# 第3章 数组

数组对于每一门编程语言来说都是重要的数据结构之一，当然不同语言对数组的实现及处理也不尽相同。Java语言中提供的数组是用来存储固定大小的同类型元素。

数组：相同类型的、用一个标识符名称封装到一起的一个对象序列或基本类型数据序列。

同一种类型数据的集合。其实数组就是一个容器。运算的时候有很多数据参与运算,那么首先需要做的是什么.不是如何运算而是如何保存这些数据以便于后期的运算，那么数组就是一种用于存储数据的方式，能存数据的地方我们称之为容器，容器里装的东西就是数组的元素，数组可以装任意类型的数据，虽然可以装任意类型的数据,但是定义好的数组只能装一种元素，也就是数组一旦定义，那么里边存储的数据类型也就确定了。

数组的最大好处就是能都给存储进来的元素自动进行编号，注意编号是从0开始。方便操作这些数据。

## 3.1 一维数组

### 3.1.1 一维数组的定义与使用

为了使用数组必须在程序中声明数组，并指定数组的元素类型。

Java语言中，声明数组有两种方式：

type var[]; 或type[] var;

声明数组时不能指定其长度（数组中元素的个数），如int a[5]; //非法

Java中使用关键字new创建数组对象，new用来在内存中产生一个容器实体，数据要存储是需要有空间的，存储很多数据的空间用new操作符来开辟。格式为：

数组名 = new 数组元素的类型 [数组元素的个数]。

例如 int[] x=new int[3]; new int[3]做了两件事情，首先使用new int[3]创建了一个数组，然后把这个数组的引用赋值给数组变量x。任何一个变量都得有自己的数据类型。注意这个x不是int类型的。int代表的是容器里边元素的类型，那么x是数组类型的。数组是一种单独的数据类型，给数组分配空间时，必须指定数组能够存储的元素个数来确定数组大小，创建数组之后不能修改数组的大小。可以使用length 属性获取数组的大小。

例3.1 遍历数组。

package chapter3.e01;

public class ArrayDefinition {

public static void main(String[] args) {

// TODO Auto-generated method stub

int[] s ; // 声明变量

int i ; // 声明变量

s = new int[5] ; //创建数组对象

for(i = 0 ; i < 5 ; i++) {

s[i] = i ;

}

for(i = 4 ; i >= 0 ; i--) {

System.out.println ("" + s[i]) ;

}

}

}

例3.2 使用属性length来获取数组的长度。

package chapter3.e02;

public class ArrayLengthTest {

public static void main(String[] args) {

int[] x = { 1, 2, 3 };

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

System.out.println(x[y]);

}

}

}

### 3.1.2 数组的内存分析

int[] arr=new int[4]；

其中 = 为赋值运算符，作用是将数组对象的内存地址赋予给变量arr。

变量arr是存放在栈内存中的，栈的特点是变量一旦超出了作用域，该变量就会从内存中消失，同时释放内存空间。

new int[4]；该语句的作用是通过JVM在堆中开辟一块新的空间，创建一个新的对象，如图3.1所示。

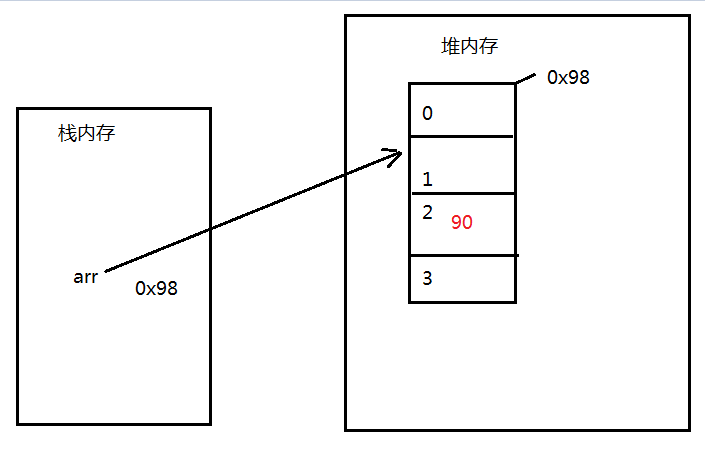


图3.1 内存中的数组

堆内存中存储的都是对象数据，对象一旦使用完毕，并不会马上从内存中消失，而是等垃圾回收器不定时收回垃圾对象后，对象才会从堆中消失并释放内存。

## 3.2 数组常见的异常

数组常见的异常有：数组索引越界、空指针等。

（1）ArrayIndexOutOfBoundsException 索引值越界。

原因：访问了不存在的索引值。注意：数组的角标从0开始。

例3.3 索引值越界产生的异常。

package chapter3.e03;

public class ArrayIndexOutOfBoundsExceptionTest {

public static void main(String[] args) {

int[] x = { 1, 2, 3 };

