



11 - ASM之FieldVisitor

舍是境界

200

关注

0.282

2022.01.21 07:11:26

字数 577

阅读 32

通过调用ClassVisitor类的visitField()方法，会返回一个FieldVisitor类型的对象。在本文当中，我们就对FieldVisitor类进行介绍。

```
1 | public FieldVisitor visitField(int access, String name, String descriptor, String sign
```

FieldVisitor类

在学习FieldVisitor类的时候，可以与ClassVisitor类进行对比，这两个类在结构上有很大的相似性：两者都是抽象类，都定义了两个字段，都定义了两个构造方法，都定义了visitXxx()方法。

class info

FieldVisitor类是一个abstract类。

```
1 | public abstract class FieldVisitor {
2 | }
```

fields

FieldVisitor字段如下：

```
1 | public abstract class FieldVisitor {
2 |     protected final int api;
3 |     protected FieldVisitor fv;
4 | }
```

constructors

FieldVisitor类定义的两构造方法。

```
1 | public abstract class FieldVisitor {
2 |     public FieldVisitor(final int api) {
3 |         this(api, null);
4 |     }
5 |
6 |     public FieldVisitor(final int api, final FieldVisitor fieldVisitor) {
7 |         this.api = api;
8 |         this.fv = fieldVisitor;
9 |     }
10 | }
```

methods

FieldVisitor类定义的方法有哪些。在FieldVisitor类当中，一共定义了4个visitXxx()方法，但是，我们只需要关注其中的visitEnd()方法就可以了，对于一些非必要的方法，我们就暂时忽略它；将visitXxx()方法精简到一个最小的认知集合，那么就只剩下visitEnd()方法了。



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

TS-类和接口

阅读 181

JavaScript进阶：组合式继承和寄生组合式继承

阅读 164

C++<第十五篇>：构造函数和析构函数

阅读 163

JavaScript之apply()与call()细谈

阅读 150



```
1 public abstract class FieldVisitor {
2     // .....
3
4     public void visitEnd() {
5         if (fv != null) {
6             fv.visitEnd();
7         }
8     }
9 }
```

在FieldVisitor类内定义多个visitXxx()方法，也需要遵循一定的调用顺序，如下所示：

```
1 (
2     visitAnnotation |
3     visitTypeAnnotation |
4     visitAttribute
5 )*
6 visitEnd
```

由于我们只关注visitEnd()方法，那么，这个调用顺序就变成如下这样：

```
1 | visitEnd
```

FieldVisitor类示例

字段常量

预期目标:

```
1 public interface HelloWorld {
2     int intValue = 100;
3     String strValue = "ABC";
4 }
```

编码实现：

```
1 import lsieun.utils.FileUtils;
2 import org.objectweb.asm.ClassWriter;
3 import org.objectweb.asm.FieldVisitor;
4
5 import static org.objectweb.asm.Opcodes.*;
6
7 public class HelloWorldGenerateCore {
8     public static void main(String[] args) throws Exception {
9         String relative_path = "sample/HelloWorld.class";
10        String filepath = FileUtils.getFilePath(relative_path);
11
12        // (1) 生成byte[]内容
13        byte[] bytes = dump();
14
15        // (2) 保存byte[]到文件
16        FileUtils.writeBytes(filepath, bytes);
17    }
18
19    public static byte[] dump() throws Exception {
20        // (1) 创建ClassWriter对象
21        ClassWriter cw = new ClassWriter(ClassWriter.COMPUTE_FRAMES);
22
23        // (2) 调用visitXxx()方法
24        cw.visit(V1_8, ACC_PUBLIC + ACC_ABSTRACT + ACC_INTERFACE, "sample/HelloWorld",
25
26            {
27                FieldVisitor fv1 = cw.visitField(ACC_PUBLIC | ACC_FINAL | ACC_STATIC, "int'
28                fv1.visitEnd();
29            }
30
31            {
32                FieldVisitor fv2 = cw.visitField(ACC_PUBLIC + ACC_FINAL + ACC_STATIC, "str'
33                fv2.visitEnd();
34            }
35    }
```

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

TS-类和接口

阅读 181

JavaScript进阶：组合式继承和寄生组合式继承

阅读 164

C++<第十五篇>：构造函数和析构函数

阅读 163

JavaScript之apply()与call()细谈

阅读 150



```
36         cw.visitEnd();
37
38         // (3) 调用toByteArray()方法
39         return cw.toByteArray();
40     }
41 }
```

