# 一、概述

***虚拟机栈出现的背景***

由于跨平台的特性，java的指令都是根据栈来设计的，不同平台的cpu架构不同，所以不能设计为基于寄存器的。

优点是跨平台，指令集小，编译器容易实现，缺点是性能下降，实现同样的共呢个需要更多的指令。

**栈是运行时的单位，而堆时存储的单位**

栈解决程序运行的问题，即程序如何执行，或者说如何处理数据。堆解决的时数据存储的问题，即数据怎么放，放在哪。

堆的空间 > 栈的空间

# 虚拟机栈是什么

***概念***

Java虚拟机栈(java virtual machine stack) ,早期也叫java栈，每个线程在创建的时候都会创建一个虚拟机栈，其内部保存一个个的栈帧，对应着一次次的方法调用，是线程私有的。

***生命周期***

生命周期和线程一致

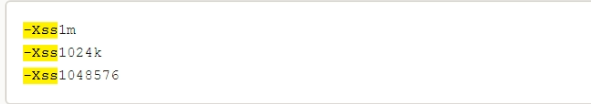
***作用***

主管java程序的运行，它保存方法的局部变量，部分结果，并参与你发给发的调用和返回

# 栈的大小设置

设置栈内存的大小

-Xss 设置栈的最大空间。栈的大小直接决定了函数调用的最大可达深度

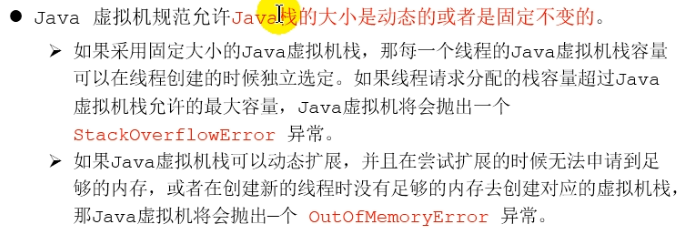




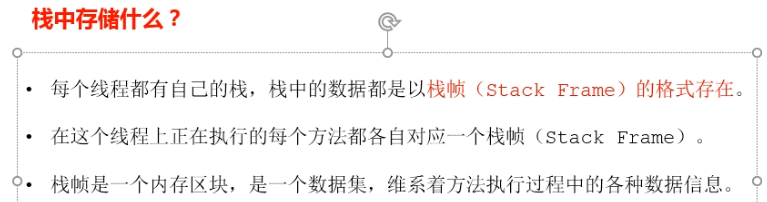
# 栈中可能出现的异常

固定时：StackOverflowError

动态时；OutOfMemoryError



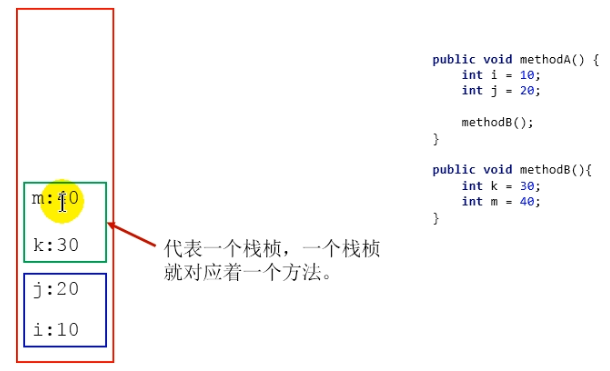
# 五、栈的运行原理

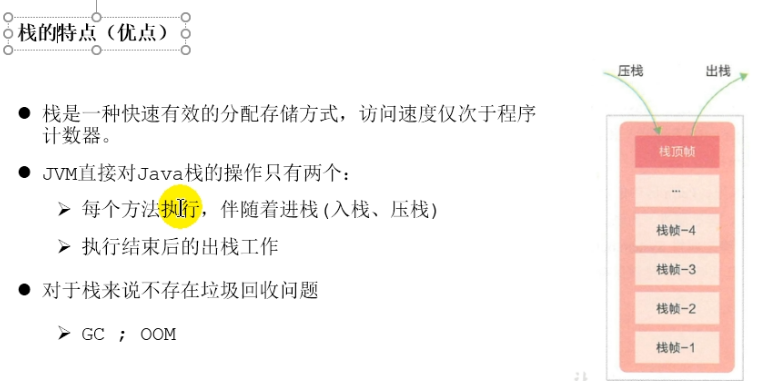


