



约的软件

linux云主机

信息安全

公司LOGO设计

螺杆干式真空泵

h5页面制作软件

19 - MethodVisitor之调用方法



舍是境界

200

关注

0.258 2022.01.29 08:38:58 字数 118 阅读 17

目标

```
1 public class HelloWorld {
2     public void test(int a, int b) {
3         int val = Math.max(a, b); // 对static方法进行调用
4         System.out.println(val); // 对non-static方法进行调用
5     }
6 }
```

编码实现

```
1 import lsieun.utils.FileUtils;
2 import org.objectweb.asm.ClassWriter;
3 import org.objectweb.asm.MethodVisitor;
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_SUPER, "sample/HelloWorld", null, "java/lang/Object", null);
25
26        {
27            MethodVisitor mv1 = cw.visitMethod(ACC_PUBLIC, "<init>", "()V", null, null);
28            mv1.visitCode();
29            mv1.visitVarInsn(ALOAD, 0);
30            mv1.visitMethodInsn(INVOKESTATIC, "java/lang/Object", "<init>", "()V", false);
31            mv1.visitInsn(RETURN);
32            mv1.visitMaxs(1, 1);
33            mv1.visitEnd();
34        }
35
36        {
37            MethodVisitor mv2 = cw.visitMethod(ACC_PUBLIC, "test", "(II)V", null, null);
38            mv2.visitCode();
39            mv2.visitVarInsn(ILOAD, 1);
40            mv2.visitVarInsn(ILOAD, 2);
41            mv2.visitMethodInsn(INVOKESTATIC, "java/lang/Math", "max", "(II)I", false);
42            mv2.visitVarInsn(ISTORE, 3);
43            mv2.visitFieldInsn(GETSTATIC, "java/lang/System", "out", "Ljava/io/PrintStream");
44            mv2.visitVarInsn(ILOAD, 3);
45            mv2.visitMethodInsn(INVOKEVIRTUAL, "java/io/PrintStream", "println", "(I)V");
46            mv2.visitInsn(RETURN);
47            mv2.visitMaxs(2, 4);
48            mv2.visitEnd();
49        }
50    }
51 }
```



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

新生儿一口气睡了五个小时
阅读 553

初一的饺子
阅读 3,298

鹧鸪天·冷月无声花落闲
阅读 882

小说完结赚1K多，新书情况以及对未来一段时间的计划
阅读 4,007

熬夜要命的呀
阅读 2,431



```
51 |         cw.visitEnd();
52 |
53 |         // (3) 调用toByteArray()方法
54 |         return cw.toByteArray();
55 |     }
56 | }
```

验证结果

```
1 | import java.lang.reflect.Method;
2 |
3 | public class HelloWorldRun {
4 |     public static void main(String[] args) throws Exception {
5 |         Class<?> clazz = Class.forName("sample.HelloWorld");
6 |         Object obj = clazz.newInstance();
7 |
8 |         Method m = clazz.getDeclaredMethod("test", int.class, int.class);
9 |         m.invoke(obj, 10, 20);
10 |    }
11 | }
```

Frame的变化

对于HelloWorld类中test()方法对应的Instruction内容如下：

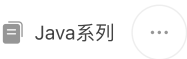
```
1 | public void test(int, int);
2 |     Code:
3 |         0: iload_1
4 |         1: iload_2
5 |         2: invokestatic #21          // Method java/lang/Math.max:(II)I
6 |         5: istore_3
7 |         6: getstatic   #27          // Field java/lang/System.out:Ljava/io/Print
8 |         9: iload_3
9 |        10: invokevirtual #33        // Method java/io/PrintStream.println:(I)V
10 |        13: return
```

该方法对应的Frame变化情况如下：

```
1 | test(II)V
2 | [sample/HelloWorld, int, int] []
3 | [sample/HelloWorld, int, int] [int]
4 | [sample/HelloWorld, int, int] [int, int]
5 | [sample/HelloWorld, int, int] [int]
6 | [sample/HelloWorld, int, int, int] []
7 | [sample/HelloWorld, int, int, int] [java/io/PrintStream]
8 | [sample/HelloWorld, int, int, int] [java/io/PrintStream, int]
9 | [sample/HelloWorld, int, int, int] []
10| [] []
```

小结

- 从Instruction的角度来讲，调用static方法是使用invokestatic指令，调用non-static方法一般使用invokevirtual指令。
- 从Frame的角度来讲，实现方法的调用，需要先将this变量和方法接收的参数放到operand stack上。



更多精彩内容，就在简书APP