验证结果:

```
1 import java.lang.reflect.Field;
2
3 public class HelloWorldRun {
4     public static void main(String[] args) throws Exception {
5         Class<?> clazz = Class.forName("sample.HelloWorld");
6         Field[] declaredFields = clazz.getDeclaredFields();
7         if (declaredFields.length > 0) {
8             System.out.println("fields:");
9             for (Field f : declaredFields) {
10                 Object value = f.get(null);
11                 System.out.println("    " + f.getName() + ": " + value);
12             }
13         }
14     }
15 }
```

输出结果:

```
1 fields:
2     intValue: 100
3     strValue: ABC
```

注意点: 在得到一个FieldVisitor对象之后, 要记得调用它的visitEnd()方法。

visitAnnotation

无论是ClassVisitor类, 还是FieldVisitor类, 又或者是MethodVisitor类, 总会有一些visitXxx()方法是在课程当中不会涉及到的。但是, 在日后的工作和学习当中, 很可能, 在某一天你突然就对一个visitXxx()方法产生了兴趣, 那该如何学习这个visitXxx()方法呢? 我们可以借助于ASMPrint类。

预期目标: 假如我们想生成如下HelloWorld类:

```
1 public interface HelloWorld {
2     @MyTag(name = "tomcat", age = 10)
3     int intValue = 100;
4 }
```



MyTag定义如下

```
1 public @interface MyTag {
2     String name();
3     int age();
4 }
```

编码实现:

```
1 import lsieun.utils.FileUtils;
2 import org.objectweb.asm.AnnotationVisitor;
3 import org.objectweb.asm.ClassWriter;
4 import org.objectweb.asm.FieldVisitor;
5
6 import static org.objectweb.asm.Opcodes.*;
7
8 public class HelloWorldGenerateCore {
9     public static void main(String[] args) throws Exception {
10         String relative_path = "sample/HelloWorld.class";
11         String filepath = FileUtils.getFilePath(relative_path);
12     }
```



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

TS-类和接口

阅读 181

JavaScript进阶: 组合式继承和寄生组合式继承

阅读 164

C++<第十五篇>: 构造函数和析构函数

阅读 163

JavaScript之apply()与call()细谈

阅读 150

```
13 // (1) 生成byte[]内容
14 byte[] bytes = dump();
15
16 // (2) 保存byte[]到文件
17 FileUtils.writeBytes(filepath, bytes);
18 }
19
20 public static byte[] dump() throws Exception {
21     // (1) 创建ClassWriter对象
22     ClassWriter cw = new ClassWriter(ClassWriter.COMPUTE_FRAMES);
23
24     // (2) 调用visitXxx()方法
25     cw.visit(V1_8, ACC_PUBLIC | ACC_ABSTRACT | ACC_INTERFACE, "sample/HelloWorld",
26
27         {
28             FieldVisitor fv1 = cw.visitField(ACC_PUBLIC | ACC_FINAL | ACC_STATIC, "int'
29
30             {
31                 AnnotationVisitor anno = fv1.visitAnnotation("Lsample/MyTag;", false);
32                 anno.visit("name", "tomcat");
33                 anno.visit("age", 10);
34                 anno.visitEnd();
35             }
36
37             fv1.visitEnd();
38         }
39
40     cw.visitEnd();
41
42     // (3) 调用toByteArray()方法
43     return cw.toByteArray();
44 }
45 }
```

小结

- FieldVisitor类，从结构上来说，与ClassVisitor很相似；对于FieldVisitor类的各个不同部分进行介绍，以便从整体上来理解FieldVisitor类。
- 对于FieldVisitor类定义的方法，我们只需要关心FieldVisitor.visitEnd()方法就可以了。
- 我们可以借助于ASMPrint类来帮助我们学习新的visitXxx()方法。

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 200 专注高性能，高可用，高扩展架构领域
总资产 147 共写了77.2W字 获得408个赞 共16个粉丝

关注

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

TS-类和接口

阅读 181

JavaScript进阶：组合式继承和寄生组合式继承

阅读 164

C++<第十五篇>：构造函数和析构函数

阅读 163

JavaScript之apply()与call()细谈

阅读 150