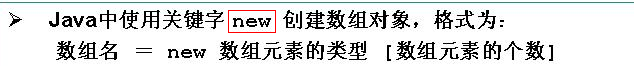
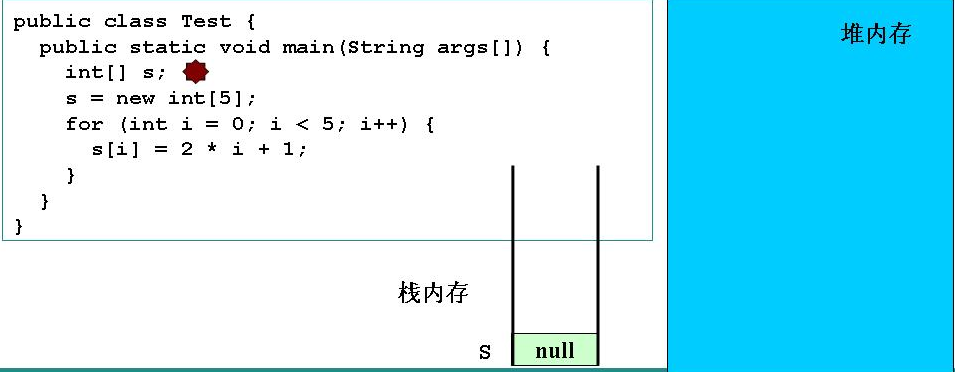
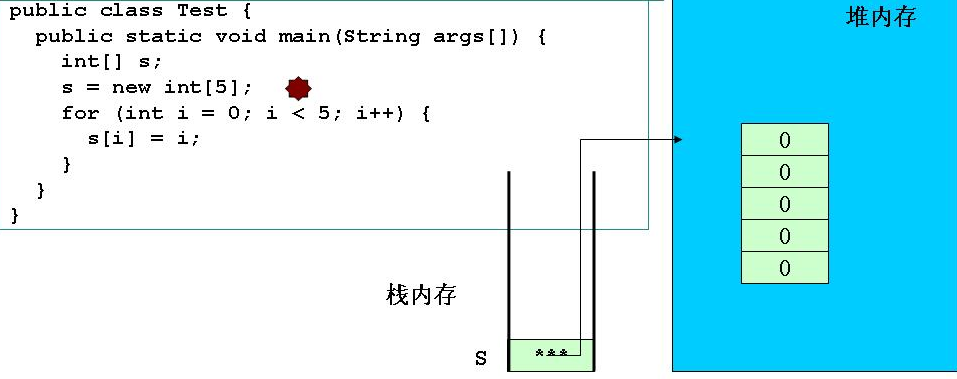
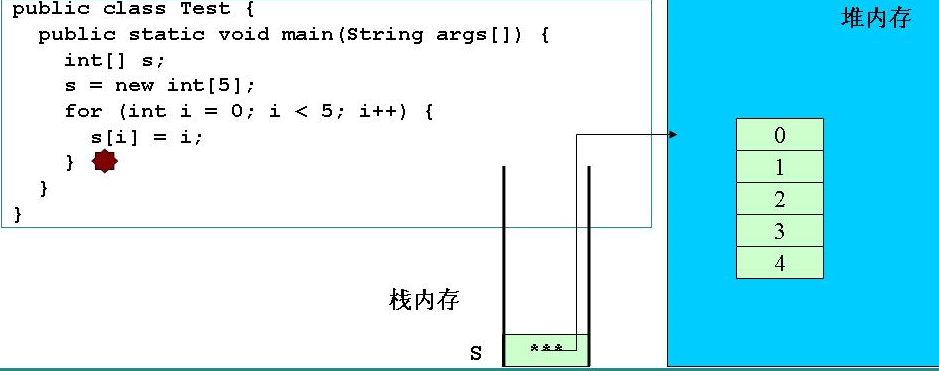
1. 数组(引用类型)：可以看成是多个相同类型的数据组合，对这些数据的统一管理。
   1. 数组变量属于引用类型(除基本类型外，全是引用类型)，数组也可以看成是对象，数组中的每个元素相当于该对象的成员变量。
   2. 数组中的元素可以是任何数据类型，包括基本类型和引用类型。
2. 一维数组的声明：int a1[]; int[] a2; double b[]; Person[] p1; String s1[];
   1. Java中声明数组时不能指定其长度(数组中元素的个数)：int a[5]//非法
3. 数组对象的创建