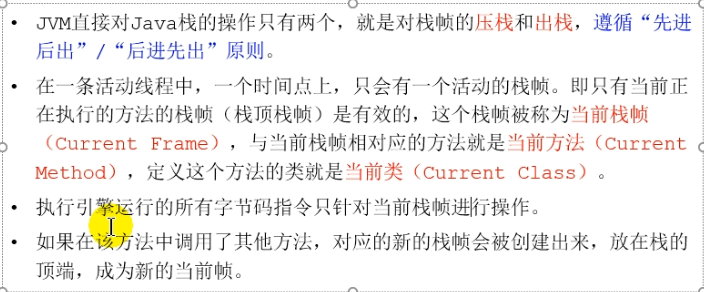
栈帧

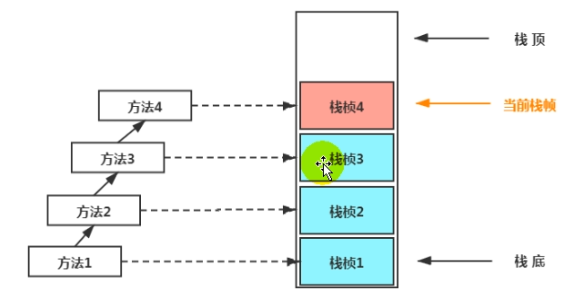
栈顶的栈帧就是当前方法，调用完就出栈了

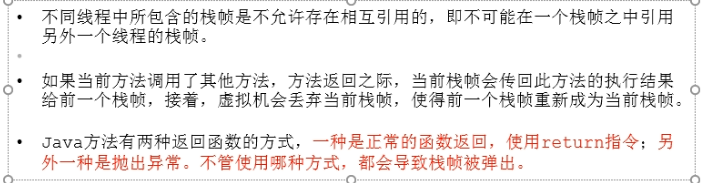
一次次的方法调用对应着一次次的栈帧的入栈和出栈





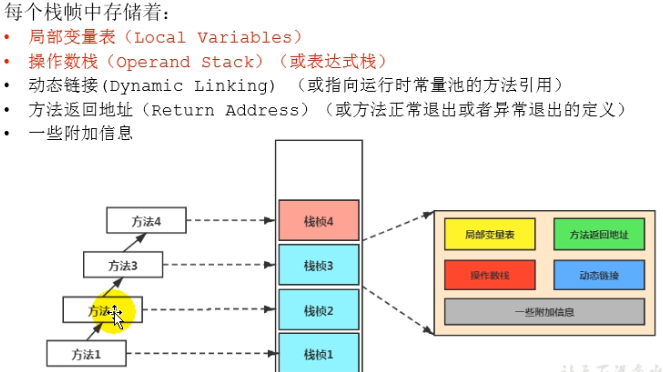




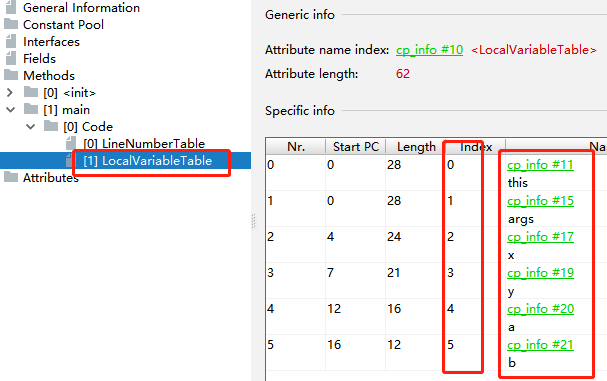


抛出未处理异常

# 四、栈帧的内部结构



## 局部变量表



局部变量表也称之为局部变量数组或本地变量表

1. 定义为一个数字数组，主要用于存储方法参数和定义在方法体内的局部变量，这些数据类型包括各类基本数据类型，对象引用，以及returnAdderss类型
2. 由于局部变量表是建立在线程的栈上，是线程的私有数据，因此不存在数据安全问题
3. **局部变量表所需的容量大小是在编译期确定下来的**，并保存在方法的Code属性的maximum local variables 数据项中，在方法运行期间是不会改变局部变量表的大小的
4. **方法嵌套调用的次数由栈的大小决定**。栈越大，方法嵌套调用次数越多，对一个函数而言，他的参数和局部变量越多，使得**局部变量表膨胀**，他的栈帧就越大，以满足方法调用所需传递的信息增大的需求，进而函数调用就会占用更多的栈空间，导致其嵌套调用次数就会减少
5. **局部变量表中的变量只在当前方法调用中有效**，在方法执行时，虚拟机通过使用局部变量表完成参数值到参数变量列表的传递过程，当方法调用结束后，随着方法栈帧的销毁，局部变量表也会随之销毁

