字节码：<https://www.jianshu.com/p/252f381a6bc4>

Class文件结构：



Magic number: 从上面的总览图中我们知道前4个字节表示的是魔数，对应我们Demo的是 0XCAFE BABE。什么是魔数？魔数是用来区分文件类型的一种标志，一般都是用文件的前几个字节来表示。比如0XCAFE BABE表示的是class文件，那么有人会问，文件类型可以通过文件名后缀来判断啊？是的，但是文件名是可以修改的（包括后缀），那么为了保证文件的安全性，讲文件类型写在文件内部来保证不被篡改。  
从java的字节码文件类型我们看到，CAFE BABE翻译过来是咖啡宝贝之意，然后再看看java图标。

版本号：我们识别了文件类型之后，接下来要知道版本号。版本号含主版本号和次版本号，都是各占2个字节。在此Demo种为0X0000 0033。其中前面的0000是次版本号，后面的0033是主版本号。通过进制转换得到的是次版本号为0，主版本号为51。  
从oracle官方网站我们能够知道，51对应的正式jdk1.7，而其次版本为0，所以该文件的版本为1.7.0。如果需要验证，可以在用java --version命令输出版本号，或者修改编译目标版本--target重新编译，查看编译后的字节码文件版本号是否做了相应的修改。

字节码中的栈帧，操作数栈，局部变量表：<https://blog.csdn.net/a616413086/article/details/51272309>

栈帧（stack frame）是用于支持虚拟机进行方法调用和方法执行的数据结构，它是虚拟机运行时数据区中的虚拟机栈的栈元素。栈帧存储了方法的局部变量表、操作数栈、动态连接和方法返回地址等信息。  
     每一个方法从调用开始到执行完成的过程，就对应着一个栈帧在虚拟机栈里面从入栈到出栈的过程。  
     对于执行引擎来说，活动线程中，只有栈顶的栈帧是有效的，称为当前栈帧，这个栈帧所关联的方法称为当前方法。执行引擎所运行的所有字节码指令都只针对当前栈帧进行操作。

（1）局部变量表  
      局部变量表是一组变量值存储空间，用于存放方法参数和方法内部定义的局部变量。在Java程序被编译成Class文件时，就在方法的Code属性的max\_locals数据项中确定了该方法所需要分配的  
      最大局部变量表的容量。  
      局部变量表的容量以变量槽（Slot）为最小单位，32位虚拟机中一个Slot可以存放一个32位以内的数据类型（boolean、byte、char、short、int、float、reference和returnAddress八种）。  
      reference类型虚拟机规范没有明确说明它的长度，但一般来说，虚拟机实现至少都应当能从此引用中直接或者间接地查找到对象在Java堆中的起始地址索引和方法区中的对象类型数据。  
      returnAddress类型是为字节码指令jsr、jsr\_w和ret服务的，它指向了一条字节码指令的地址。  
      虚拟机是使用局部变量表完成参数值到参数变量列表的传递过程的，如果是实例方法（非static），那么局部变量表的第0位索引的Slot默认是用于传递方法所属对象实例的引用，在方法中通过this访问。  
      Slot是可以重用的，当Slot中的变量超出了作用域，那么下一次分配Slot的时候，将会覆盖原来的数据。Slot对对象的引用会影响GC（要是被引用，将不会被回收）。  
  系统不会为局部变量赋予初始值（实例变量和类变量都会被赋予初始值）。也就是说不存在类变量那样的准备阶段。

（2）操作数栈  
      Java虚拟机的解释执行引擎被称为"基于栈的执行引擎"，其中所指的栈就是指－操作数栈。  
      操作数栈也常被称为操作栈。  
     和局部变量区一样，操作数栈也是被组织成一个以字长为单位的数组。但是和前者不同的是，它不是通过索引来访问，而是通过标准的栈操作—压栈和出栈—来访问的。比如，如果某个指令把一个值压入到操作数栈中，稍后另一个指令就可以弹出这个值来使用。  
      虚拟机在操作数栈中存储数据的方式和在局部变量区中是一样的：如int、long、float、double、reference和returnType的存储。对于byte、short以及char类型的值在压入到操作数栈之前，也会被转换为int。  
      虚拟机把操作数栈作为它的工作区——大多数指令都要从这里弹出数据，执行运算，然后把结果压回操作数栈。比如，iadd指令就要从操作数栈中弹出两个整数，执行加法运算，其结果又压回到操作数栈中，看看下面的示例，它演示了虚拟机是如何把两个int类型的局部变量相加，再把结果保存到第三个局部变量的：

Iload：将局部变量表的元素压入操作数栈

Istore:将操作栈中的元素加入局部变量表