型：类，接口，数组

//常量：null，它是可以赋值给引用数据类型的，表示该引用不再指向堆内存的数据

arr = **null**;

System.***out***.println(arr[1]);

}

}

### 数组操作之遍历数组

#### 案例代码七

**package** com.itheima;

/\*

\* 需求：数组遍历(依次输出数组中的每一个元素)

\*/

**public** **class** ArrayOperatorDemo {

**public** **static** **void** main(String[] args) {

//定义一个数组

**int**[] arr = {11,22,33,44,55};

//原始做法

System.***out***.println(arr[0]);

System.***out***.println(arr[1]);

System.***out***.println(arr[2]);

System.***out***.println(arr[3]);

System.***out***.println(arr[4]);

System.***out***.println("------------------");

//用循环改进代码

//用for循环获取到数据0-4

**for**(**int** x=0; x<5; x++) {

System.***out***.println(arr[x]);

}

System.***out***.println("------------------");

//为了解决我们去数数组中的元素个数问题，数组就提供了一个属性：length

//用于获取数组中的元素个数

//使用格式：数组名.length

System.***out***.println("arr数组共有"+arr.length+"个元素");

System.***out***.println("------------------");

//数组遍历的标准版代码

**for**(**int** x=0; x<arr.length; x++) {

System.***out***.println(arr[x]);

}

}

}

### 数组操作之获取最值



#### 案例代码八

package com.itheima\_03;

/\*

\* 需求：数组获取最值(获取数组中的最大值最小值)

\*/

public class ArrayTest2 {

public static void main(String[] args) {

//定义数组

int[] arr = {12,98,45,73,60};

//定义参照物

int max = arr[0];

//遍历数组，获取除了0以外的所有元素，进行比较

for(int x=1; x<arr.length; x++) {

if(arr[x] > max) {

max = arr[x];

}

}

System.out.println("数组中的最大值是："+max);

}

}

### 数组练习之评委打分

#### 案例代码九

**package** com.itheima;

**import** java.util.Scanner;

/\*

\* 需求：在编程竞赛中，有6个评委为参赛的选手打分，分数为0-100的整数分。

\* 选手的最后得分为：去掉一个最高分和一个最低分后 的4个评委平均值。

\* 请写代码实现。(不考虑小数部分)

\*

\* 分析：

\* A:定义一个长度为6的数组

\* B:评委打分的数据采用键盘录入实现

\* C:写代码获取数组的最大值(最高分)

\* D:写代码获取数组的最小值(最低分)

\* E:写代码求数组中的元素和(总分)

\* F:平均分：(总分-最高分-最低分)/(arr.length-2)

\* G:输出平均分即可

\*/

**public** **class** ArrayTest {

**public** **static** **void** main(String[] args) {

//定义一个长度为6的数组

**int**[] arr = **new** **int**[6];

//评委打分的数据采用键盘录入实现

Scanner sc = **new** Scanner(System.***in***);

**for**(**int** x=0; x<arr.length; x++) {

System.***out***.println("请输入第"+(x+1)+"个评委给出的分数：");

**int** score = sc.nextInt();

arr[x] = score;

}

//写代码获取数组的最大值(最高分)

**int** max = arr[0];

**for**(**int** x=1; x<arr.length; x++) {

**if**(arr[x] > max) {

max = arr[x];

}

}

//写代码获取数组的最小值(最低分)

**int** min = arr[0];

**for**(**int** x=1; x<arr.length; x++) {

**if**(arr[x] < min) {

min = arr[x];

}

}

//写代码求数组中的元素和(总分)

**int** sum = 0;

**for**(**int** x=0; x<arr.length; x++) {

sum += arr[x];

}

//平均分：(总分-最高分-最低分)/(arr.length-2)

**int** avg = (sum - max - min)/(arr.length-2);

//输出平均分即可

System.***out***.println("该选手的最终得分是："+avg);

}

}

### 数组练习之不死神兔

#### 案例代码十

**package** com.itheima;

/\*

\* 需求：有一对兔子，从出生后第3个月起每个月都生一对兔子，小兔子长到第三个月后每个月又生一对兔子，

\* 假如兔子都不死，问第二十个月的兔子对数为多少？

\*

\* 规律：

\* 第一个月：1

\* 第二个月：1

\* 第三个月：2

\* 第四个月：3

\* 第五个月：5

\* ...

\*

\* 从第三个月开始，每个月的兔子对数是前两个月的兔子对数之和

\* 第一个月和第二个月的兔子对数都是1

\*

\* 分析：

\* A:由于数据比较多，所以我们定义数组实现

\* int[] arr = new int[20];

\* B:给数组中的元素赋值

\* arr[0] = 1;

\* arr[1] = 1;

\* C:从第三个月开始，根据规律赋值

\* arr[2] = arr[1] + arr[0];

\* arr[3] = arr[2] + arr[1];

\* arr[4] = arr[3] + arr[2];

\* ...

\* D:输出第二十个月的兔子对数，其实就是输出arr[19]

\*/

**public** **class** ArrayTest2 {

**public** **static** **void** main(String[] args) {

//定义数组

**int**[] arr = **new** **int**[20];

//给数组中的元素赋值

arr[0] = 1;

arr[1] = 1;

//从第三个月开始，根据规律赋值

**for**(**int** x=2; x<arr.length; x++) {

arr[x] = arr[x-1] + arr[x-2];

}

//输出第二十个月的兔子对数，其实就是输出arr[19]

System.***out***.println("第二十个月的兔子对数是："+arr[19]);

}

}