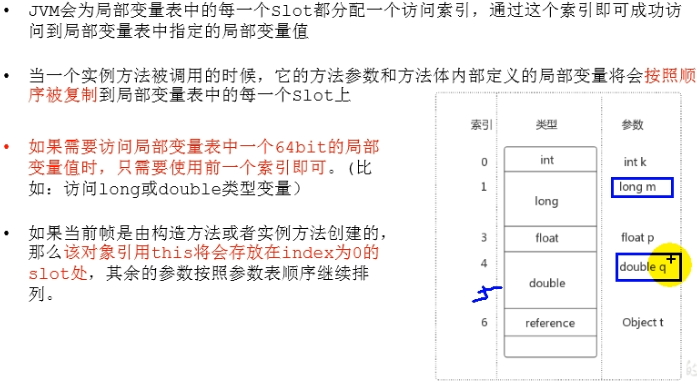
### 变量槽的理解

1、参数的存放总是在局部变量数组的index0开始，到数组长度-1的索引结束

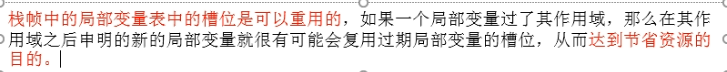
2、局部变量表，最基本的存储单元是Slot（变量槽）

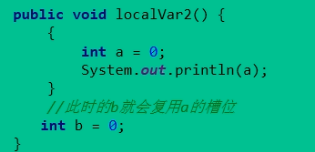
3、32位以内的类型只占一个slot（包括returnAddress类型），64位的类型（double和long）占用两个slot

