

27从零开始学Java27之二维数组更复杂

前言

配套开源项目资料

一. 多维数组

1. 概念

2. 分类

二. 二维数组

1. 概念

2. 定义

3. 初始化

3.1 动态初始化

3.2 静态初始化

4. 扩展案例

4.1 动态初始化的变形写法

4.2 遍历二维数组

5. 内存分析

三. 多维数组

四. 结语

五. 配套视频

六. 今日作业

1. 第一题

作者：孙玉昌，昵称【**一一哥**】，另外【**壹壹哥**】也是我哦

千锋教育高级教研员、CSDN博客专家、万粉博主、阿里云专家博主、掘金优质作者

前言


在前几篇文章中，壹哥给大家介绍了Java里的一维数组，涉及到了数组的创建初始化、数组遍历、拷贝、扩容、排序、查找等核心内容，这些内容都是数组中的重点，壹哥希望大家要在这些知识点上多付出点时间和精力。但是我们知道，除了有一维数组之外，还有二维数组和多维数组，这些内容同样很重要。所以今天壹哥就在一维数组的基础上，继续给大家讲解多维数组的内容。

-----前戏已做完，精彩即开始-----

全文大约【3000】字，不说废话，只讲可以让你学到技术、明白原理的纯干货！本文带有丰富案例及配图视频，让你更好地理解 and 运用文中的技术概念，并可以给你带来具有足够启迪的思考.....

配套开源项目资料

Github:



GitHub – SunLtd/LearnJava

Contribute to SunLtd/LearnJava development by creating an account on GitHub.

GitHub

Gitee:



一一哥/从零开始学Java

从零开始学Java系列 稀土掘金专栏地址: <https://juejin.cn/column/7175082165548351546> CSDN专...

Gitee

一. 多维数组

1. 概念

多维数组可以看成是数组的数组。比如二维数组就是一个特殊的一维数组，它的每个元素都是一个一维数组。其他多维数组的概念，大家可以参照二维数组以此类推。

2. 分类

多维数组可以分为：

- 二维数组；
- 三维数组；
- 其他数组；

其实我们在开发时，真正常用的数组也就是一维数组和二维数组，更多维度的数组很少使用。所以接下来在本篇文章中，壹哥主要是给大家讲解二维数组相关的内容。

二. 二维数组

1. 概念

二维数组就是一种数组的数组，其本质上还是一个一维数组，只是它的数据元素又是一个一维数组。如果你对这个概念想象不出来，壹哥给大家举个栗子，相信吸烟的同学一下子就会明白。

一根烟 = 一个变量

一包烟 = 20根烟 = 一维数组

一条烟 = 10包烟 = 二维数组

也就是说，我们买的”一条烟“，就是一种”二维数组“，明白了吧？

2. 定义

定义二维数组的基本语法如下：

定义方式一：

元素类型[][] 数组名称;

定义方式二：

元素类型 数组名称[][];

在开发时，第一种定义方式更常用一些。这是数组的定义，我们还要对数组进行初始化。二维数组在初始化时，也分为**动态初始化**和**静态初始化**两种情况。我们可以把数组的定义和初始化在多条语句中分开完成，也可以直接放在一个语句中完成。

3. 初始化

3.1 动态初始化

二维数组进行动态初始化时，语法格式如下：

```
元素类型[][] 数组名称 = new 元素类型[二维数组的长度][一维数组的长度];
```

例如：

```
int[][] arr = new int[3][4];
```

在上述示例中，表示我们定义了一个整型的二维数组arr。3表示在二维数组中有3个一维数组，每个一维数组就是一行；4表示每个一维数组中有4个数据元素，这样就形成了二维数组的4列。这样我们就可以把这个二维数组arr看成是一个**3行4列的数组或矩阵**，访问二维数组中的某个元素时，需要使用arr[row][col]的形式。所以二维数组中，总的元素个数最多有 $3*4=12$ 个。

