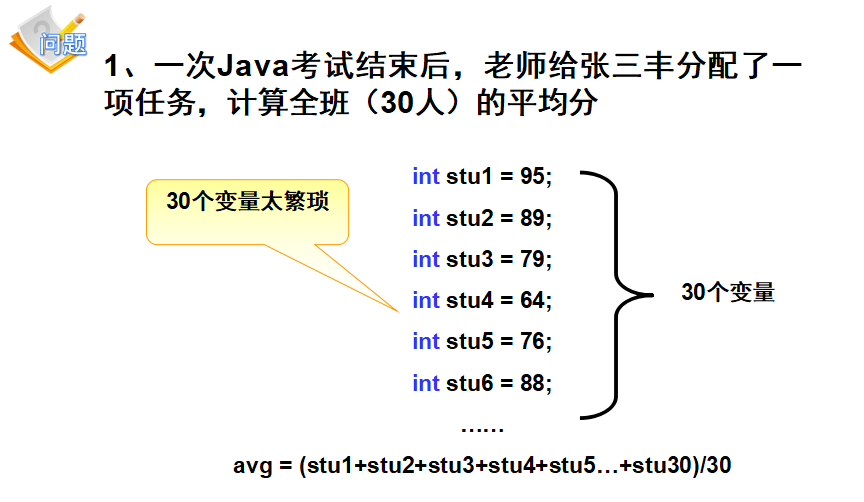
数组

山西优逸客科技有限公司官网

*http://www.sxuek.com/*

|  |  |  |
| --- | --- | --- |
| 编辑时间 | 编辑版本 | 编辑人 |
| 2018-7-17 | V2.0 | 优逸客大数据实训组 |
|  |  |  |

# 第一章 数组概述



数组的定义：

数组(Array)，是多个相同类型数据按一定顺序排列的集合，并使用一个名字命名，并通过编号的方式对这些数据进行统一管理。

数组是相同数据类型的集合，数组也是对象，数组元素相当于对象的成员变量；其实就是一个容器，和变量很像。变量只能保存一个数，而数组可以保存一组数。

数组的特点：

1）数组本身是引用数据类型，而数组中的元素可以是任何数据类型，包括基本数据类型和引用数据类型。

2）创建数组对象会在内存中开辟一整块连续的空间，而数组名中引用的是这块连续空间的首地址。

3）数组的长度一旦确定，就不能修改。

4）可以直接通过下标(或索引)的方式调用指定位置的元素，速度很快。

注意：引用类型的变量名（也叫对象名）存放在栈中，值（也叫对象）存放在堆中！

基本类型的变量名和值都存放在栈中；

数组的分类：

按照维度：一维数组、二维数组、三维数组、…

按照元素的数据类型分：基本数据类型元素的数组、引用数据类型元素的数组(即对象数组)

数组的四要素：

1、数组类型；

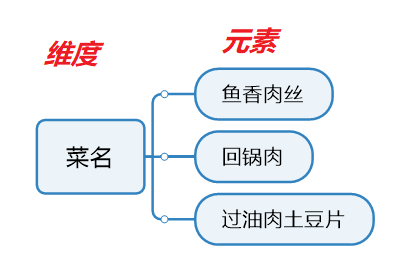
2、数组名（对象名或引用）；

3、数组元素（多个）；

4、下标（从0开始，到数组的长度-1）；

注意：访问数组的元素，不是仅仅通过数组名,而是通过数组名+下标，语法：score[下标]；

# 第二章 一维数组的使用



## 2.1 声明：

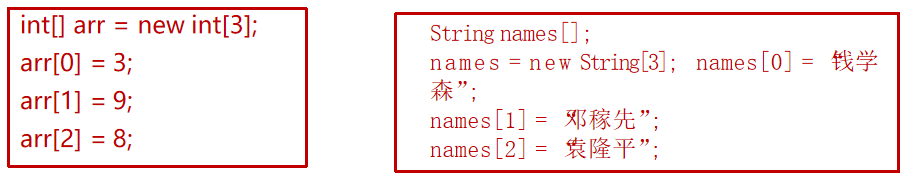
|  |
| --- |
| *type var[] 或 type[] var；*  *例如：int a[]; int[] a1; doubleb[];String[] c; //引用类型变量数组* |

