常量池中的14种常量项的结构总表

|  |  |  |  |
| --- | --- | --- | --- |
| 常量 | 项目 | 类型 | 描述 |
| CONSTANT\_Utf8\_info | tag | 1bit | 值为1 |
| length | 2bit | UTF-8编码的字符串占用的字符数 |
| bytes | 1bit | 长度为length的UTF-8编码的字符串 |
| CONSTANT\_Integer\_info | tag | 1bit | 值为3 |
| bytes | 4bit | 按照高位在前存储的int值 |
| CONSTANT\_Float\_info | tag | 1bit | 值为4 |
| bytes | 4bit | 按照高位在前存储的float值 |
| CONSTANT\_Long\_info | tag | 1bit | 值为5 |
| bytes | 8bit | 按照高位在前存储long值 |
| CONSTANT\_Double\_info | tag | 1bit | 值为6 |
| bytes | 8bit | 按照高位在前存储double值 |
| CONSTANT\_Class\_info | tag | 1bit | 值为7 |
| index | 2bit | 指向全限定名常量项的索引 |
| CONSTANT\_String\_info | tag | 1bit | 值为8 |
| index | 2bit | 指向字符串字面量的索引 |
| CONSTANT\_Fieldref\_info | tag | 1bit | 值为9 |
| index | 2bit | 指向声明字段的类或者接口描述符CONSTANT\_Class\_info的索引项 |
| index | 2bit | 指向字段描述符CONSTANT\_NameAndType的索引项 |
| CONSTANT\_Methodref\_info | tag | 1bit | 值为10 |
| index | 2bit | 指向声明方法的类描述符CONSTANT\_Class\_info的索引项 |
| index | 2bit | 指向名称及类型描述符CONSTANT\_NameAndType的索引项 |
| CONSTANT\_InterfaceMethodref\_info | tag | 1bit | 值为11 |
| index | 2bit | 指向声明方法的接口描述符CONSTANT\_Class\_info的索引项 |
| index | 2bit | 指向名称及类型描述符CONSTANT\_NameAndType的索引项 |
| CONSTANT\_NameAndType\_info | tag | 1bit | 值为12 |
| index | 2bit | 指向该字段或方法名称常量项的索引 |
| index | 2bit | 指向该字段或方法描述符常量项的索引 |
| CONSTANT\_MethodHandle\_info | tag | 1bit | 值为15 |
| reference\_kind | 1bit | 值必须在1~9之间，它决定了方法句柄的类型方法句柄类型的值表示方法句柄的字节码行为 |
| reference\_index | 2bit | 值必须是对常量池的有效索引 |
| CONSTANT\_MethodType\_info | tag | 1bit | 值为16 |
| descriptor\_index | 2bit | 值必须是对常量池的有效索引，常量池在该索引处的项必须是CONSTANT\_Utf8\_info结构，表示方法的描述符 |
| CONSTANT\_InvokeDynamic\_info | tag | 1bit | 值为18 |
| bootstarp\_method\_attr\_index | 2bit | 值必须是对当前Class文件中引导方法表的bootstrap\_methods[]数组的有效索引 |
| name\_and\_type\_index | 2bit | 值必须是对当前常量池的有效索引，常量池在该索引处的项必须是CONSTANT\_NameAndType\_info结构，表示方法名和方法描述符 |

字段表结构

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 数量 |
| 2bit | access\_flags | 1 |
| 2bit | name\_index | 1 |
| 2bit | descriptor\_index | 1 |
| 2bit | attributes\_count | 1 |
| 2bit | attributes | attributes\_count |

**字段访问标志**

|  |  |  |
| --- | --- | --- |
| 标志名称 | 标志值（换成二进制看） | 含义 |
| ACC\_PUBLIC | 0x0001 | 字段是否public |
| ACC\_PRIVATE | 0x0002 | 字段是否private |
| ACC\_PROTECTED | 0x0004 | 字段是否protected |
| ACC\_STATIC | 0x0008 | 字段是否static |
| ACC\_FINAL | 0x0010 | 字段是否final |
| ACC\_VOLATILE | 0x0040 | 字段是否volatile |
| ACC\_TRANSIENT | 0x0080 | 字段是否transient |
| ACC\_SYNTHETIC | 0x1000 | 字段是否由编译器自动产生 |
| ACC\_ENUM | 0x4000 | 字段是否enum |