Byte，short，char，boolean 在存储前都转为int （ 0F 非0T）

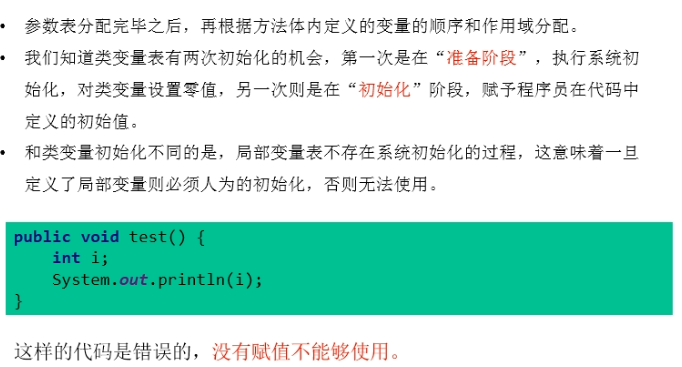


### 变量槽位的复用

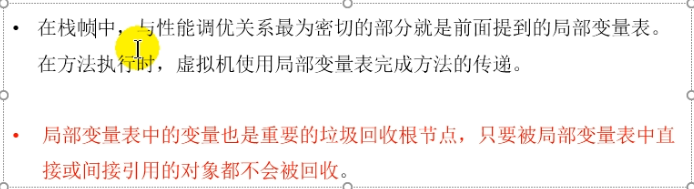




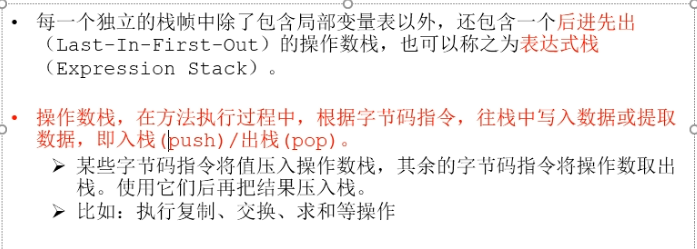
### 静态变量和局部变量的对比



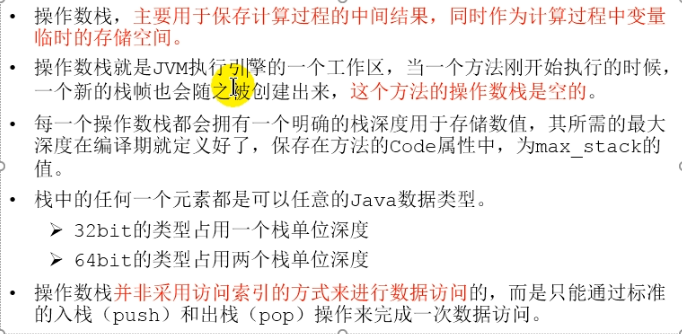
### 补充说明

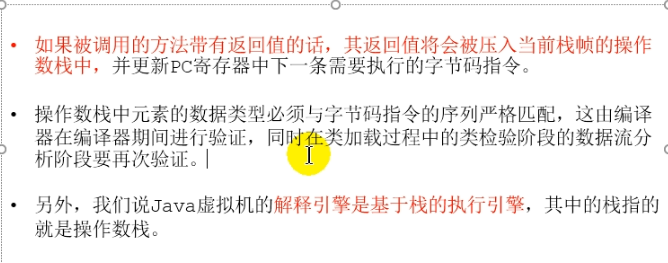


## 二、操作数栈



操作数栈主要用于保存计算过程的中间结果，同时作为计算过程中变量临时的存储空间。



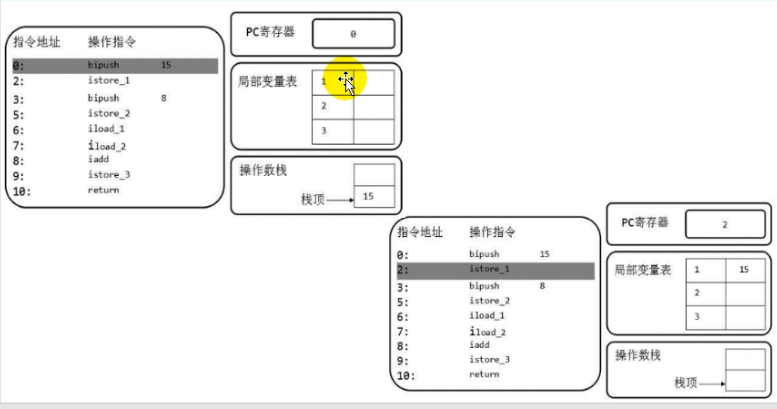


Java虚拟机的解释引擎是基于栈的执行引擎，栈指的就是操作数栈

### 代码追踪

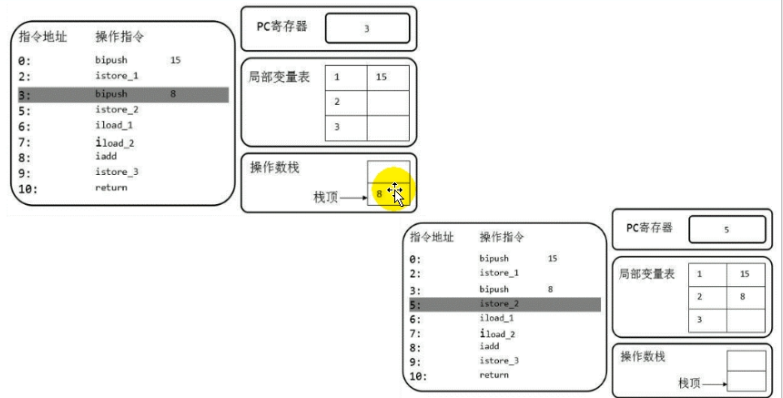
1. **bipush** 15： 把15压入操作数栈

