复习：

1. 判断结构：if, if-else, if-else-if --- 如果if或者else之后的代码块只有一句，那么这个时候{}可以省略

if(i > 5)

System.out.println(i);

2. 选择结构：switch-case。switch()需要一个值，这个值可以是byte/short/char/int, JDK1.7开始允许使用字符串。case之后需要一个break表示这个case的结束。---有break决定了case的顺序是否影响结果

3. 循环结构：while, do-while, for --- 用于完成重复的相同或者相似的事儿

1\*3 2\*3 3\*3

### break和continue

break:用于选择结构和循环结构，表示终止一层循环

continue:只能用于循环结构中，表示跳出本次循环继续下次

练习：

1. 百钱买百鸡： 100文钱， 3文钱1只公鸡，2文钱1只母鸡，1文钱3只小鸡 --- 100只鸡

2. 输入一个数字，判断这个数字是否是一个质数。 --- 质数：除了1和本身以外不能被别的数字整除 --- 2~n-1

因数

100 100 50

3. 将一个数字分解为质因数

120 = 2 \* 2 \* 2 \* 3 \* 5

## 数组

存储多个同一类型的数据---容器---大小固定

数组对放入其中的元素进行编号，编号从0开始的---下标

### 定义格式

数据类型[] 数组名 = new 数据类型[数组的大小]; --- 动态初始化

int[] arr = new int[5]; --- 表示定义了一个能存储5个整型元素的数组

arr[2] = 10; // 表示向数组arr的下标为2的位置上放入元素10

int i = arr[2];

数据类型[] 数组名 = new 数据类型[]{元素1，元素2，元素3 ...}; --- 静态初始化

int[] arr = new int[]{2,1,5,3,7}; ---表示定义了一个元素类型是int的数组，并且数组中的元素是2,1,5,3,7 --- 当给定了元素的时候，也就同时规定了数组的大小为5

数据类型[] 数组名 = {元素1， 元素2，元素3 ... };

int[] arr = {2,4,1,7,8,3}; --- 数组的大小是6

注意：当使用的下标超过了数组的最大下标的时候就会出现：ArrayIndexOutOfBoundsException --- 数组下标越界异常

在数组定义的时候，[]可以在数组名之前也可以在数组名之后，但是[]在数组名之前，那么声明的所有的变量实际上都是数组；如果在数组名之后只规定当前的这个变量是一个数组不影响其他的变量

### 内存

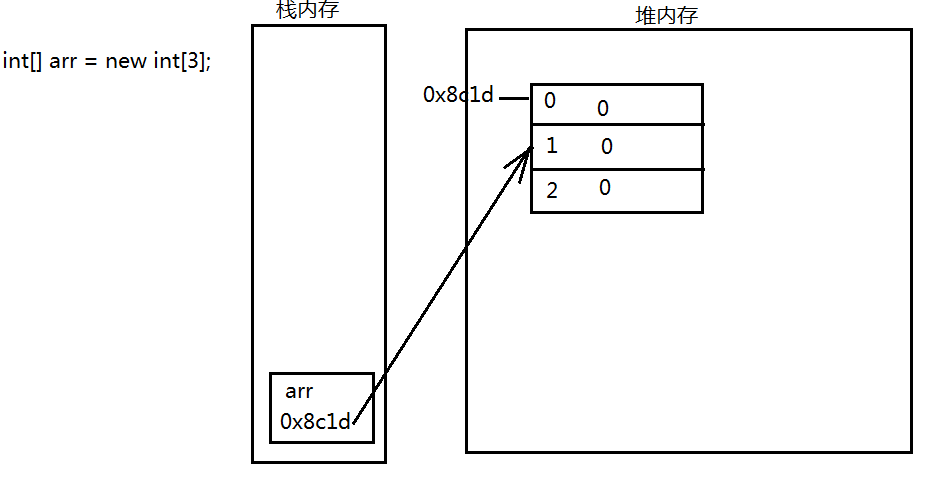
Java将所占用的内存划分为了5块：栈内存、堆内存、方法区、本地方法栈、PC计数器(寄存器).