**方法访问标志**

|  |  |  |
| --- | --- | --- |
| 标志名称 | 标志值（换成二进制看） | 含义 |
| ACC\_PUBLIC | 0x0001 | 方法是否为public |
| ACC\_PRIVATE | 0x0002 | 方法是否为private |
| ACC\_PROTECTED | 0x0004 | 方法是否为protected |
| ACC\_STATIC | 0x0008 | 方法是否为static |
| ACC\_FINAL | 0x0010 | 方法是否为final |
| ACC­­­\_SYNCHRONIZED | 0x0020 | 方法是否为synchronized |
| ACC\_BRIDGE | 0x0040 | 方法是否是由编译器产生的桥接方法 |
| ACC\_VARARGS | 0x0080 | 方法是否接受不定参数 |
| ACC\_NATIVE | 0x0100 | 方法是否为native |
| ACC\_ABSTRACT | 0x0400 | 方法是否为abstract |
| ACC\_STRICTFP | 0x0800 | 方法是否为strictfp |
| ACC\_SYNTHETIC | 0x1000 | 方法是否是由编译器自动产生 |

**虚拟机规范预定义的属性**

|  |  |  |
| --- | --- | --- |
| 属性名称 | 使用位置 | 含义 |
| Code | 方法表 | Java代码编译成的字节码指令 |
| ConstantValue | 字段表 | final关键字定义的常量值 |
| Deprecations | 类、方法表、字段表 | 被声明为deprecated的方法和字段 |
| Exceptions | 方法表 | 方法抛出的异常 |
| EnclosingMethod | 类文件 | 仅当一个类为局部类或者匿名类时才能拥有这个属性，这个属性标识这个类所在的外围方法 |
| InnerClasses | 类文件 | 内部类列表 |
| LineNumberTable | Code属性 | Java源码的行号与字节码指令的对应关系 |
| LocalVariableTable | Code属性 | 方法的局部变量描述 |
| StackMapTable | Code属性 | JDK1.6中新增的属性，供新的类型检查验证器（Type Checker）检查和处理目标方法的局部变量和操作数栈所需要的类型是否匹配 |
| Signature | 类、方法表、字段表 | JDK1.5中新增的属性，这个属性用于支持泛型情况下的方法签名，在java语言中，任何表、接口、初始化方法或成员的泛型签名如果包含了类型变量（Type Variables）或参数化类型（Parameterized Types），则Signature属性会为它记录泛型签名信息，由于java的泛型采用擦除法实现，在为了避免类型信息被擦除后导致签名混乱，需要这个属性记录泛型中的相关信息 |
| SourceFile | 类文件 | 记录源文件名称 |
| SourceDebugExtension | 类文件 | JDK1.6中新增的属性，SourceDebugExtension属性用于存储额外的调试信息。譬如在进行JSP文件调试时，无法通过java堆栈来定位到JSP文件的行号，JSR-45规范为这些非java语言编写，却需要编译成字节码并进行在java虚拟机中的程序提供了一个进行调试的标准机制，使用SourceDebugExtension属性就可以用于存储这个标准所新加入的调试信息 |
| Synthetic | 类、方法表、字段表 | 标识方法或字段为编译器自动生成的 |
| LocalVariableTypeTable | 类 | JDK1.5中新增的属性，它使用特征签名代替描述符，是为了引入泛型方法之后能描述泛型参数化类型而添加 |
| RuntimeVisibleAnnotation | 类、方法表、字段表 | JDK1.5中新增的属性，为动态注解提供支持。RuntimeVisibleAnnotations属性用于指明哪些注解是运行时可见的 |
| RuntimeInvisibleAnnotations | 类、方法表、字段表 | JDK1.5中新增的属性，与RuntimeVisibleAnnotations属性作用刚好相反，用于指明哪些注解是运行时不可见的 |
| RuntimeVisibleParameterAnnotations | 方法表 | JDK1.5中新增的属性，作用与RuntimeVisibleAnnotations属性类似，只不过作用对象为方法参数 |
| RuntimeInvisibleParameterAnnotations | 方法表 | JDK1.5中新增的属性，作用与RuntimeInvisibleAnnotations属性类似，只不过作用对象为方法参数 |
| AnnotationDefault | 方法表 | JDK1.5中新增的属性，用于记录注解类元素的默认值 |
| BootstrapMethods | 类文件 | JDK1.7中新增的属性，用于保存invokedynamic指令引用的引导方法限定符 |