数组

数组是存储同一数据类型的多个元素的集合也可以看成是一个容器

数组可以存储基本数据类型，也可以存储引用数据类型

数组定义：

格式1：数据类型[] 数组名; int[] arr;

格式2：数据类型 数组名[]; int arr[];

两种定义格式，常用第一种

数组初始化：

Java中的数组必须先初始化，然后才能使用，初始化就是为数组中的元素分配内存空间，并为每个数组元素赋值

动态初始化：初始化时只指定数组长度，由系统为数组分配初始值

静态初始化：初始化时指定每个数组元素的初始值，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]

int[] arr = new int[3];

定义了一个int类型的数组，这个数组可以存放三个int数据

Java中内存分配

Java中的内存分配

Java程序在运行是，需要在内存中分配空间。为了提高运算效率，就对空间进行了不同区域的划分，因为每一片区域都有特定的处理数据方式和内存管理模式

栈：存储局部变量

堆：存储所有new出来的东西

方法区（重点，面向对象部分详细讲解）

本地方法区（和系统相关）

寄存器（给CPU使用）

局部变量：在方法定义中或者方法声明上的变量都称为局部变量

栈内存特点：栈内存的数据用完就释放掉（脱离作用域就会释放）

堆内存的特点：

A:每一个new出来的东西都有地址值

B:每个变量都有默认值

byte,short,int,long 默认 0

float,double 默认 0.0

char 默认 ‘\u0000’

boolean 默认 false

引用类型 默认 null

C:使用完毕就会变成垃圾，但是并没有立即回收，会在垃圾回收器空闲的时候回收

数组在内存中的概述：（刘意课程Day5 17-19）

int[] arr = new int[2];

int[] arr 这个是局部变量，是存储在栈内存中的，如果直接输出arr,得到的则是一个地址值，这个地址值指向的是堆内存中对应的new出来的值的地址

new int[2]这个是new出来的，存放在堆内存中。而这个堆内存的地址则是给了栈内存中的arr

int[] arr = new int[3];

int[] arr2 = arr ;

上面的例子中，把arr给了arr2，实则是把arr的地址给了arr2，所以访问arr2[n]的时候，访问的是arr中的元素

栈内存的两个引用指向同一个堆内存空间，无论是他们谁的操作，都是针对同一个地方的操作

数组静态初始化：只给元素，不给数组长度

数据类型[] 数组名 = new 数据类型[]{元素1,元素2,…};

数据类型[] 数组名 = {元素1,元素2,…}

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

int[] arr = {1,2,3};//后面大括号里也是存放在堆内存里

数组操作的两个常见小问题：

数组越界:索引大于数组长度报错数组索引越界异常

原因：你访问了不存在的索引

空指针异常：数组已经不指向堆内存了，还用数组名去访问就报错空指针异常

数组常见练习：

数组遍历：(遍历巧用arr.length)

逐一输出；for循环输出；foreach循环遍历

int[] arr = new int[5];

for(int s : arr){

System.out.println(s);

}

数组获取最值

二维数组

理解：二维数组就是元素为一维数组的一个数组

格式：

数据类型[][] 变量名 = new 数据类型[m][n];

int[][] arr = new int[3][2];

理解：定义了一个二维数组arr，这个二维数组有三个一维数组，名称是arr[0],arr[1],arr[2]。每个一维数组有2个元素

数据类型[] 变量名[] = new 数据类型[m][n];

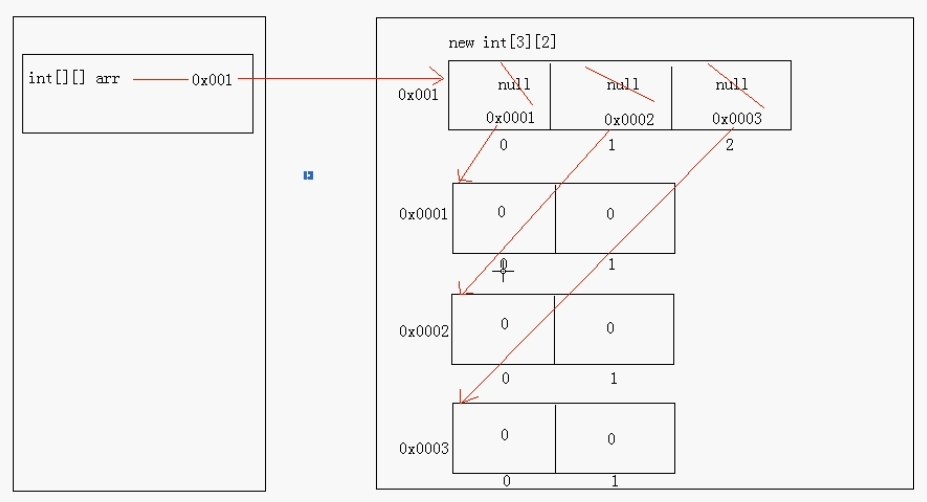
上述也是定义一个二维数组

二维数组内存模型详解：

int[][] arr = new int[3][2];

int[][] arr存储在栈内存中

new int[3][2]存放在堆内存中，而在堆内存中，将int[3]看做是一维数组，有三个元素，每个元素中存放的是下一级一维数组的地址值

格式2：

数据类型[][] 变量名 = new 数据类型[m][];

m表示这个二维数组有多少个一维数组，这一次没有直接给出一维数组的元素个数，可以动态的给出，举例

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

arr[0] = new int[2];

arr[1] = new int[3];

arr[2] = new int[5];

格式3：

数据类型[][] 变量名 = {{元素…},{元素…},{元素…}};

int[][] arr = {{1,2,4},{5,7},{8}};

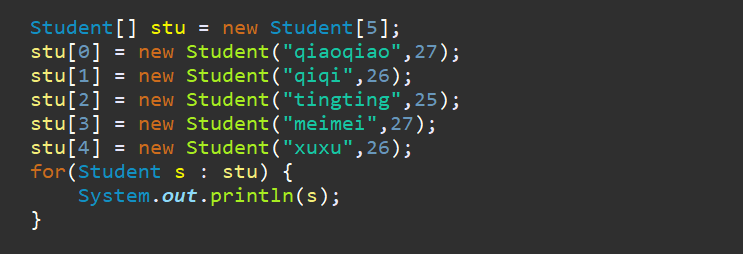
Java参数传递问题（刘意课程 day6\_10）

Java中只有值传递

基本类型传递的是数据值。实参把自己的值拷贝给形参，无论形参怎么变化，都无法改变实参本身

引用类型传递的是地址值。实参把自己所指向的地址给形参，所以形参可以操作地址所指向的堆内存中的数据

对象数组的概述和使用



Arrays类概述及其常用方法

public class Arrays extends [Object](mk:@MSITStore:G:\JAVA\API\JDK_API_1_6_zh_CN.CHM::/java/lang/Object.html)

此类包含用来操作数组（比如排序和搜索）的各种方法。此类还包含一个允许将数组作为列表来查看的静态工厂。

Arrays类概述（具体看API）

针对数组进行操作的工具类

提供了排序，查找等功能

public static String toString(int[] array)

把数组转成字符串（仅以int数组举例，可以对任意数组操作）

public static void sort(int[] array)

把数组进行排序

public static int binarySearch(int[] arr,int key)

对数组进行二分查找

方法非常多，具体请看API