复习：

1. 关系运算符： ==
2. 逻辑运算符：^ && ||---短路特性
3. 位运算符：& | ^ << >> >>> ~ --- 针对的是整数的补码
4. 三元运算符： 逻辑值?表达式1 :表达式2;

总结：运算符的优先级

() ++ -- ~ ！ 算术 << >> >>> 关系 逻辑 & | ^ 三元 赋值

1. 判断结构：if,if-else, if-else if
2. 选择结构：switch-case --- byte/short/char/int,String；break

### 循环结构

#### while循环

while(逻辑值){

Code;

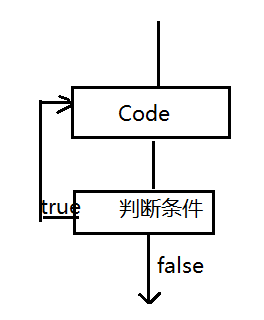
}

do{

Code;

} while(逻辑值);

无论如何代码块都会执行一次。



#### for循环

定义循环变量;

while(判断循环变量是否成立){

代码块;

改变循环变量;

}

for(定义循环变量;循环条件;改变循环变量){

代码块;

}

#### break和continue

break：可以在选择和循环结构中使用。表示终止一层循环

continue：只能用在循环结构中。表示跳出本次一层循环继续下次的循环

在Java中，break和continue支持标号形式

练习：输入一个数字，判断这个数字是否是一个质数---除了1和本身没有别的因子

int num = s.nextInt();

2~num -1

练习：打印1000以内的所有的质数

1. 先获取1-1000这1000个数字
2. 获取到这个数字之后要判断这个数字是否一个质数

练习：打印九九乘法表

1\*1=1

1\*2=2 2\*2=4

1\*3=3 2\*3=6 3\*3=9

思考：如果用一个循环打印九九乘法表？

练习：输入一个数字，分解数字的质因数

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

## 数组

存储多个同一类型的数据的容器---大小是固定的

会对数组中的元素自动的进行编号，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/长度]; --- 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] = 5; --- 表示向arr数组中下标为2的位置放入元素5

System.out.println(arr[2]);

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

int[] arr = new int[]{2,5,1,6,4,8};---同时规定了数组的长度就是6

arr[3] = 7;

数据类型[] 数组名 = {元素1, 元素2, ... , 元素n}; --- 不能分开定义

int[] arr = {2,1,6,4,3,1,8};

注意：[]在前，紧跟数据类型的；[]在后紧跟具体的变量的

### 内存

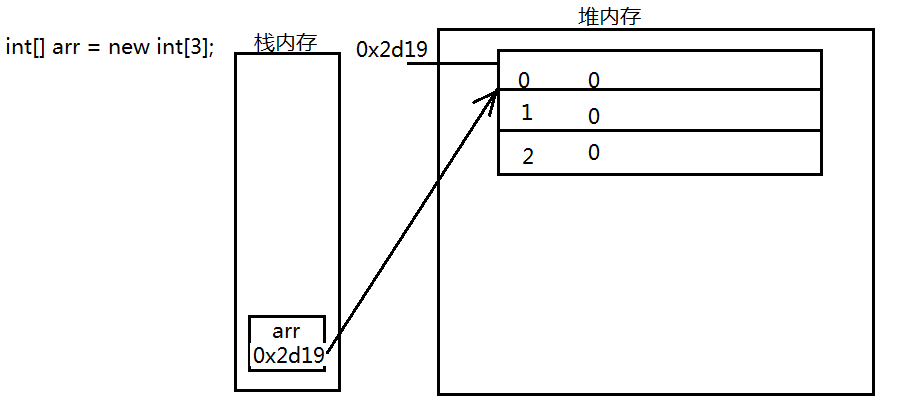
Java将内存分为了5块：栈内存、堆内存、方法区、本地方法栈、寄存器

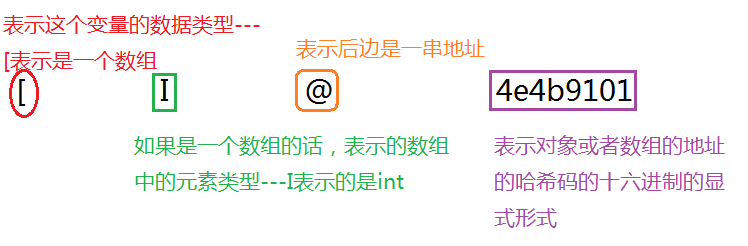
栈内存：存储变量的。栈内存不会自动给变量赋值。而且变量用完之后会从栈内存中立即移除以释放内存空间

