

数组知识扫盲(Array)

本总结分为两部分，第一部分为数组知识扫盲，主要来源于李刚《Java程序员基本修养》；第二部分主要关于数组算法，主要来自于leetcode算法题目和网上的一些博客。

- 数组知识
 - Java数组的基本语法
 - Java数组的静态特性
 - Java数组内存分配机制
 - 初始化Java数组的两种方式
 - 初始化基本类型/应用类型数组内存分配
 - 数组引用变量和数组对象
 - 多维数组的内存分配
- 算法

数组知识

数组这种结构，在我们的编程过程中经常会用到，但是最近阅读李刚《Java程序员基本修养》才发现，小小数组也有大知识。Never say you know Java.

1. 数组必须经过初始化才能使用。所谓初始化，指的就是创建**数组对象**，也就是在内存中为**数组分配空间**，并为每个数组指定初始值。值得注意的是，数组一旦初始化之后，数组长度以及在内存中分配的空间是固定的，不可变的。
2. 数组的初始化有两种方式。
 - 静态初始化：初始化时不指定数组长度，系统会自动的根据程序员显示的赋值来决定数组的长度；
 - 动态初始化：初始化时候由程序员指定长度，但是初始值由系统自动分配。

1. 静态初始化：

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

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

2. 动态初始化：

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

动态初始化中，对于不同的数据类型，系统默认分配的值不同。

byte, int, short, long是0;

float, double是0.0

char是'\u0000'

boolean是false

引用类型一律都是null

3. 注意不要同时使用静态初始化和动态初始化，也就是说不要同时显式定义元素初始值和数组长度

3. 数组一定要初始化么？

不一定。因为在Java中，数组变量并不是数组本身，它只是一个引用变量，而是数组变量指向的在堆中数组对象。因此只要为数组变量指定堆中的数组对象，该数组变量就可以使用。

4. 数组变量和数组对象

在上面的解释中，反复提到数组对象和数组变量。请注意这两者是完全不同的两样东西。数组变量只是一个引用变量，它是存放在栈里面的，它如果没有指向任何的数组对象，那么它毫无意义；而数组对象是在堆里面的一些连续内存空间，他如果没有任何的数组变量指向它，那么就会被Java垃圾回收机制回收。而所谓的数组初始化，只是在堆中为数组分配空间并且使得栈里面的数组变量指向它。

```
int[] a; //这只是在栈里面命名了一个数组变量，又称引用变量，但是它没有指向任何的数组对象。  
int[] a = new int[5]; //这是在栈里面命名了一个数组变量，而且在堆里面开辟了连续空间，使得这个数组变量指向这段连续内存空间。在这里采用动态初始化，系统为数组设定默认值0。  
因此我们知道，数组变量本身不需要初始化，需要初始化的是数组对象。
```

5. 引用类型的数组

引用类型的数组其实和基本数据类型数组大体一致，只是引用类型数组变量指向的在内存中那块区域多每一个小块仍然是一个引用变量，但是这个变量是存放在堆中的，因为它毕竟是数组的元素，它还必须指向堆里面一个强制约束的数组对象。

6. Java中的多维数组 按照上面的解释，其实在Java中，并不存在真正意义上的多维数组。

```
int[][] a; //同样的这只是在栈里面命名了一个数组变量，但是系统并不知道这是你想要的二维数组。而且此时这个变量并没有指向堆里面的内存。  
int[][] a = new int[3][]; //不仅在栈里面命名了一个变量，而且指向了堆里面一段连续的内存，长度为3，每一段内存其实都是一个引用变量，但此时这段引用变量还没有指向堆里面的任何一块内存。因此此时我们可以用 a.length方法来获得数组长度。但是a[0][0]是错误的，会抛出NullPointerException异常。  
a[0] = new int[3]; //此时我们为数组中的第一块内存区域的引用变量指向了堆内存中另一段连续内存，且每一块内存存放的都是系统默认初始值0。
```