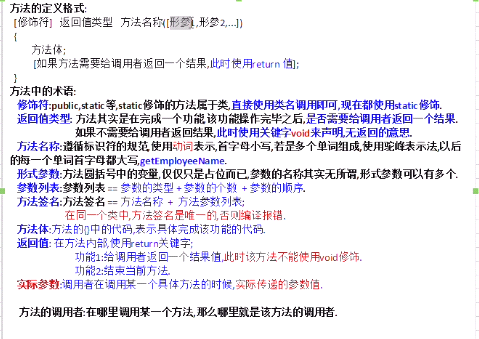
Day 04

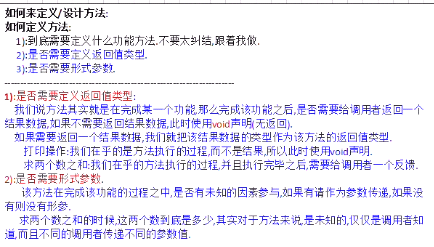
**01 引出方法**

**02方法定义和基本调用操作**

**03方法中的专业术语**

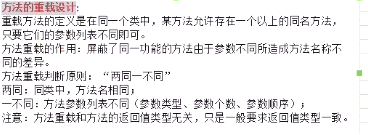


**04如何设计方法**

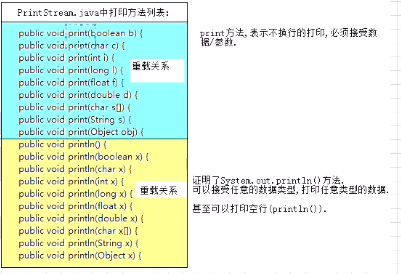


**05方法的执行流程**

**06方法重载设计**



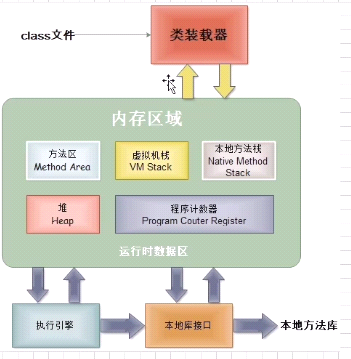
**08方法重载在jdk中的使用**



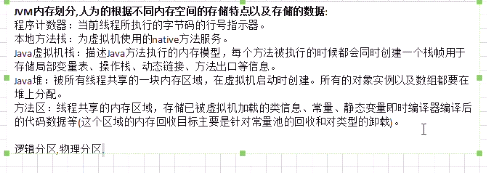
**09 方法的递归操作**

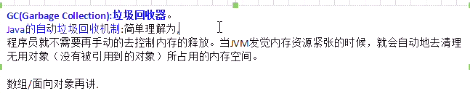
方法在执行的时候会调用自身，但是有退出条件

**10jvm内存模型**



Jvm的内存被分为三个分区：堆栈 堆内存和常量方法区

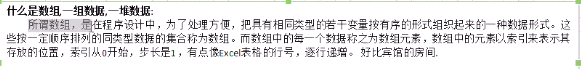




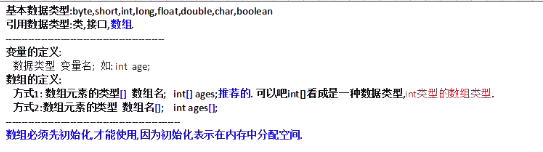
当jvm发现资源紧张时会自动进行垃圾收集或者手动强制垃圾回收机制运行

GC会不时运行释放内存空间

**11引出数组和数组定义**

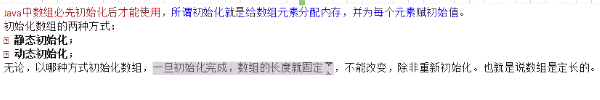


类型一致



数组在使用之前必须初始化（length表示数组的长度，可能与 数组实际存储的数据个数不符）

**12数组的静态初始化和内存分析**



数组必须先初始化才能使用，即在内存中分配空间。

数组初始化后长度就固定了

数组是定长的，元素个数是固定的。除非赋其他长度的数组。

静态初始化：

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

或者 int[] var={1,2,3}; 这种写法必须初始化

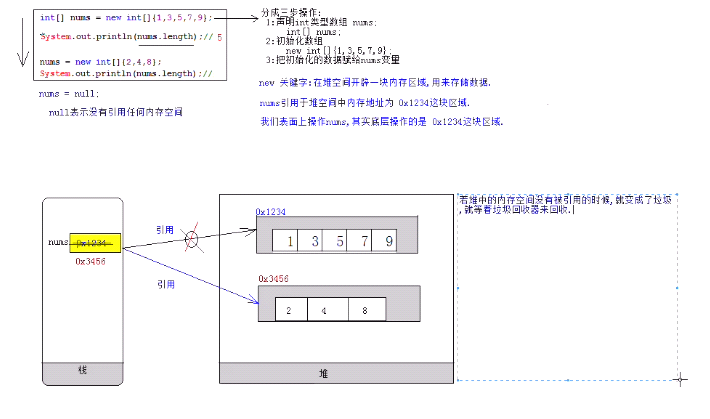
动态初始化：

Int[] var=new int[3];

Var[0]=1;

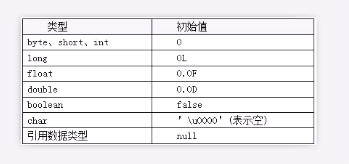
Var[1]=2;

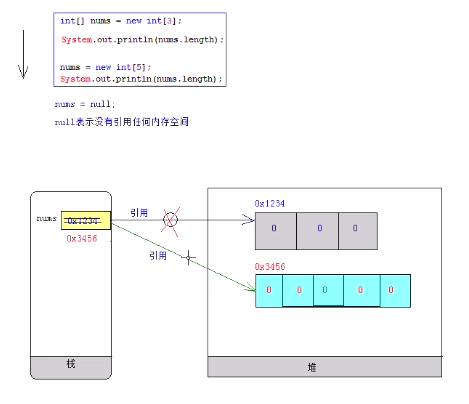
Var[2]=3;



**13 数组的动态初始化和内存分析**

不同数据类型的初始值：





**14 获取设置遍历数组的元素**

**15操作数组的常见异常**

NullPointerException：空指针异常，数组未初始化就使用

ArrayIndexOutOfBoundsException：数组索引越界异常

**16获取数组中最大和最小元素**

定义方法求max和min

**17格式化打印数组**

定义一个方法遍历数组打印元素

**18逆序排列数组元素**

**19元素出现索引**

**20小结**



栈：变量名存在栈中；基本变量的值直接存在于栈中。

每一个方法有一个栈帧

堆：使用new关键字在堆中分配空间