注意：Java语言中声明数组时不能指定其长度(数组中元素的数)。

例如： int a[5]; //非法

## 2.2 初始化：

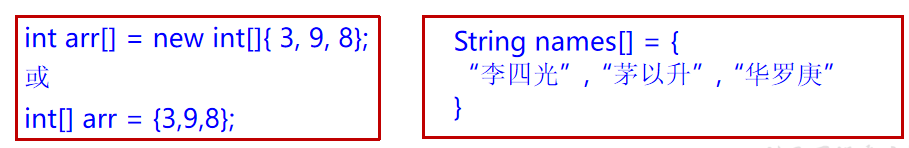
（1）动态初始化：数组声明且为数组元素分配空间与赋值的操作分开进行；



int[] arr = new int[10] ：声明一个int类型的长度为10的数组；

数组名 ：栈空间；数组元素 ： 堆空间；

（2）静态初始化：在定义数组的同时就为数组元素分配空间并赋值。



## 2.3 数组元素的引用：

（1）定义并用运算符new为之分配空间后，才可以引用数组中的每个元素；

（2）数组元素的引用方式：*数组名[数组元素下标]*

数组元素下标可以是整型常量或整型表达式。如a[3] , b[i] , c[6\*i];

数组元素下标从0开始；

长度为n的数组合法下标取值范围: 0 —>n-1；如int a[]=new int[3]; 可引用的数组元素为a[0]、a[1]、a[2]

（3）每个数组都有一个属性length指明它的长度，例如：a.length 指明数组a的长度(元素个数)。

数组一旦初始化，其长度是不可变的。

## 2.4 数组元素的默认初始化值：

数组是引用类型，它的元素相当于类的成员变量，因此数组一经分配空间，其中的每个元素也被按照成员变量同样的方式被隐式初始化。例如：

|  |
| --- |
| *public class Test {*  *public static void main(String argv[]){*  *int a[]= new int[5];*  *System.out.println(a[3]); //a[3]的默认值为0*  *}*  *}* |

对于基本数据类型而言，默认初始化值各有不同;

对于引用数据类型而言，默认初始化值为null(注意与0不同！);



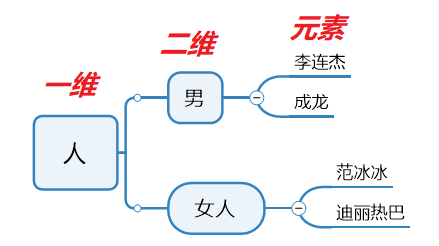
## 2.5 创建基本数据类型数组:

Java中使用关键字new来创建数组；如下是创建基本数据类型元素的一维数组

## 2.6 一维数组的内存解析：

## 2.7 一维数组案例练习：

# 第三章 多维数组的使用



## 3.1 声明及初始化：

Java 语言里提供了支持多维数组的语法。如果说可以把一维数组当成几何中的线性图形，那么二维数组就相当于是一个表格，像右图Excel中的表格一样。

对于二维数组的理解，我们可以看成是一维数组array1又作为另一个一维数组array2的元素而存在。其实，从数组底层的运行机制来看，其实没有多维数组。

|  |
| --- |
| **二维数组[][]：数组中的数组** |
| **格式1（动态初始化）：int[][] arr = new int[3][2];** |
| 定义了名称为arr的二维数组 二维数组中有3个一维数组 每一个一维数组中有2个元素;  一维数组的名称分别为arr[0], arr[1], arr[2]  给第一个一维数组1脚标位赋值为78写法是：arr[0][1] = 78; |
| **格式2（动态初始化）：int[][] arr = new int[3][];** |
| 二维数组中有3个一维数组。  每个一维数组都是默认初始化值null (注意：区别于格式1）可以对这个三个一维数组分别进行初始化;  arr[0] = new int[3];  arr[1] = new int[1];  arr[2] = new int[2];  注：**int[][]arr = new int[][3]; //非法** |
| **格式3（静态初始化）：int[][] arr = new int[][]{{3,8,2},{2,7},{9,0,1,6}};** | |
| 定义一个名称为arr的二维数组，二维数组中有三个一维数组 每一个一维数组中具体元素也都已初始化  第一个一维数组 arr[0] = {3,8,2};  第二个一维数组 arr[1] = {2,7};  第三个一维数组 arr[2] = {9,0,1,6};  第三个一维数组的长度表示方式：arr[2].length; | |
| 注意特殊写法情况：int[] x,y[]; x是一维数组，y是二维数组。  注意：Java中多维数组不必都是规则矩阵形式 | |