System.out.println(x[3]);

// java.lang.ArrayIndexOutOfBoundsException

}

}

运行结果如下：



（2）NullPointerException 空指针异常

原因： 引用类型变量没有指向任何对象，而访问了对象的属性或者是调用了对象的方法。

例3.3 空指针异常。

package chapter3.e04;

public class NullPointerExceptionTest {

public static void main(String[] args) {

int[] x = { 1, 2, 3 };

x = null;

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

}

}

运行结果如下：



## 3.3 Arrays的使用

Array是Java类库中提供的类，类中的方法用于对数组的各种操作，如：

遍历：toString()将数组的元素以字符串的形式返回

排序：sort()将数组按照升序排列

查找：binarySearch()在指定数组中查找指定元素，返回元素的索引，如果没有找到返回（-插入点-1）注意：使用查找的功能的时候，数组一定要先排序。

例3.5 Arrays的使用。

package chapter3.e05;

import java.util.Arrays;

public class ArraysTest {

public static void main(String[] args) {

int[] a = new int[] { 2, 84, 63, 18, 94, 25 };

System.out.println(Arrays.toString(a));

Arrays.sort(a);

System.out.println(Arrays.toString(a));

System.out.println(Arrays.binarySearch(a, 100));

}

}

运行结果如下：

[2, 84, 63, 18, 94, 25]

[2, 18, 25, 63, 84, 94]

-7

## 3.4 二维数组

二维数组的实质是存储一维数组的数组，定义格式如下：

数组类型[][] 数组名 = new 数组类型[一维数组的个数][每一个一维数组中元素的个数];

例3.6 二维数组的长度。

package chapter3.e06;

import java.util.Arrays;

public class TwoDArrayTest {

public static void main(String[] args) {

// TODO Auto-generated method stub

int[][] a = new int[3][4];

System.out.println(a.length);

System.out.println(a[0].length);

}

}

运行结果如下：

3

4

导致上述结果的原因如下：

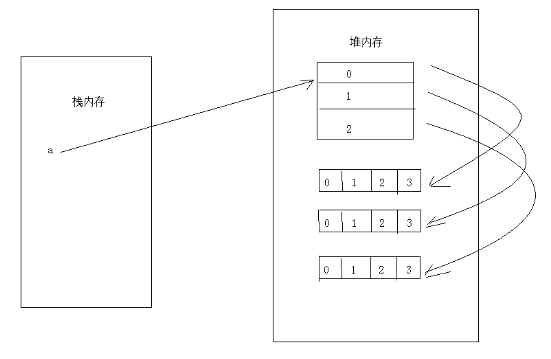


图3-2 内存中二维数组

二维数组的初始化方法有2种，一种是静态初始化，一种是动态初始化。

静态初始化如下：

int[][] x=new int[][]{0,1,2}{1,2,3};

例3.7 动态初始化二维数组。

package chapter3.e07;

public class ArrayDynamicAssignment {

public static void main(String[] args) {

int a = 0;

int[][] x = new int[2][3];

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

for (int j = 0; j < x[i].length; j++) {

x[i][j] = a++;

System.out.println(x[i][j]);

}

}

}

}

运行结果如下：

0

1

2

3

4

5

本章学习了一维数组的定义、初始化方法、数组中常见的几种异常、数组的内存分析以及二维数组等。其中数组的内存分析为理解的重点。

## 3.5 作业

### 编程题

1. 要求输入一组英文字符串数组，让这个字符串数组中前面一个元素的最后一个字母和下一个元素的首位上的字母相等，并且每个元素的长度大于等于2且小于等于100。

2. 请将”我” “爱” “北” “京”存入数组，然后正着和反着输出。