

Day04 (数组)

2018 年 1 月 18 日

8:58

Random: 用于产生随机数

使用步骤

A: 导包

```
import java.util.Random;
```

B: 创建对象

```
Random r = new Random();
```

C: 获取随机数

```
int number = r.nextInt(10);
```

获取数据的范围: [0,10) 包括 0, 不包括 10

数组: 存储同一种类型的多个元素的容器

定义格式:

A: 数据类型[] 数组名; (推荐方式)

B: 数据类型 数组名[]

数组的初始化:

所谓的初始化就是为数组开辟内存空间, 并未数组中的每个元素赋予初始值。

有两种方式实现数组的初始化:

动态初始化: 只给出长度, 由系统给出初始化值

静态初始化：给出初始化值，由系统决定长度

动态初始化：

数据类型[] 数组名 = new 数据类型[数组长度];

如：int[] arr = new int[3];

左边：

int：说明数组中的元素类型是 int 类型

[]：说明这是一个数组

arr：这是数组的名称

右边：

new：为数组申请内存分配，开辟空间

int：说明数组中的元素类型是 int 类型

[]：说明这是一个数组

3：数组的长度，其实就是数组中的元素个数

在主方法中，有四类八种基本数据，使用时不用 new，但是如

果使用别的类型时需要使用 new

访问数组的元素：

数组中的每个元素都是有编号的，索引从 0 开始，最大的索引是数组
的长度-1

通过数组名和索引的配合使用就可以指定编号的元素值

访问格式：数组名[索引]

索引就是在数组中位移几个单位

栈里放局部变量，堆里放对象

栈存储的是局部变量

栈里的数据使用完毕立即回收

局部变量：就是定义在方法中的变量

方法结束，局部变量就会被销毁

堆内存存储的是 new 出来的东西，是实体，叫对象

堆内存数据特点：

每一个对象都是有地址值的

每一个对象的数据都有默认值（byte,short,long,0;

float,double,0.0; char:'\u0000'; boolean false; 引用类型 null)

使用完毕，会在垃圾回收器空闲时回收（在栈里没有指向时回

收数据）

Java 中的内存分配

两个数组的内存图，两个引用，指向两个不同的堆空间

如果 new 里的对象在主方法未执行完前要回收堆内存，可以将栈里的指向对象的局部变量赋值 null，即可结束指向。

两个数组指向的引用同一个堆内存的地址，一个地方的改变都会改变另一个。

new 几次就有几个对象，而不是将以前的对象改变

如：

```
int[] arr = new int[3];  
System.out.println(arr);
```

arr = new int[4]; //并不是改变 arr 数组的长度，而是在堆内存

中新建了一个对象，以前的 arr 对应的地址对象被回收

```
System.out.println(arr);
```

上述代码打印出的地址值是不一样的

静态初始化：给出初始化值，由系统决定长度

静态初始化格式：

```
数据类型[] 数组名 = new 数据类型[] {元素 1, 元素 2, 元素 3, ...};
```

简化格式：

```
数据类型[] 数组名 = {元素 1, 元素 2, 元素 3, ...};
```

注意：

简化版的写法只能在申明数组的时候才可以使用，如果已经申明了数组，想要改变数组，就只能用全格式改变

动态初始化数组，数组的长度可以是变量

数组常见的小问题

`ArrayIndexOutOfBoundsException`:数组索引越界异常：

产生原因：访问了不存在的索引

`NullPointerException`:空指针异常

产生原因：数组已经不再指向堆内存的数据，还使用数组去访问元素时。

数组的遍历：依次输出数组中的每一个元素

可以使用 for 循环实现

获取数组的长度：数组名.length

数组获取最值：

获取最值的思路：

先要有参照物，参照物必须是数组中的；一般拿第一个元素

遍历数组，让后面所有的元素依次和参照物进行比较，如果元素比参

照物大（小）就留下

整个循环遍历比较结束，留下的就是最大（小）值

也可以使用索引记录数组中最大值的位置。

二维数组：其实就是元素是一维数组的数组

定义格式：

A：数据类型[][] 数组名；（推荐方式）

B：数据类型 数组名[][];

C：数据类型[] 数组名[];

初始化：

A：动态初始化

数据类型[][] 数组名 = new 数据类型[m][n];

m: 表示二维数组中一维数组的个数

n: 表示一维数组中的元素个数

B: 静态初始化

数据类型[][] 数组名 = new 数据类型[][]{{元素...},{元素...},{元素...},...};

简化格式: 数据类型[][] 数组名 = {{元素...},{元素...},{元素...},...};

二维数组名配合索引可以获取到每一个一维数组

每一个一维数字配合索引名可以获取数组中的元素

假如有一个二维数组: arr

要从中获取一维数组: arr[索引];

从中获取二维数组的元素: arr[索引][索引];

二维数组的本质是一个一维数组，它的元素是一个一维数组的地址值

二维数组的遍历

了解

System.out.println();输出内容并换行

System.out.print();只输出内容

Java中的内存分配：
栈(存储的是局部变量)
堆(存储的是new出来的东西)
方法区(面向对象进阶讲)
本地方法区(和系统相关)
寄存器(给CPU使用)

栈：
存储的是局部变量。
局部变量就是定义在方法中的变量。
使用完毕,立即回收

```
int[] arr = new int[3];  
  
System.out.println(arr);      001  
  
System.out.println(arr[0]);   0  
System.out.println(arr[1]);   0  
System.out.println(arr[2]);   0
```

堆：存储的是new出来的东西，实体，对象
A:每一个对象都有地址值
B:每一个对象的数据都有默认值
byte,short,int,long 0
float,double 0.0
char '\u0000'
boolean false
引用类型 null
C:使用完毕后,会在垃圾回收器空闲的时候被回收