## 3.2 二维数组内存分析：

## 3.3 二维数组案例练习：

# 第四章 数组中涉及到的常见算法

数组中涉及的常见算法：

1）数组元素的赋值；

2）求数值型数组中元素的最大值、最小值、平均数、总和等；

3）数组的复制、反转、查找(线性查找、二分法查找)；

4）数组元素的排序算法；

## 4.1 数组元素的赋值

基本类型的赋值，赋的是值（内容），其中一个变量对其更改不影响另外一个；

引用类型的赋值，赋的是地址，两个引用共同指向一个地址（对象），所以其中一个引用对其更改影响另外一个；

注意：如果希望引用类型赋值时，只赋内容，则可以使用循环赋值的方式，语法：

## 4.2 数值型数组中元素的最大值、最小值、平均数、总和

## 4.3 数组的复制、反转、查找(线性查找、二分法查找)

### 4.3.1 冒泡排序

介绍：

冒泡排序的原理非常简单，它重复地走访过要排序的数列，一次比较两个元 素，如果他们的顺序错误就把他们交换过来。

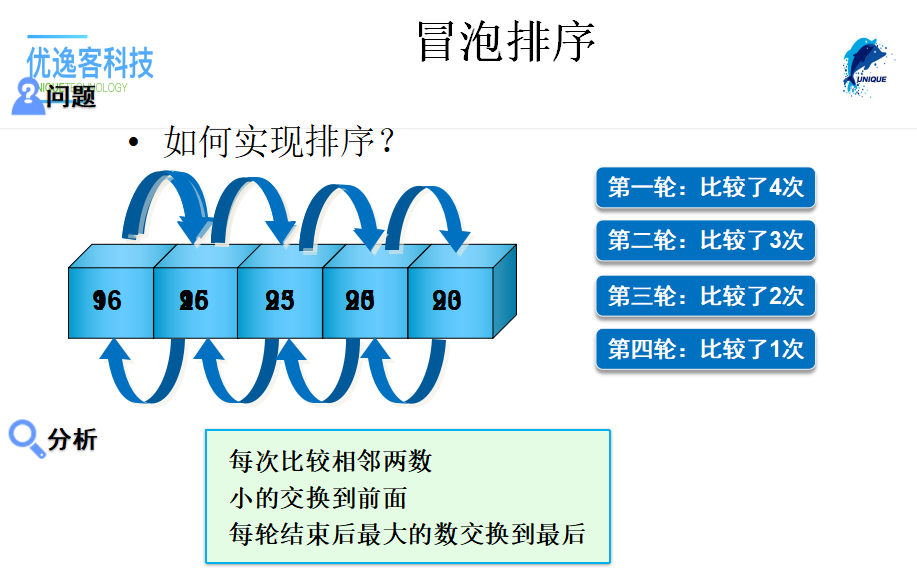
排序思想：

比较相邻的元素。如果第一个比第二个大（升序），就交换他们两个。

对每一对相邻元素作同样的工作，从开始第一对到结尾的最后一对。这步 做完后，最后的元素会是最大的数。

针对所有的元素重复以上的步骤，除了最后一个。

持续每次对越来越少的元素重复上面的步骤，直到没有任何一对数字需要 比较为止。



## 4.5 应用案例

# 第五章 Arrays工具类的使用

java.util.Arrays类即为操作数组的工具类，包含了用来操作数组（比 如排序和搜索）的各种方法。

# 第六章 数组使用中的常见异常

