

螺杆干式真空泵

h5页面制作软件

linux云主机

公司LOGO设计

贝克真空泵

柳州螺蛳粉配方

06 - ASM之ClassVisitor



舍是境界

200

关注

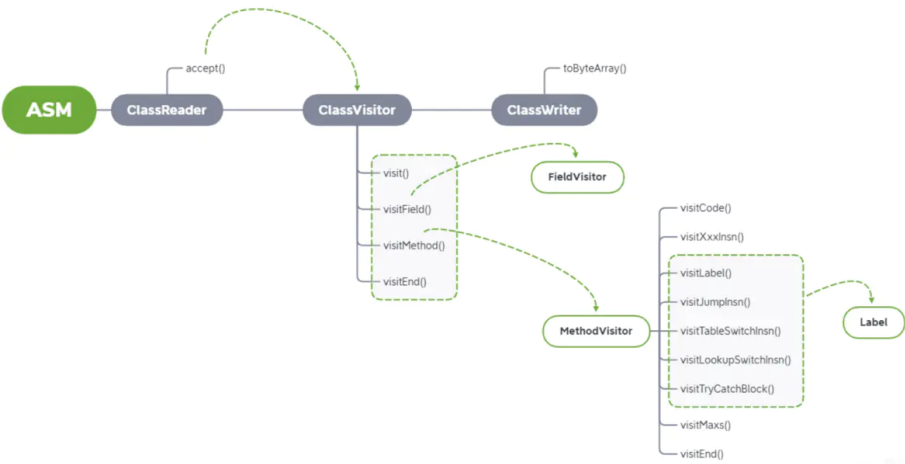
0.34

2022.01.15 07:37:17

字数 1,229

阅读 30

在ASM Core API中，最重要的三个类就是ClassReader、ClassVisitor和ClassWriter类。在进行Class Generation操作的时候，ClassVisitor和ClassWriter这两个类起着重要作用，而并不需要ClassReader类的参与。在本文当中，我们将对ClassVisitor类进行介绍。



ClassReader、ClassVistor、ClassWriter

ClassVisitor类

class info

ClassVisitor是一个抽象类。由于ClassVisitor类是一个abstract类，所以不能直接使用new关键字创建ClassVisitor对象。

```
1 public abstract class ClassVisitor {
2 }
```

同时，由于ClassVisitor类是一个abstract类，要想使用它，就必须有具体的子类来继承它。比较常见的ClassVisitor子类有ClassWriter类（Core API）和ClassNode类（Tree API）。

```
1 public class ClassWriter extends ClassVisitor {
2 }
3
4 public class ClassNode extends ClassVisitor {
5 }
```

fields

ClassVisitor类定义的字段有哪些。

```
1 public abstract class ClassVisitor {
2     protected final int api;
3     protected ClassVisitor cv;
4 }
```

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

java继承的调用顺序(java中子类继承父类程序执行顺序)

阅读 323

Swift-Alamofire二次封装

阅读 121

TS-类和接口

阅读 181

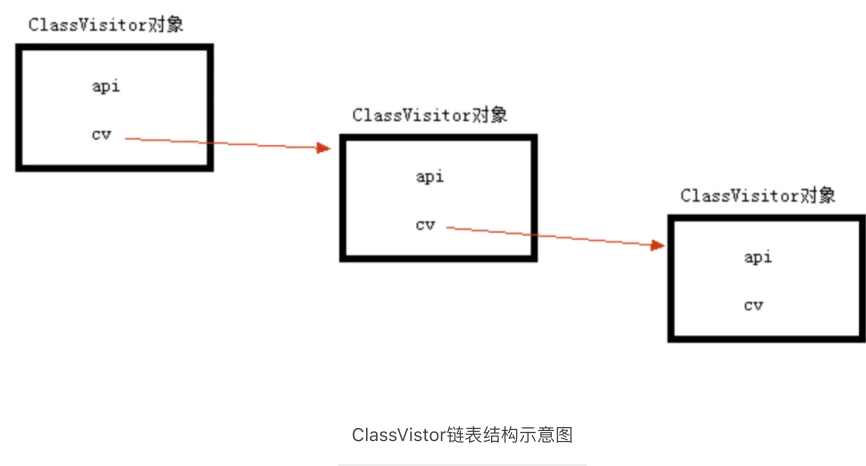
C++<第三十二篇>：转换构造函数和类型转换函数

阅读 63

Java 基础 12. Java 注解和反射

阅读 274

- api字段：它是一个int类型的数据，指出了当前使用的ASM API版本，其取值有Opcodes.ASM4、Opcodes.ASM5、Opcodes.ASM6、Opcodes.ASM7、Opcodes.ASM8和Opcodes.ASM9。我们使用的ASM版本是9.0，因此我们在给api字段赋值的时候，选择Opcodes.ASM9就可以了。
- cv字段：它是一个ClassVisitor类型的数据，它的作用是将多个ClassVisitor串连起来。



