

## 二维数组

定义：数组的每一个元素又是一个一维数组

格式：

方式一：数据类型[][] 数组名=new 数据类型[一维数组的个数][每个一维数组元素的个数];

通过该种方式定义可以直接指定一维数组的个数以及每个一维数组元素的个数

`int [][] arr=new int[2][3];`//二维数组中包含两个一维数组，每个一维数组中包含三个元素

方式二：数据类型[][] 数组名=new 数据类型[一维数组的个数][];

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

该方式只指定了二维数组中有几个一维数组并没定义一维数组中有几个元素。

数组名[一维数组的下标]=new 数据类型[当前一维数组的元素个数];

获取

数组名[一维数组的下标][当前一维数组中元素的下标]

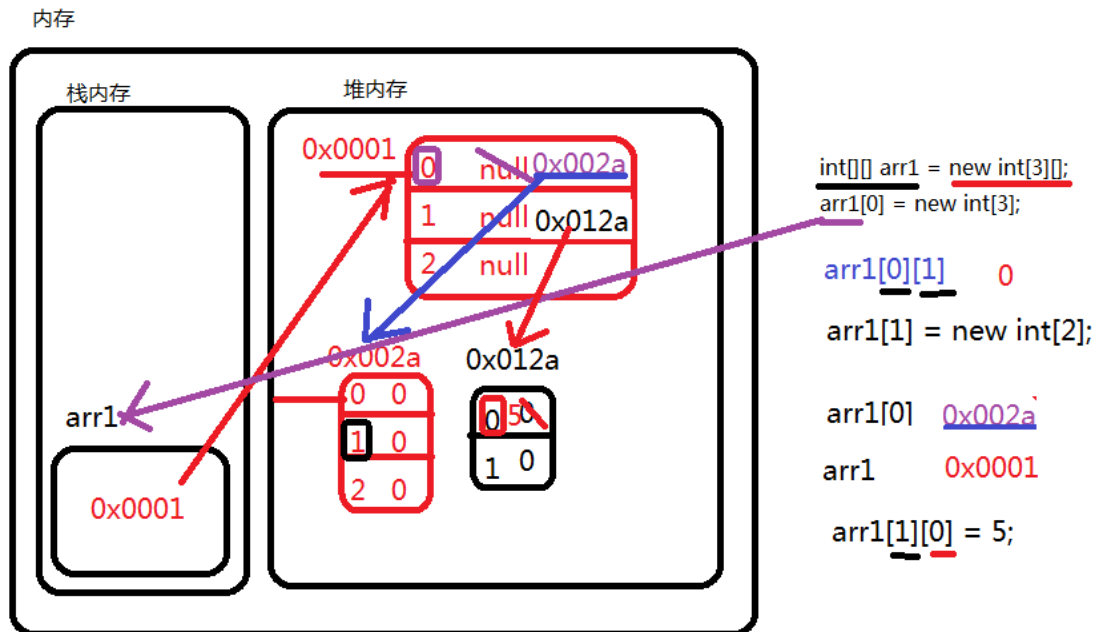
方式三：

数据类型[][] 数组名={{元素 11,元素 12,...}, {元素 21, 元素 22, 元素 23...}....}

该方式直接给出了元素的值，并且给定了一维数组的个数和元素的个数

```
int [][] arr2={{1,2,3}{5,6,3}};
```

二维数组内存图:



1. 遍历二维数组:

如何获取一维数组的个数: 二维数组名.length

如何获取一维数组的元素的个数: 二维数组名[一维数组的下标].length

也可以使用增强 for 循环, 注意外层循环的情况

2. 求二维数组所有元素的和

案例: 杨辉三角

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

获取用户从键盘输入的行数，打印相应行数的杨辉三角。

```
public static void test1(){
    Scanner sc=new Scanner(System.in);
    System.out.println("请输入一个数");
    int num=sc.nextInt();
    int[][] arr=new int[num][];
    for (int i = 0; i < num ; i++) {
        arr[i]=new int[i+1];
        for (int j = 0; j < i+1 ; j++) {
            if(j==0 || j==i){
                arr[i][j]=1;
            }else{
                arr[i][j]=arr[i-1][j]+arr[i-1][j-1];
            }
        }
    }
    for (int[] arr1:arr) {
        for (int i:arr1) {
            System.out.print(i);
        }
        System.out.println();
    }
}
```