以下是动态初始化的案例代码。

```
1  /**
2   * @author 一一哥Sun
3   * QQ: 2312119590
4   * CSDN、掘金、知乎找我哦
5   */
6  public class Demo15 {
7
8      public static void main(String[] args) {
9          // 二维数组--动态初始化
10
11          //二维数组在动态初始化时，必须指定二维数组的长度。
12          //创建一个包含了3个一维数组的二维数组，每个一维数组的长度是2。
13          int[][] arr2 = new int[3][2];
14
15          //给二维数组赋值
16          //给二维数组中的第一个一维数组赋值
17          arr2[0][0] = 111;
18          arr2[0][1] = 222;
19
20          //给二维数组中的第二个一维数组赋值
21          arr2[1][0] = 333;
22          arr2[1][1] = 444;
23
24          //给二维数组中的第三个一维数组赋值
25          arr2[2][0] = 555;
26          arr2[2][1] = 666;
27
28          //二维数组取值
29          System.out.println( "二维数组中第一个元素里的第一个元素"+arr2[0][0]);
30          System.out.println( "二维数组中第一个元素里的第二个元素"+arr2[0][1]);
31          System.out.println( "二维数组中第二个元素里的第一个元素"+arr2[1][0]);
32          System.out.println( "二维数组中第二个元素里的第二个元素"+arr2[1][1]);
33
34          //获取二维数组的长度
35          System.out.println("二维数组的长度"+arr2.length);
36          System.out.println("二维数组中第一个一维数组的长度"+arr2[0].length);
37          System.out.println("二维数组中第二个一维数组的长度"+arr2[1].length);
38          System.out.println("二维数组中第三个一维数组的长度"+arr2[2].length);
39      }
40
41  }
```

注意：我们在对二维数组进行动态初始化时，必须先指定出二维数组的长度，但一维数组的长度可以后续使用前再单独指定。

3.2 静态初始化

二维数组中，每个一维数组的长度并不要求相同，可以各不相同。二维数组进行静态初始化时，语法格式如下：

```
元素类型[][] 数组名称 = new 元素类型[][]{{一维数组1,一维数组2,一维数组3....};
```

例如：

```
int[][] arr = new int[][]{{2,3},{5,2,1},{10,45,22,54}};
```

但是这种方式的语法显得有些啰嗦，所以在实际开发时，我们一般是采用简化的静态初始化方式，如下所示：

```
元素类型[][] 数组名称 =m{{一维数组1,一维数组2,一维数组3....};
```

例如：

```
int[][] arr = {{2,3},{5,2,1},{10,45,22,54}};
```

以下是静态初始化的案例代码。

```
1  /**
2   * @author 一哥Sun
3   * QQ: 2312119590
4   * CSDN、掘金、知乎找我哦
5   */
6  public class Demo16 {
7
8      public static void main(String[] args) {
9          // 二维数组--静态初始化
10
11          //静态初始化一个二维数组。静态初始化时可以很灵活
12          int[][] arr = {{11,22,33},{66,88}};
13
14          //给二维数组赋值。
15          //重新给二维数组中第一个一维数组里的第3个数据元素(即33)赋值, 用99替换原来的3
16          3. arr[0][2] = 99;
17          //重新给二维数组中第二个一维数组里的第1个数据元素(即66)赋值, 用77替换原来的6
18          6. arr[1][0] = 77;
19
20          //二维数组取值
21          System.out.println( "二维数组中第一个一维数组里的第一个元素"+arr[0][0]);
22          System.out.println( "二维数组中第一个一维数组里的第二个元素"+arr[0][1]);
23          System.out.println( "二维数组中第二个一维数组里的第三个元素"+arr[0][2]);
24          System.out.println( "二维数组中第二个一维数组里的第一个元素"+arr[1][0]);
25          System.out.println( "二维数组中第二个一维数组里的第二个元素"+arr[1][1]);
26
27          //获取二维数组的长度
28          System.out.println("二维数组中的长度"+arr.length);
29          System.out.println("二维数组中第一个一维数组的长度"+arr[0].length);
30          System.out.println("二维数组中第二个一维数组的长度"+arr[1].length);
31      }
32
33  }
```