constructors

ClassVisitor类定义的构造方法有哪些。

```
1 public abstract class ClassVisitor {
2     public ClassVisitor(final int api) {
3         this(api, null);
4     }
5
6     public ClassVisitor(final int api, final ClassVisitor classVisitor) {
7         this.api = api;
8         this.cv = classVisitor;
9     }
10 }
```

methods

ClassVisitor类定义的方法有哪些。在ASM当中，使用到了Visitor Pattern（访问者模式），所以ClassVisitor当中许多的visitXxx()方法。

虽然，在ClassVisitor类当中，有许多visitXxx()方法，但是，我们只需要关注这4个方法：visit()、visitField()、visitMethod()和visitEnd()。因为这4个方法是ClassVisitor类的精髓或骨架，认识了这4个方法，其它的visitXxx()都容易扩展；

```
1 public abstract class ClassVisitor {
2     public void visit(
3         final int version,
4         final int access,
5         final String name,
6         final String signature,
7         final String superName,
8         final String[] interfaces);
9     public FieldVisitor visitField( // 访问字段
10         final int access,
11         final String name,
12         final String descriptor,
13         final String signature,
14         final Object value);
15     public MethodVisitor visitMethod( // 访问方法
16         final int access,
17         final String name,
18         final String descriptor,
19         final String signature,
20         final String[] exceptions);
21     public void visitEnd();
22 }
```

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

java继承的调用顺序(java中子类继承父类程序执行顺序)

阅读 323

Swift-Alamofire二次封装

阅读 121

TS-类和接口

阅读 181

C++<第三十二篇>：转换构造函数和类型转换函数

阅读 63

Java 基础 12. Java 注解和反射

阅读 274

```
22 |      // .....
23 |  }
```

在ClassVisitor的visit()方法、visitField()方法和visitMethod()方法中都带有signature参数。这个signature参数“泛型”密切相关；换句话说，如果处理的是一个带有泛型信息的类、字段或方法，那么就需要给signature参数提供一定的值；如果处理的类、字段或方法不带有“泛型”信息，那么将signature参数设置为null就可以了。

如果大家对signature参数感兴趣，我们可以使用之前介绍的ASMPrint类去打印一下某个泛型类的ASM代码。例如，java.lang.Comparable是一个泛型接口，我们就可以使用ASMPrint类来打印一下它的ASM代码，从来查看signature参数的值是什么。

方法的调用顺序

在ClassVisitor类当中，定义了多个visitXxx()方法。这些visitXxx()方法，遵循一定的调用顺序。这个调用顺序，是参考自ClassVisitor类的API文档。

```
1 | visit
2 | [visitSource][visitModule][visitNestHost][visitPermittedSubclass][visitOuterClass]
3 | (
4 |     visitAnnotation I
5 |     visitTypeAnnotation I
6 |     visitAttribute
7 | )*
8 | (
9 |     visitNestMember I
10 |    visitInnerClass I
11 |    visitRecordComponent I
12 |    visitField I
13 |    visitMethod
14 | )*
15 | visitEnd
```

其中，涉及到一些符号，它们的含义如下：

- ◻：表示最多调用一次，可以不调用，但最多调用一次。
- ◯和！：表示在多个方法之间，可以选择任意一个，并且多个方法之间不分前后顺序。
- *：表示方法可以调用0次或多次。

根据我们关注的四个visit方法，模型我们可以简化如下：

```
1 | visit
2 | (
3 |     visitField I
4 |     visitMethod
5 | )*
6 | visitEnd
```

也就是说，先调用visit()方法，接着调用visitField()方法或visitMethod()方法，最后调用visitEnd()方法。

visitXxx()方法与ClassFile

ClassVisitor的visitXxx()方法与ClassFile之间存在对应关系。在ClassVisitor中定义的visitXxx()方法，并不是凭空产生的，这些方法存在的目的是为了生成一个合法的.class文件，而这个.class文件要符合ClassFile的结构，所以这些visitXxx()方法与ClassFile的结构密切相关。

visit()方法

```
1 | public void visit(
2 |     final int version,
3 |     final int access,
```

