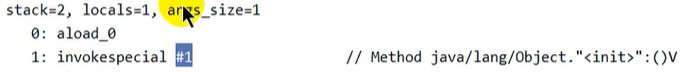
**（帧数据区：方法返回地址、动态链接、一些附加信息；帧数据区中保存着能够访问常量池的一些指针。）**

***动态链接或者叫做指向运行时常量池的方法引用：（Dynamic Linking）*：（就是该方法在常量池中对应的符号引用，唯一标识，到时候执行该方法时才知道具体该执行哪一块的代码）**

1、每一个栈帧内部都包含一个指向运行时常量池中该栈帧所属方法的引用包含这个引用的目的就是为了当前方法的代码能够实现动态链接（Dynamic Linking）。比如：invokedynamic指令。

2、在Java源文件被编译到字节码文件中时，所有的的变量和方法引用都作为符号引用（Symbolic Reference）保存在class文件的常量池里。比如：描述一个方法调用了另外的其他方法时，就是通过常量池中指向方法的符号引用来表示的，那么动态链接的作用就是为了将这些符号引用转换位调用方法的直接引用。





（注意哈：这里指向的还是一个常量池的符号引用，不是实体，通过这个引用才去找实体，这也体现出动态链接对多态的作用。）

([java 为什么需要常量池 - CodesGeek - 博客园 (cnblogs.com)](https://www.cnblogs.com/daydaynobug/articles/6600631.html))

### (常量池探秘

每个 java 文件编译为 class 文件后，都将产生当前类独有的常量池，我们称之为静态常量池。class 文件中的常量池包含两部分：**字面值**（literal）和**符号引用**（Symbolic Reference）。其中字面值可以理解为 java 中定义的字符串常量、final 常量等；符号引用指的是一些字符串，这些字符串表示当前类引用的外部类、方法、变量等的引用地址的抽象表示形式，在类被jvm装载并第一次使用这些符号引用时，这些符号引用将会解析为直接引用。符号常量包含：

* 类和接口的全限定名
* 字段的名称和描述符
* 方法的名称和描述符

jvm在进行类装载时，将class文件中常量池部分的常量加载到方法区中，此时方法区中的保存常量的逻辑区域称之为运行时常量区。

)

([aload、aload\_1、iload都是什么意思\_好一个迷途小书童的博客-CSDN博客](https://blog.csdn.net/qq_27416233/article/details/90018541))

**为什么需要运行时常量池？**