**istore\_1** ：把操作数栈中的数据拿出放入局部变量表索引为1的地方（实例方法0处放的this）

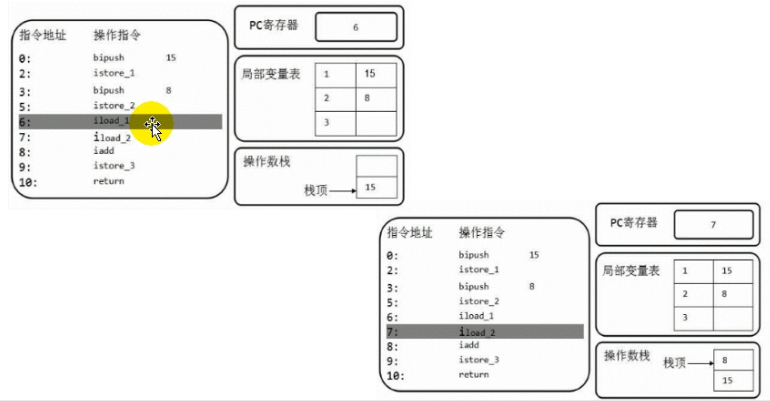


1. **bipush** 8 : 把8这个数压入操作数栈中

**istore\_2 :** 把操作数战中的数据放入局部变量表中2这个索引位

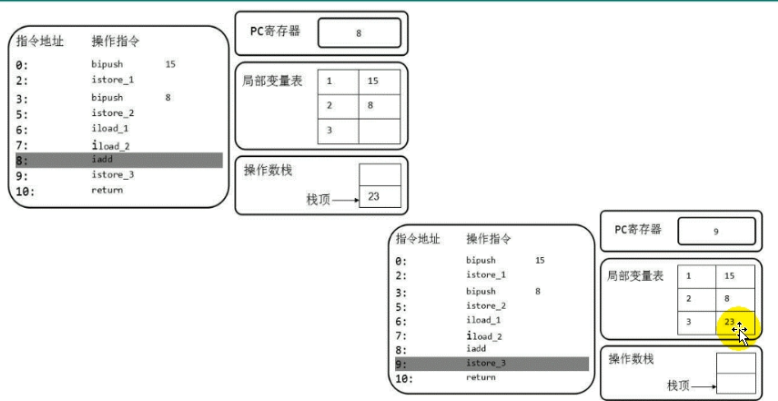


3、**iload\_1** 和**iload\_2** ：把局部变量表中的索引为1和索引为2的数据取出放入操作数栈中



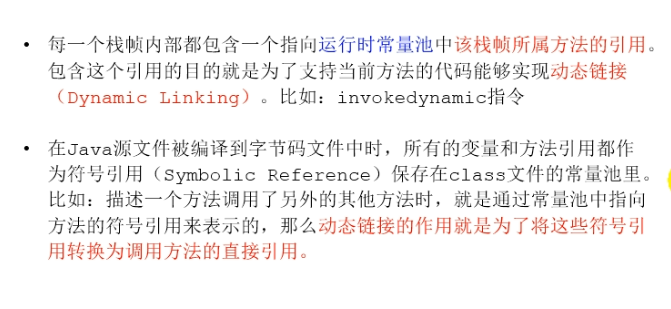
1. **iadd** : 把操作数栈中的数据相加

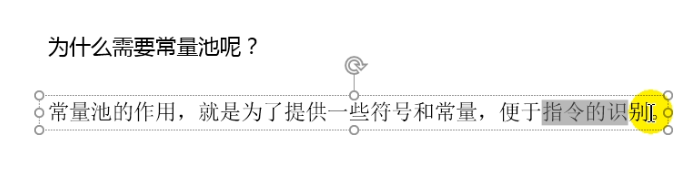
**istore\_3:** 把操作数栈中的数据放入局部变量表中索引为3的地方

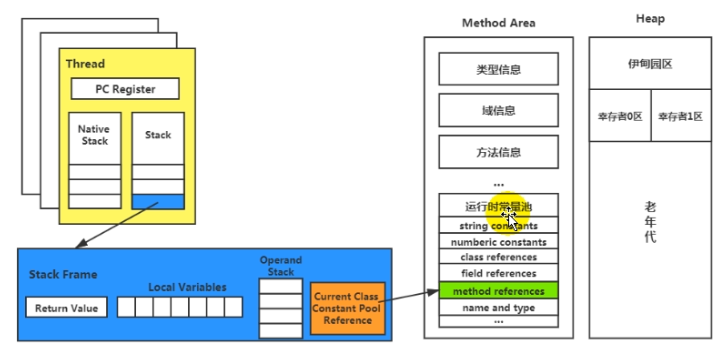


## 动态链接

指向运行时常量池的方法引用







## 方法返回地址

存放着调用该方法的 pc寄存器的值（程序计数器）