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

java继承的调用顺序(java中子类继承父类程序执行顺序)

阅读 323

Swift-Alamofire二次封装

阅读 121

TS-类和接口

阅读 181

C++<第三十二篇>：转换构造函数和类型转换函数

阅读 63

Java 基础 12. Java 注解和反射

阅读 274



```
4 | final String name,
5 | final String signature,
6 | final String superName,
7 | final String[] interfaces);

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 | }
```

ClassVisitor方法	参数	ClassFile
ClassVisitor.visit()	version	minor_version 和 major_version
	access	access_flags
	name	this_class
	signature	attributes 的一部分信息
	superName	super_class
	interfaces	interfaces_count 和 interfaces
ClassVisitor.visitField()		field_info
ClassVisitor.visitMethod()		method_info

ClassVistor与class信息对应关系图

visitField()方法

```
1 | public FieldVisitor visitField( // 访问字段
2 |     final int access,
3 |     final String name,
4 |     final String descriptor,
5 |     final String signature,
6 |     final Object value);

1 | field_info {
2 |     u2          access_flags;
3 |     u2          name_index;
4 |     u2          descriptor_index;
5 |     u2          attributes_count;
6 |     attribute_info attributes[attributes_count];
7 | }
```

ClassVisitor方法	参数	field_info
ClassVisitor.visitField()	access	access_flags
	name	name_index
	descriptor	descriptor_index
	signature	attributes 的一部分信息
	value	attributes 的一部分信息

visitFiled与class对应图

visitMethod()方法

```
1 | public MethodVisitor visitMethod( // 访问方法
2 |     final int access,
3 |     final String name,
```

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

java继承的调用顺序(java中子类继承父类程序执行顺序)

阅读 323

Swift-Alamofire二次封装

阅读 121

TS-类和接口

阅读 181

C++<第三十二篇>：转换构造函数和类型转换函数

阅读 63

Java 基础 12. Java 注解和反射

阅读 274

```
3 | final String descriptor,  
4 | final String signature,  
5 | final String[] exceptions);  
6 |  
  
1 | method_info {  
2 |     u2          access_flags;  
3 |     u2          name_index;  
4 |     u2          descriptor_index;  
5 |     u2          attributes_count;  
6 |     attribute_info attributes[attributes_count];  
7 | }
```

ClassVisitor方法	参数	method_info
ClassVisitor.visitMethod()	access	access_flags
	name	name_index
	descriptor	descriptor_index
	signature	attributes 的一部分信息
	exceptions	attributes 的一部分信息

visitMethod与class对应关系图

visitEnd()方法

visitEnd()方法，它是这些visitXxx()方法当中最后一个调用的方法。

为什么visitEnd()方法是“最后一个调用的方法”呢？是因为在ClassVisitor当中，定义了多个visitXxx()方法，这些个visitXxx()方法之间要遵循一个先后调用的顺序，而visitEnd()方法是最后才去调用的。等到visitEnd()方法调用之后，就表示说再也不去调用其它的visitXxx()方法了，所有的“工作”已经做完了，到了要结束的时候了。

```
1 | /*  
2 |  * Visits the end of the class.  
3 |  * This method, which is the last one to be called,  
4 |  * is used to inform the visitor that all the fields and methods of the class have been  
5 |  */  
6 | public void visitEnd() {  
7 |     if (cv != null) {  
8 |         cv.visitEnd();  
9 |     }  
10 | }
```


小结


本文对ClassVisitor进行了相关介绍，讲了ClassVistor的类结构，以及方法和class的对应关系，希望对你能有帮助，总结如下：

- 介绍了ClassVisitor类的不同部分。我们去了解这个类不同的部分，是为了能够熟悉ClassVisitor这个类。
- 在ClassVisitor类当中，定义了许多visitXxx()方法，这些方法的调用要遵循一定的顺序。
- 在ClassVisitor类当中，定义的visitXxx()方法中的参数与ClassFile结构密切相关。

 1人点赞 >



 Java系列



更多精彩内容，就在简书APP



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

java继承的调用顺序(java中子类继承父类程序执行顺序)

阅读 323

Swift-Alamofire二次封装

阅读 121

TS-类和接口

阅读 181

C++<第三十二篇>：转换构造函数和类型转换函数

阅读 63

Java 基础 12. Java 注解和反射

阅读 274