

Class文件结构

并且在1997年发布的第一版Java虚拟机规范中就曾经承诺过：“In the future, we will consider bounded extensions to the Java virtual machine to provide better support for other languages”（在未来，我们会对Java虚拟机进行适当的扩展，以便更好地支持其他语言运行于JVM之上）。

时至今日,这个目标已经实现。涌现Clojure、Groovy、JRuby、Jython、Scala一批运行在java虚拟机上的语言。

实现语言无关性的基础仍然是虚拟机和字节码存储格式。Java虚拟机不和包括Java在内的任何语言绑定，它只与“Class文件”这种特定的二进制文件格式所关联，Class文件中包含了Java虚拟机指令集和符号表以及若干其他辅助信息。基于安全方面的考虑，Java虚拟机规范要求Class文件中使用许多强制性的语法和结构化约束，但任一门功能性语言都可以表示为一个能被Java虚拟机所接受的有效的Class文件。作为一个通用、机器无关的执行平台，任何其他语言的实现者都可以将Java虚拟机作为语言的产品交付媒介。

Java,Scale等各种语言中的各种变量、关键字和运算符的语义最终都是由多条字节码命令组合而成的，因此字节码命令所能提供的语义描述能力肯定会比单一语言本身更加强大。

Class文件结构

任何一个Class文件都对应着唯一——一个类或接口的定义信息，但反过来说，类或接口并不一定都得定义在Class文件里（譬如类或接口也可以通过类加载器直接生成）。

Class文件是一组以8位字节为基础单位的、顺序严格的二进制流，中间没有添加任何分隔符或空隙，当遇到需要占用8位字节以上空间的数据项时，则会按照高位在前（Big-Endian）的方式分割成若干个8位字节进行存储。

Class文件格式采用一种类似于C语言结构体的伪结构来存储数据，这种伪结构中只有两种数据类型：无符号数和表，后面的解析都要以这两种数据类型为基础。

无符号数属于基本的数据类型，以u1、u2、u4、u8来分别代表1个字节、2个字节、4个字节和8个字节的无符号数，无符号数可以用来描述数字、索引引用、数量值或者按照UTF-8编码构成字符串值。

表是由多个无符号数或者其他表作为数据项构成的复合数据类型，所有表都习惯性地以 “_info” 结尾。表用于描述有层次关系的复合结构的数据，整个Class文件本质上就是一张表。

Class的结构不像XML等描述语言，由于它没有任何分隔符号，所以在表6-1中的数据项，无论是顺序还是数量，甚至于数据存储的字节序这样的细节，都是被严格限定的，哪个字节代表什么含义，长度是多少，先后顺序如何，都不允许改变。

魔数 (Magic Number) CAFEBAFE --->>> 次版本号 (Minor Version) 和 主版本号 --->>> 常量池入口(Class文件的资源仓库)