

解析JAVA中的内存分配 PDF转换可能丢失图片或格式，建议  
阅读原文

[https://www.100test.com/kao\\_ti2020/278/2021\\_2022\\_\\_E8\\_A7\\_A3\\_E6\\_9E\\_90JAVA\\_c104\\_278559.htm](https://www.100test.com/kao_ti2020/278/2021_2022__E8_A7_A3_E6_9E_90JAVA_c104_278559.htm) 照编译原理的观点,程序运行时的内存分配有三种策略,分别是静态的,栈式的,和堆式的. 静态存储分配是指在编译时就能确定 每个数据目标在运行时刻的存储空间需求,因而在编译时就可以给他们分配固定的内存空间.这种分配策略要求程序代码中不允许有可变数据结构(比如可变数组) 的存在,也不允许有嵌套或者递归的结构出现,因为它们都会导致编译程序无法计算准确的存储空间需求. 栈式存储分配也可称为动态存储分配,是由一个类似于堆栈的运行栈来实现的.和静态存储分配相反,在栈式存储方案中,程序对数据区的需求在编译时是完全未知的,只有到运行的时候才能够知道,但是 规定在运行中进入一个程序模块时,必须知道该程序模块所需的数据区大小才能够为其分配内存.和我们在数据结构所熟知的栈一样,栈式存储分配按照先进后出的 原则进行分配。 静态存储分配要求在编译时能知道所有变量的存储要求,栈式存储分配要求在过程的入口处必须知道所有的存储要求,而堆式存储分 配则专门负责在编译时或运行时模块入口处都无法确定存储要求的数据结构的内存分配,比如可变长度串和对象实例.堆由大片的可利用块或空闲块组成,堆中的内存可以按照任意顺序分配和释放. 堆和栈的比较 上面的定义从编译原理的教材中总结而来,除静态存储分配之外,都显得很呆板和难以理解,下面撇开静态存储分配,集中比较堆和栈:从堆和栈的功能和作用来通俗的比较,堆主要用来存放对象的 , 栈主要是用来执行程序.而这种不同又主要是由于堆和栈的

特点决定的: 在编程中, 例如C/C++中, 所有的方法调用都是通过栈来进行的, 所有的局部变量, 形式参数都是从栈中分配内存空间的。实际上也不是什么分配, 只是从栈顶向上用就行, 就好像工厂中的传送带(conveyor belt)一样, Stack Pointer会自动指引你到放东西的位置, 你所要做的只是把东西放下来就行。退出函数的时候, 修改栈指针就可以把栈中的内容销毁。这样的模式速度最快, 当然要用来运行程序了。需要注意的是, 在分配的时候, 比如为一个即将要调用的程序模块分配数据区时, 应事先知道这个数据区的大小, 也就说是虽然分配是在程序运行时进行的, 但是分配的大小多少是确定的, 不变的, 而这个"大小多少"是在编译时确定的, 不是在运行时。堆是应用程序在运行的时候请求操作系统分配给自己内存, 由于从操作系统管理的内存分配, 所以在分配和销毁时都要占用时间, 因此用堆的效率非常低。但是堆的优点在于, 编译器不必知道要从堆里分配多少存储空间, 也不必知道存储的数据要在堆里停留多长的时间, 因此, 用堆保存数据时会得到更大的灵活性。事实上, 面向对象的多态性, 堆内存分配是必不可少的, 因为多态变量所需的存储空间只有在运行时创建了对象之后才能确定。在C++中, 要求创建一个对象时, 只需用new命令编制相关的代码即可。执行这些代码时, 会在堆里自动进行数据的保存。当然, 为达到这种灵活性, 必然会付出一定的代价: 在堆里分配存储空间时会花掉更长的时间! 这也正是导致我们刚才所说的效率低的原因, 看来列宁同志说的好, 人的优点往往也是人的缺点, 人的缺点往往也是人的优点。100Test 下载频道开通, 各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)