以上是定义和初始化二维数组的基本案例, 为了加深大家对二维数组的理解, 接下来壹哥再设计几个代码案例, 巩固二维数组的用法。

4. 扩展案例

4.1 动态初始化的变形写法

如果我们在定义二维数组时，没有给定一维数组的长度，那么在使用数组之前，一定要先对一维数组进行初始化。否则，此时如果直接使用一维数组，就会产生空指针异常

java.lang.NullPointerException。

```
1  /**
2   * @author 一一哥Sun
3   * QQ: 2312119590
4   * CSDN、掘金、知乎找我哦
5   */
6  public class Demo17 {
7
8      public static void main(String[] args) {
9          //动态初始化时未限定一维数组的长度
10         int[][] arr = new int[3][];
11
12         //如果我们在定义二维数组时，没有给定一维数组的长度，那么在使用数组之前，一定要
13         //初始化里面的一维数组。
14         //否则，如果直接使用一维数组，就会产生空指针异常java.lang.NullPointerException
15         // arr3[0][0] = 100;
16         //初始化一维数组
17         arr[0] = new int[2];
18         arr[1] = new int[1];
19         arr[2] = new int[4];
20
21         //给数组赋值
22         arr[0][0] = 1;
23         arr[0][1] = 2;
24         arr[1][0] = 10;
25         arr[2][0] = 100;
26         arr[2][1] = 200;
27         arr[2][2] = 300;
28         arr[2][3] = 400;
29
30         // 二维数组取值
31         System.out.println("二维数组中的第一个元素中的第一个元素" + arr[0][0]);
32         System.out.println("二维数组中的第一个元素中的第二个元素" + arr[0][1]);
33         System.out.println("二维数组中的第二个元素中的第一个元素" + arr[1][0]);
34     }
35 }
```