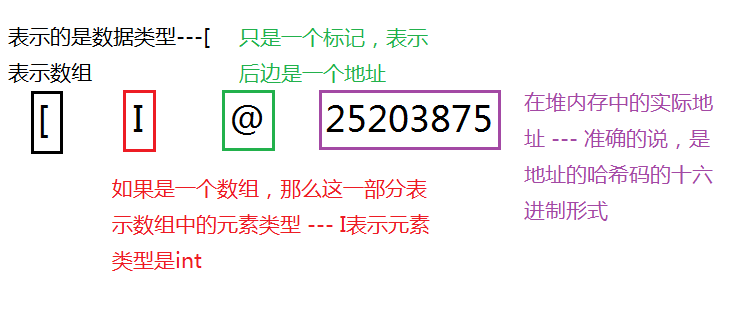
栈内存：

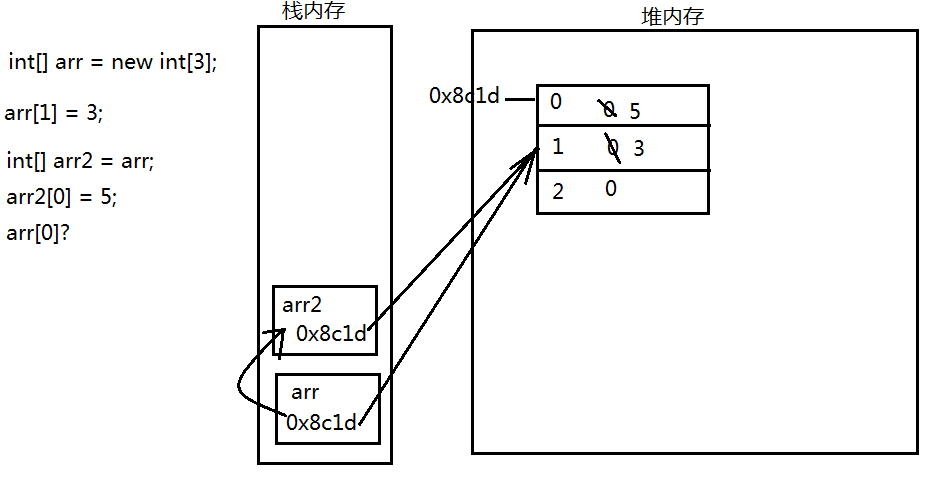
用于存储的变量。变量在定义之后会在栈内存中占用一块空间，如果在程序中给变量赋值，才会在栈内存的对应空间中保存这个值。这个变量用完以后会立即移除出栈内存。

