

二维数组的定义

基本与一维数组类似

//定义一个3行5列的二维数组

//方法1，先new对象，然后再初始化每个元素

```
int[][] a = new int[3][5];
```

```
a[0][0]=1;
```

```
a[0][1]=2;
```

```
a[0][2]=3;
```

//方法2，直接赋初值来创建对象

```
int[][] b = {{1, 1, 1, 1, 1}, {2, 2, 2, 2, 2}, {3, 3, 3, 3, 3}};
```

//方法3，new完对象直接初始化

```
int[][] a = new int[][] {{1, 1, 1, 1, 1}, {2, 2, 2, 2, 2}, {3, 3, 3, 3, 3}};
```

定义二维数组必须指定其行数，列数可以指定，可以不指定。

这样定义是正确的：`int[][] d = new int[3][];`

这样定义是错误的：`int[][] d = new int[][4];` `int[][] d = new int[][];`

也可以定义不规则数组：

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

```
arr[0] = new int[3];
```

```
arr[1] = new int[5];
```

3 数组的长度

`length`是数组的一个属性(不是方法!), 对于一维数组`int[] b = {1,2,3}`; `b.length`的值是3,

对于二维数组, 首先研究一下它的配置细节: `int[][] arr = new int[2][3]`;

上面这个程序片段中, 配置了 2 行 3 列的二维数组对象, 由于数据类型是 `int` , 所以数组元素的预设元素为 0 。

其实 `arr[0]` 、 `arr[1]` 是两个一维数组对象, 其长度各为 3 , 而 `arr` 类型是 `int[] []` , 内容值为 `arr[0]` 与 `arr[1]` 。其关系如图1 所示。

图 1 二维数组的配置关系

从图 1 中可以看到, `arr` 引用 `int[] []` 类型的对象, 内容包括 `arr[0]` 与 `arr[1]` , 而 `arr[0]` 与 `arr[1]` 再分别引用一个一维数组对象。

因此 `arr.length` 与 `arr[0].length` 所表示的长度意义是不同的, `arr.length`表示的是数组的行数, `arr[i].length`表示的是指定行所包含的元素数。本例中, `arr.length=2, arr[0].length=3`。