

计算机等级考试二级java数组辅导 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/138/2021\\_2022\\_\\_E8\\_AE\\_A1\\_E7\\_AE\\_97\\_E6\\_9C\\_BA\\_E7\\_c97\\_138589.htm](https://www.100test.com/kao_ti2020/138/2021_2022__E8_AE_A1_E7_AE_97_E6_9C_BA_E7_c97_138589.htm) 数组（array）是相同类型变量的集合，可以使用共同的名字引用它。数组可被定义为任何类型，可以是一维或多维。数组中的一个特别要素是通过下标来访问它。数组提供了一种将有联系的信息分组的便利方法。注意：如果你熟悉C/C++，请注意，Java数组的工作原理与它们不同。

### 3.11.1 一维数组

一维数组（one-dimensional array）实质上是相同类型变量列表。要创建一个数组，你必须首先定义数组变量所需的类型。通用的一维数组的声明格式是：`type var-name[]`。其中，`type`定义了数组的基本类型。基本类型决定了组成数组的每一个基本元素的数据类型。这样，数组的基本类型决定了数组存储的数据类型。例如，下面的例子定义了数据类型为`int`，名为`month_days`的数组。`int month_days[]`。尽管该例子定义了`month_days`是一个数组变量的事实，但实际上没有数组变量存在。事实上，`month_days`的值被设置为空，它代表一个数组没有值。为了使数组`month_days`成为实际的、物理上存在的整型数组，你必须用运算符`new`来为其分配地址并且把它赋给`month_days`。运算符`new`是专门用来分配内存的运算符。你将在后面章节中更进一步了解运算符`new`，但是你现在需要使用它来为数组分配内存。当运算符`new`被应用到一维数组时，它的一般形式如下：`array-var = new type[size]`。其中，`type`指定被分配的数据类型，`size`指定数组中变量的个数，`array-var`是被链接到数组的数组变量。也就是，使用运算

符new 来分配数组，你必须指定数组元素的类型和数组元素的个数。用运算符new 分配数组后，数组中的元素将会被自动初始化为零。下面的例子分配了一个12个整型元素的数组并把它们和数组month\_days 链接起来。month\_days = new int[12]. 通过这个语句的执行，数组month\_days 将会指向12个整数，而且数组中的所有元素将被初始化为零。让我们回顾一下上面的过程：获得一个数组需要2步。第一步，你必须定义变量所需的类型。第二步，你必须使用运算符new来为数组所要存储的数据分配内存，并把它们分配给数组变量。这样Java 中的数组被动态地分配。如果动态分配的概念对你陌生，别担心，它将在本书的后面详细讨论。一旦你分配了一个数组，你可以在方括号内指定它的下标来访问数组中特定的元素。所有的数组下标从零开始。例如，下面的语句将值28赋给数组month\_days 的第二个元素。month\_days[1] = 28. 下面的程序显示存储在下标为3的数组元素中的值

。System.out.println ( month\_days [ 3 ]). 综上所述，下面程序定义的数组存储了每月的天数。// Demonstrate a one-dimensional array.class Array {public static void main(String args[]) {int month\_days[].month\_days = new int[12].month\_days[0] = 31.month\_days[1] = 28.month\_days[2] = 31.month\_days[3] = 30.month\_days[4] = 31.month\_days[5] = 30.month\_days[6] = 31.month\_days[7] = 31.month\_days[8] = 30.month\_days[9] = 31.month\_days[10] = 30.month\_days[11] = 31.System.out.println("April has " month\_days[3] " days.")}} 当你运行这个程序时，它打印出4月份的天数。如前面提到的，Java 数组下标从零开始，因此4月份的天数数组元素

为month\_days[3] 或30。将对数组变量的声明和对数组本身的分配结合起来是可以的，如下所示：`int month_days[] = new int[12]`. 这将是通常看见的编写Java程序的专业做法。数组可以在声明时被初始化。这个过程和简单类型初始化的过程一样。数组的初始化（array initializer）就是包括在花括号之内用逗号分开的表达式的列表。逗号分开了数组元素的值。Java会自动地分配一个足够大的空间来保存你指定的初始化元素的个数，而不必使用运算符new。例如，为了存储每月中的天数，下面的程序定义了一个初始化的整数数组：`// An improved version of the previous program.class AutoArray {public static void main(String args[]) {int month_days[] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31,30, 31 }.System.out.println("April has " month_days[3] " days.")}}`当你运行这个程序时，你会看到它和前一个程序产生的输出一样。Java严格地检查以保证你不会意外地去存储或引用在数组范围以外的值。Java的运行系统会检查以确保所有的数组下标都在正确的范围以内（在这方面，Java与C/C++从根本上不同，C/C++不提供运行边界检查）。例如，运行系统将检查数组month\_days的每个下标的值以保证它包括在0和11之间。如果你企图访问数组边界以外（负数或比数组边界大）的元素，你将引起运行错误。下面的例子运用一维数组来计算一组数字的平均数。`// Average an array of values.class Average {public static void main(String args[]) {double nums[] = {10.1, 11.2, 12.3, 13.4, 14.5}.double result = 0. int i. for(i=0. iresult = result + nums[i]. System.out.println("Average is " result / 5.)}}`

### 3.11.2 多维数组在Java中，多维数组

（multidimensional arrays）实际上是数组的数组。你可能期望

，这些数组形式上和行动上和一般的多维数组一样。然而，你将看到，有一些微妙的差别。定义多维数组变量要将每个维数放在它们各自的方括号中。例如，下面语句定义了一个名为twoD的二维数组变量。100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)