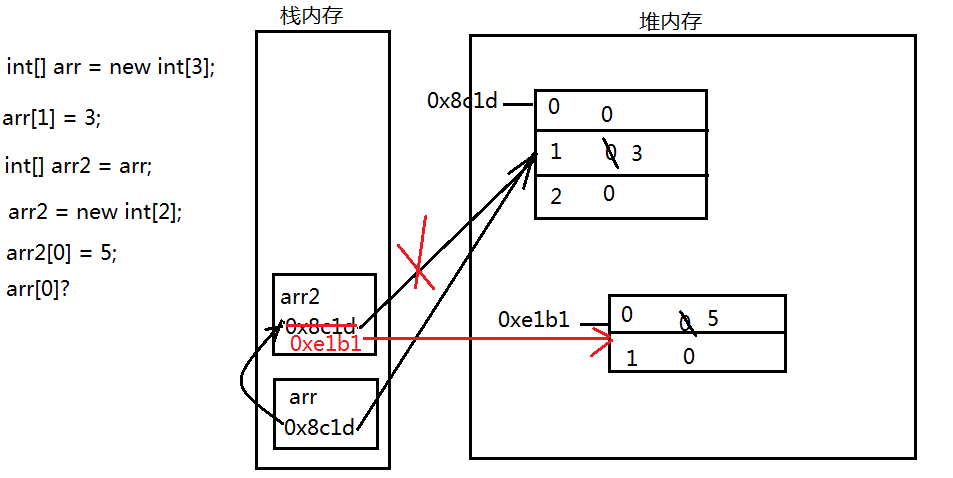
堆内存：

存储的是对象（数组也是一种对象）。堆内存会对存入其中的元素赋予一个默认值: byte/short/int - 0, long - 0L, float - 0.0f, double - 0.0, char - ‘\u0000’, boolean - false, 除了这八种基本类型以外的其他类型的默认值都是null。当存入堆内存中的对象使用完成之后，不会立即移除出堆内存而是在不定的某个时刻被释放。









### 数组的应用

1. 获取数组的元素 --- 数组名[下标]

2. 获取数组的长度 --- 数组名.length

3. 遍历数组

// 方式一：普通for循环遍历

for(int i = 0; i < arr.length; i++){

System.out.println(arr[i]);

}

// 方式二：增强for循环

for(int i : arr){

System.out.println(i);

}

// 方式三：利用Arrays

import java.util.Arrays;

// 将数组中的元素依次拿出来拼接成一个字符串来返回

String str = Arrays.toString(数组);

4. 获取数组中的最值（最大值/最小值）

5. 对数组元素进行排序 --- 升序排列

时间复杂度：找到这个程序中会重复执行的这段代码，把重复的这段代码执行的时间认为成单位1，执行这个单位1的次数就是时间复杂度 --- 时间复杂度计算的时候一定找的这个表达式的最高阶，找到最高阶之后，只考虑项不考虑系数。时间复杂度用O() --- 对于时间复杂度而言一定满足以下几种形式：(logn)x, nx, nx(logn)y

logab = logcb/logca lognx=xlogn

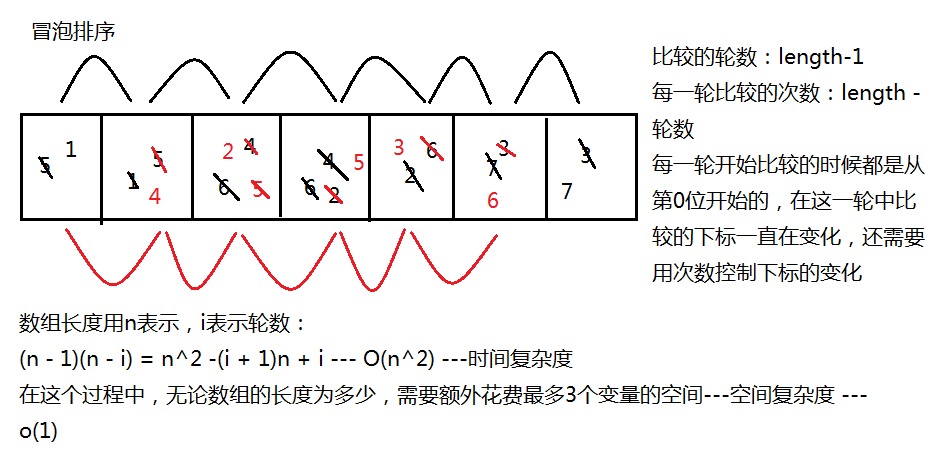
3n^2 --- O(n^2)

2n^2 --- O(n^2)

空间复杂度：这段代码在执行过程中需要额外花费的变量的空间的个数。--- 空间复杂度表现形式和时间复杂度一样 ---空间复杂度用o()表示

3 = 3 \* n0 --- o(n0) --- o(1)

注意：时间复杂度表示了代码被重复执行的次数不决定时间的长短；空间复杂度表示了这段代码执行需要耗费的内存大小



预习：

选择排序

二分查找