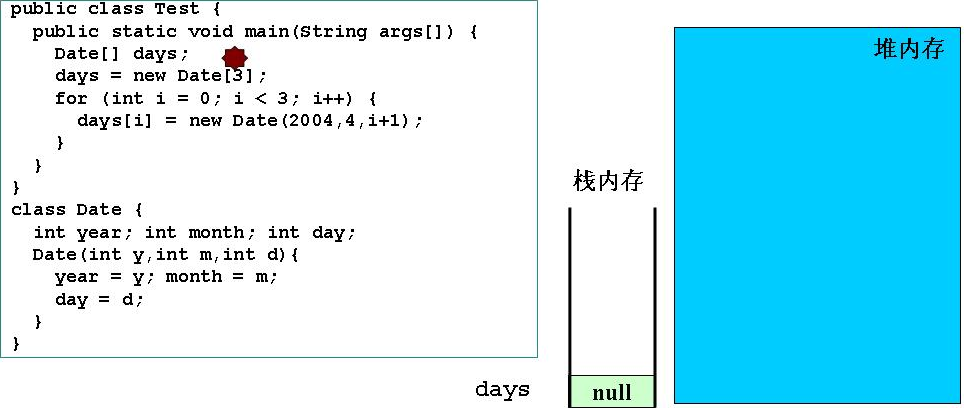
1. 例1：内存分析

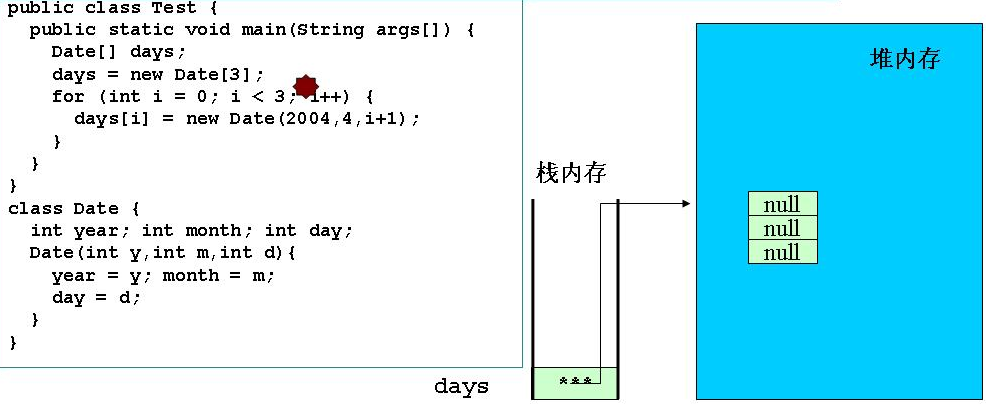




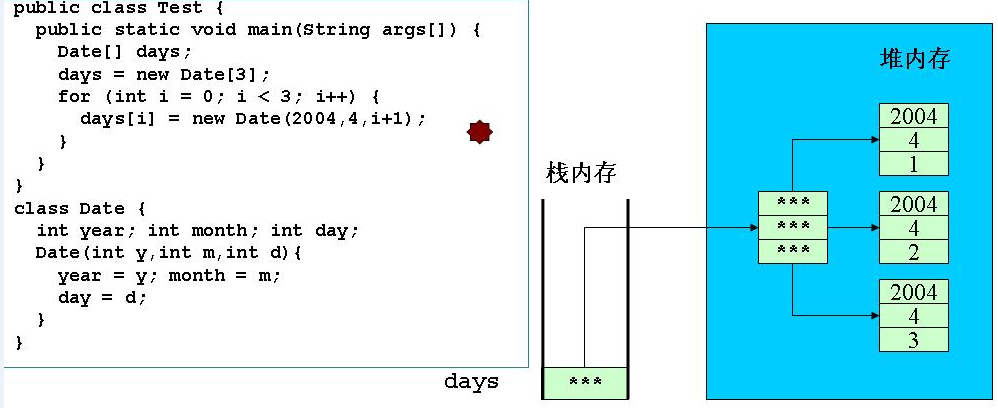


1. 例2：内存分析

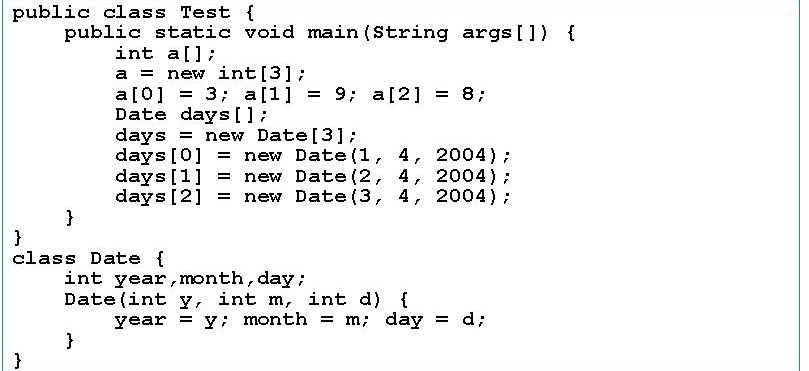