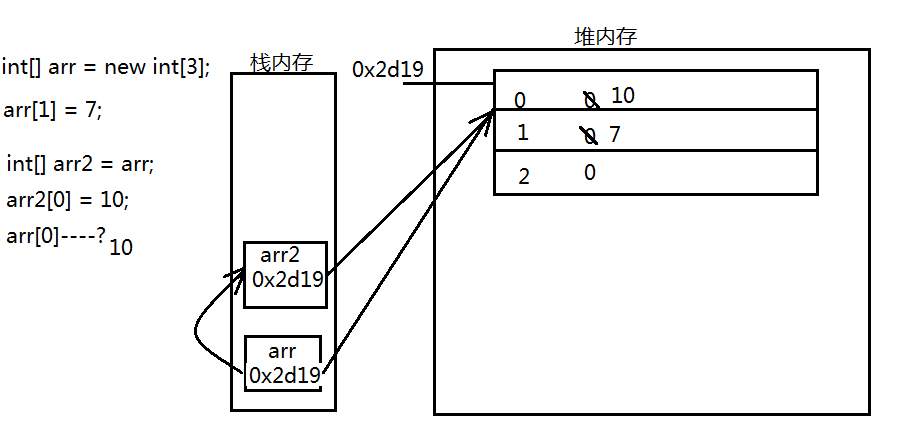
堆内存：存储的是数组或者对象。碰到new关键字的时候会到堆内存中开辟空间。堆内存会对存入其中的元素自动的赋予一个默认值---byte/short/int-0, long-0L, float-0.0f, double-0.0, char- ‘\u0000’, boolean-false, 除了这八种基本类型以外的所有的类型的默认值都是null。 堆内存中的数据用完之后会在不定的某个时刻被回收。

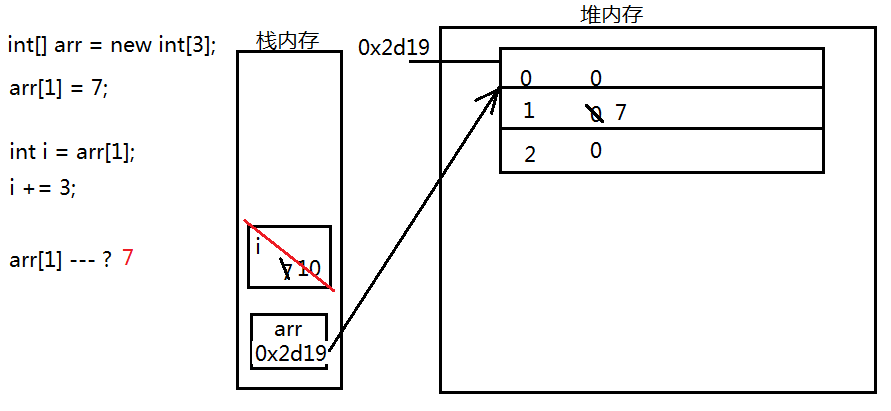
boolean[] bs = new boolean[3]; bs[0] --->false

String[] ss = new String[5]; ss[0] ---> null









如果操作的下标超过了最大下标，则会出现ArrayIndexOutOfBoundsException

### 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
3. 遍历数组

// 利用普通的for循环来遍历数组

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

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

}

// 增强for循环

for(int i : arr){

System.out.println(i);

}

Arrays.toString(arr);

1. 获取数组中的最值（最大值/最小值）

// 定义一个变量来记录最大值

int max = arr[0];

// 拿着max依次和其他的元素比较

for(int i = 1; i < arr.length; i++){

if(max < arr[i])

max = arr[i];

}

System.out.println(max);

// 定义一个变量记录最大值的下标

int max = 0;

for(int i = 1; i < arr.length; i++){

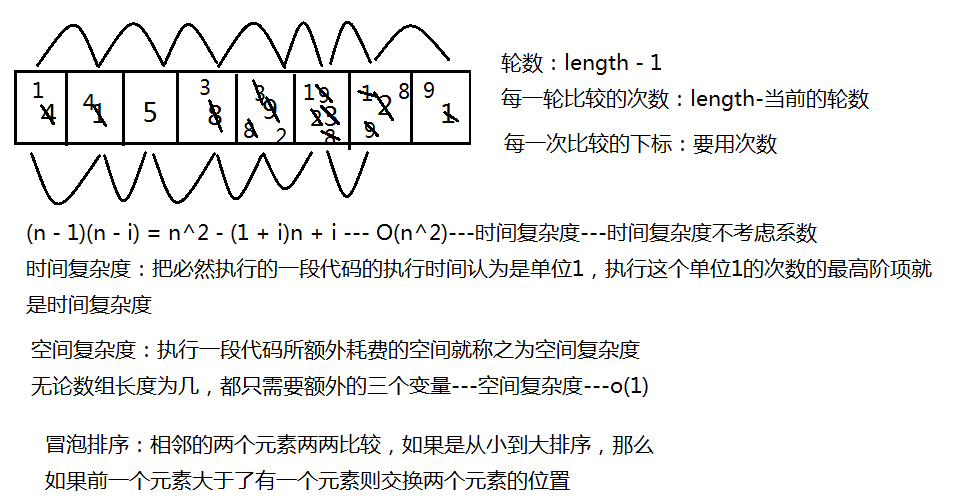
if(arr[max] < arr[i])

max = i;

}

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

1. 数组排序



// 冒泡排序

for(int i = 1; i < arr.length; i++){ // 控制轮数

for(int j = 1; j <= arr.length - i; j++){ // 控制次数和下标

if(arr[j - 1] > arr[j]){

int temp = arr[j - 1];

arr[j - 1] = arr[j];

arr[j] = temp;

}

}

}

System.out.println(Arrays.toString(arr));