arr[0]=new int[2] 和 arr[1]=new int[1] 是在为最高维分配引用空间，也就是限制了最高维能保存数据的最大长度。

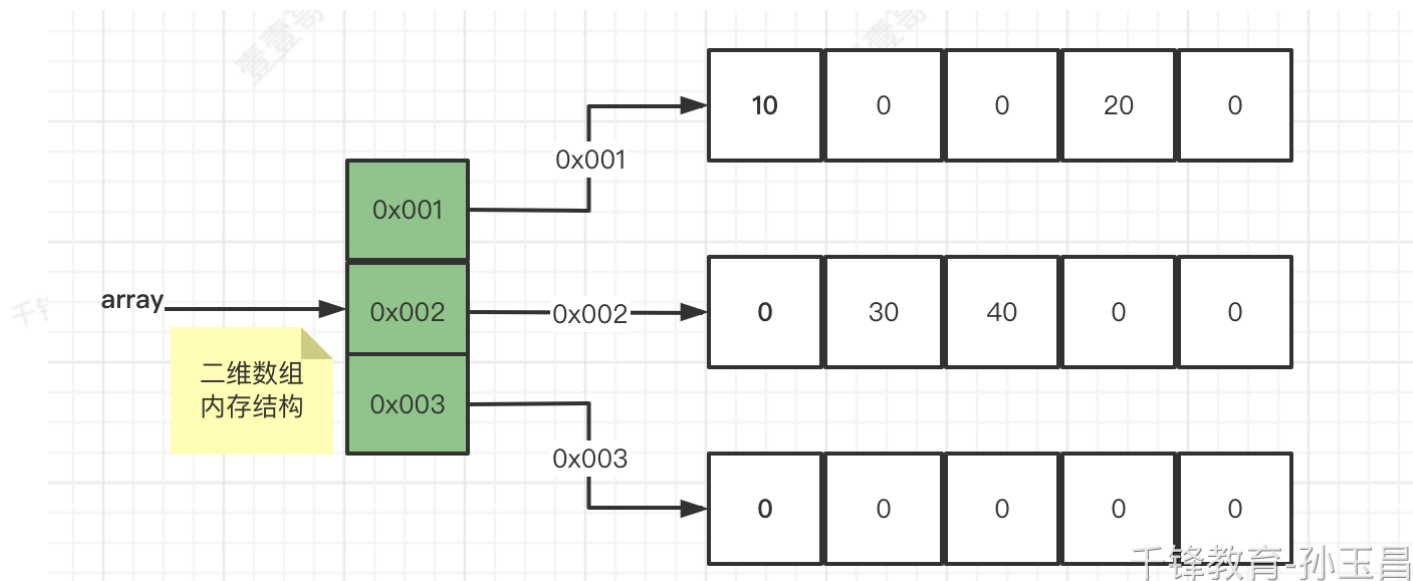
4.2 遍历二维数组

因为二维数组中包含若干个一维数组，我们之前遍历一维数组时要使用一个循环，所以我们对二维数组进行遍历时就要使用嵌套的双重循环。当然，for/while/foreach等循环都可以使用。

```
1  /**
2   * @author 一一哥Sun
3   * QQ: 2312119590
4   * CSDN、掘金、知乎找我哦
5   */
6  public class Demo18 {
7
8      public static void main(String[] args) {
9          // 二维数组--遍历
10
11         //定义一个二维数组
12         int[][] arr = {{11,22,33},{55,66},{77,88,99}};
13
14         //使用双重for循环，遍历二维数组
15         //第一层循环：遍历二维数组中有几个一维数组
16         for (int i = 0; i < arr.length; i++) {
17             //arr[i]是一维数组
18             //第二层循环：遍历每个一维数组
19             for(int j = 0 ; j < arr[i].length ; j++) {
20                 System.out.println("二维数组中第" + (i+1) + "个一维数组里的元素" + ar
21                 r[i][j]);
22             }
23         }
24
25     }
```

5. 内存分析

对于一个二维数组来说，比如 `int[][] array={{10,0,0,20,0},{0,30,40,0,0},{0,0,0,0,0}}`；这个二维数组的结构如下图所示：



array[0]会获取array数组的第1个元素，因为array数组的每个元素都是一个一维数组，因此array[0]指向的数组就是{ 10, 0, 0, 20, 0 }，其他的以此类推。

高维数组中的每个数据元素，保存的是低位数组的首地址。所以我们访问array[0]就相当于是在访问二维数组中的第一个一维数组的首地址，这样就可以得到第一个一维数组。

三. 多维数组

理论上，在Java中我们可以定义出任意维度的数组。但高于二维的多维数组，在实际开发中很少使用，大家知道Java中支持这种数据结构即可。如果你感兴趣，可以尝试着自己构造一个三维数组出来，壹哥就不在这里过多讲解啦，可以参考如下：

```
1 //动态初始化
2 int[][][] arr= new int[3][3][2];
3
4 //静态初始化
5 int[][][] arr2= {
6     {
7         {1,2,3},{4,5,6},{7,8,9}
8     },
9     {
10        {1,2},{3,4}
11    },
12    {
13        {1,2},{3,4,5}
14    }
15 };
```

-----正片已结束，来根事后烟-----

四. 结语

至此，壹哥就把二维数组的内容给大家介绍完了，现在你知道如何使用二维数组了吗？二维数组的定义、动态和静态初始化、内存分析都是今天的重点，大家要牢牢掌握哦。

在下一篇文章中，壹哥会给大家讲解Java里的可变参数，敬请期待哦。另外如果你独自学习觉得有很多困难，可以加入壹哥的学习互助群，我们一起交流学习。

五. 配套视频

如果你不习惯阅读技术文章，或是对文中的技术概念不能很好地理解，可以来看看壹哥帮你筛选出的视频教程。与本文配套的Java学习视频，链接如下：

<https://player.bilibili.com/player.html?bvid=BV1Ja411x7XB&p=65&page=65>

六. 今日作业

1. 第一题

千锋教育-孙玉昌

利用二维数组，打印出一个杨辉三角形来。