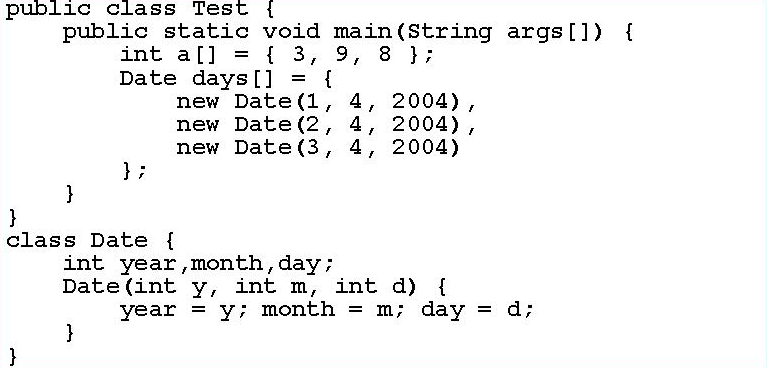
days中存放的是**引用**。



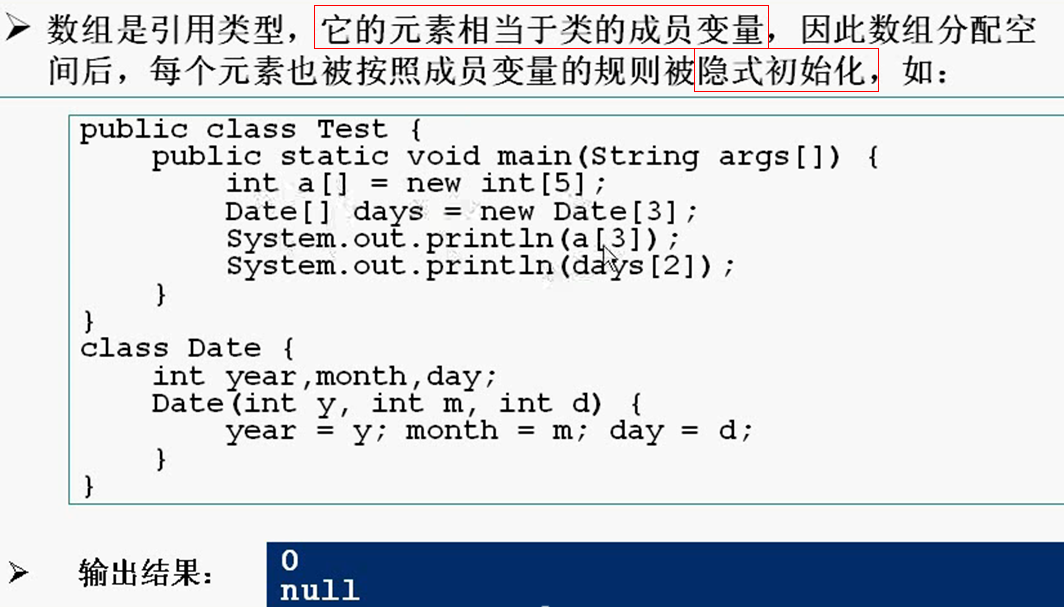
1. 数组初始化
   1. 动态初始化：数组定义与为数组元素分配空间和赋值操作分开进行(必须先分配new空间，再能往里面放东西)，例如：



