



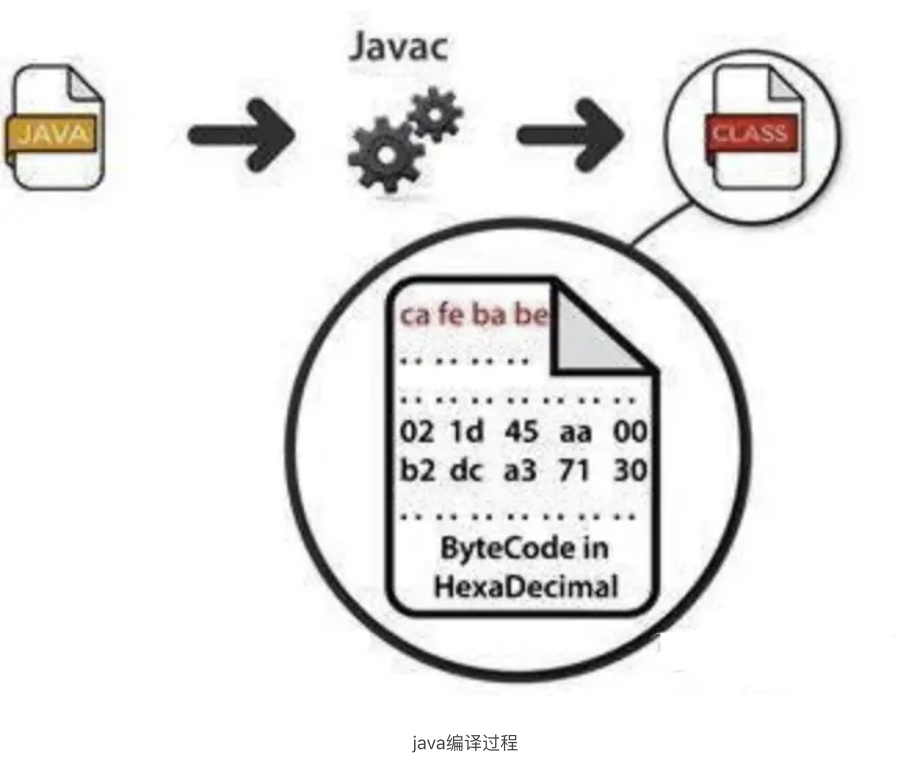
03 - ASM和ClassFile

 舍是境界 200 关注

 0.3 2022.01.10 07:08:58 字数 811 阅读 23

ClassFile

在.class文件中，存储的是ByteCode数据。但是，这些ByteCode数据并不是杂乱无章的，而是遵循一定的数据结构。



这个 .class 文件遵循的数据结构就是由 [Java Virtual Machine Specification](#)中定义的 [The class File Format](#)，如下所示。

```
1 ClassFile {
2     u4          magic;
3     u2          minor_version;
4     u2          major_version;
5     u2          constant_pool_count;
6     cp_info     constant_pool[constant_pool_count-1];
7     u2          access_flags;
8     u2          this_class;
9     u2          super_class;
10    u2          interfaces_count;
11    u2          interfaces[interfaces_count];
12    u2          fields_count;
13    field_info   fields[fields_count];
14    u2          methods_count;
15    method_info  methods[methods_count];
16    u2          attributes_count;
17    attribute_info attributes[attributes_count];
18 }
```



 舍是境界 200 关注

总资产 147

- 36 - ASM之方法计时
阅读 12
- 42 - ASM之Class Transformation总结
阅读 13
- 41 - ASM之优化、删除等复杂的变换
阅读 17

热门故事

妻子去世半年，我再娶一个小十岁的女人有错吗？

代替公主和亲后，我成了敌国后宫“升职”最快的妃子

直播间打赏五十万，女主播主动私信我要见面

生完二胎，老公给我雇了一个“90后”小保姆

- 推荐阅读**
- 去步行街——2021.12.20
阅读 997
 - 事情一定会与愿望违背
阅读 1,122
 - 你总会成为某个人的例外
阅读 455
 - 删掉你微信的人
阅读 450
 - 有人在偷偷温暖你
阅读 363

字节码类库

ASM是操作字节码的类库，但并不是唯一的，还有许多其它的操作字节码的类库。

常见的字节码类库

在下面列举了几个比较常见的字节码类库：

- [Apache Commons BCEL](#)：其中BCEL为Byte Code Engineering Library首字母的缩写。
- [Javassist](#)：Javassist表示Java programming assistant
- [ObjectWeb ASM](#)：本文介绍的类库。
- [Byte Buddy](#)：在ASM基础上实现的一个类库。

字节码的类库和ClassFile之间的关系

对于上图，我们用三句来描述它们的关系：

- 中间层－多个.class文件，虽然每个类里面的内容各不相同，但它们里面的内容都称为字节码（ByteCode）。
- 中下层－不论这些.class文件内容有怎样大的差异，它们都共同遵守同一个数据结构，即ClassFile。
- 中上层－为了方便于人们对于字节码（ByteCode）内容的操作，逐渐衍生出了各种操作字节码的类库。
- 上下层－不考虑中间层，我们可以说，不同的字节码类库是在同一个ClassFile结构上发展起来的。

既然有多个可以选择的字节码类库，那么我们为什么要选择ASM呢？

ASM的特点

问题：与其它的操作Java字节码的类库相比，ASM有哪些与众不同的地方呢？

回答：在实现相同的功能前提下，使用ASM，运行速度更快（运行时间短，属于“时间维度”），占用的内存空间更小（内存空间，属于“空间维度”）。

The ASM was designed to be as fast and as small as possible.

Being as fast as possible is important in order not to slow down too much the applications that use ASM at runtime, for dynamic class generation or transformation.

And being as small as possible is important in order to be used in memory



constrained environments, and to avoid bloating the size of small applications or libraries using ASM.

ASM与ClassFile的关系

ASM通过对ClassFile的操作，可以提供对代码的相关analysis, generation and transformation功能

学习ASM有三个不同的层次：

- 第一个层次，ASM的应用层面。也就是说，我们可以使用ASM来做什么呢？对于一个.class文件来说，我们可以使用ASM进行analysis、generation和transformation操作。
- 第二个层次，ASM的源码层面。也就是，ASM的代码组织形式，它为分Core API和Tree API的内容。
- 第三个层次，Java ClassFile层面。从JVM规范的角度，来理解.class文件的结构，来理解ASM中方法和参数的含义。

小结

本文介绍了ASM和classFile的关系和字节码结构，以及常见字节码类库，希望对你能有所帮助。



1人点赞 >





Java系列



更多精彩内容，就在简书APP




"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下



舍是境界  专注高性能，高可用，高扩展架构领域
总资产147 共写了77.1W字 获得407个赞 共16个粉丝

关注

推荐阅读

 更多精彩内容 >