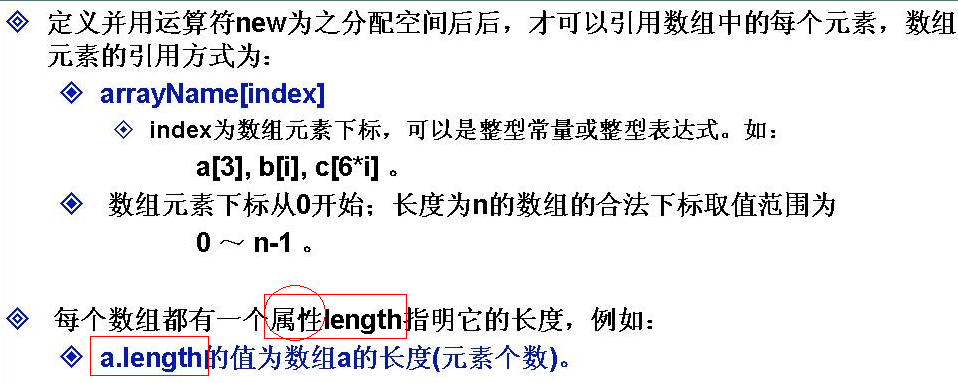
* 1. 静态初始化：在定义数组的同时就为数组元素分配空间并赋值，例如：



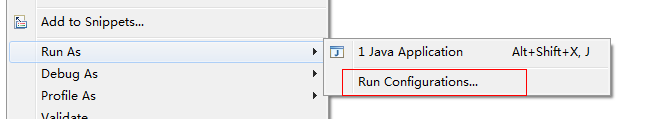
* 1. 数组元素的默认初始化：

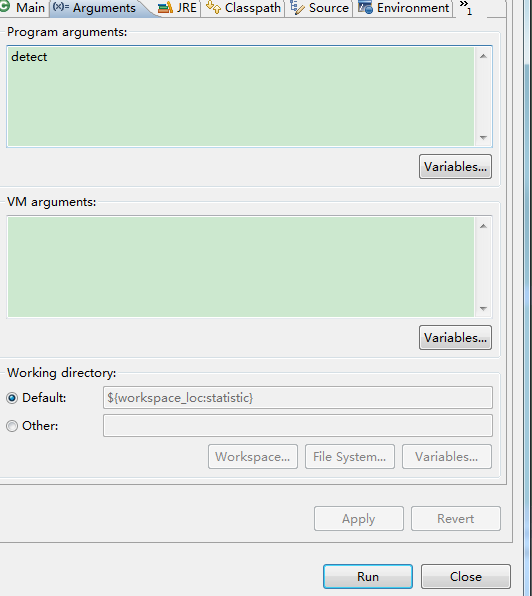


1. 数组元素的引用：

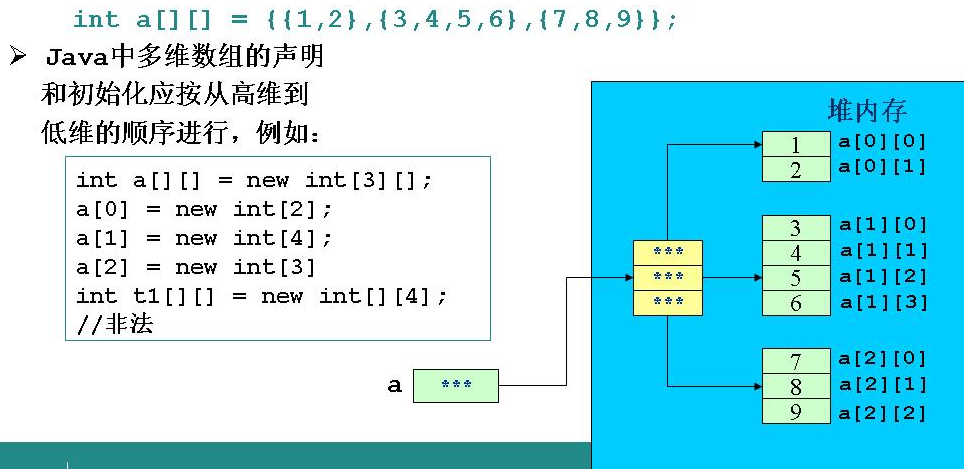


1. public static void main(String[] args) {}主函数参数中的args字符串数组，用来接收传递给该主函数的参数，采用java TestArgs arg1 arg2 argn的方式传递给它，在eclipse中运行时可以选择”右键”->Run As-> Run Configurations填写相应的参数，如下图：

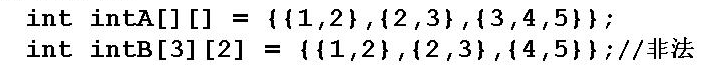




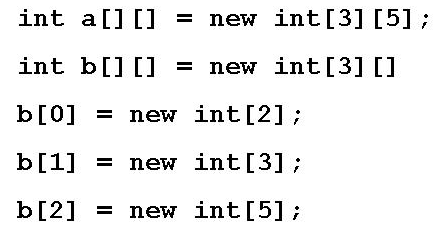
1. 将字符串转换成基础类型(采用基础类型包装类中的parseXxx方法，例如double d = Double.parseDouble("1.3"); )
2. 二维数组：二维数组可以看成是以数组为元素的数组



1. 二维数组初始化
   1. 静态初始化：(不要写数，Java会自动确定)



* 1. 动态初始化



1. 数组的拷贝：使用java.lang.System类的静态方法

public static void arraycopy(Object src, int srcPos, Object dest, int destPos, int length)