

面向对象源码分析 —— Javassist

Part0 前言

Javassist 简介

Javassist 是一个高效的 Java 字节码操作库，专为动态类生成和修改而设计。它通过简洁的 API，使开发者能够在运行时创建新类、修改现有类、插入监控代码和动态重定义类，而无需深入了解 JVM 字节码结构。Javassist 特别适用于 AOP 场景，例如在方法执行前后插入日志或性能监控代码。此外，它还支持生成动态代理类，用于实现懒加载和安全检查等功能。凭借其高效性和灵活性，Javassist 成为动态编程、性能优化和框架开发中的重要工具。

Javassist 的各模块

- `javassist`：核心模块，提供对字节码的操作和生成
- `javassist.bytecode`：提供接口允许程序直接修改字节码
- `javassist.compiler`：将源代码编译为字节码
- `javassist.expr`：提供对方法表达式的支持

我选择分析核心模块，具体为 `ClassPool` 和 `CtClass` 及其相关模块，负责管理和修改字节码。

Part1 主要功能分析与建模

主要功能分析

`ClassPool` 负责加载类并缓存其字节码，以便后续操作，使用示例如下：

```
ClassPool pool = ClassPool.getDefault();
CtClass cc = pool.makeClass("List");
```

`CtClass` 能够添加、删除和修改类的方法，使用示例如下：

```
CtClass ctClass = pool.get("com.example.MyClass");

// 添加新方法
CtMethod newMethod = CtNewMethod.make("public void newMethod() {
System.out.println(\"Hello, world!\"); }", ctClass);
ctClass.addMethod(newMethod);

// 添加新字段
CtField newField = new CtField(CtClass.intType, "newField", ctClass);
ctClass.addField(newField);

// 获取要删除的方法
CtMethod methodToDelete = ctClass.getDeclaredMethod("methodName");
// 删除该方法
ctClass.removeMethod(methodToDelete);

// 获取要修改的方法
CtMethod methodToModify = ctClass.getDeclaredMethod("methodName");
// 修改方法体
```

```
methodToModify.setBody("{ System.out.println(\"This is the new method body.\");
}");

// 插入字节码
methodToModify.insertBefore("{ System.out.println(\"Before method execution\");
}");
methodToModify.insertAfter("{ System.out.println(\"After method execution\");
}");
```

需求建模

根据需求和使用示例可以建立以下模型：

需求模型

【用例名称】

操作一个 Java 类的字节码。

【场景】

- Who：调用者， `ClassPool` 实例， `CtClass` 实例
- Where：内存
- When：编译时、运行时

【用例描述】

1. 调用者创建一个 `CtClass` 实例
 1. 调用者定义类的结构
 2. 调用者选择方法和字段的属性
2. 调用者删除或修改类的方法
3. 调用者调用类的构造函数和方法
4. 类在运行时进行字节码的操作

【用例价值】

动态生成和修改类和方法的字节码

【约束和限制】

修改后的字节码必须符合 Java 虚拟机规范，否则会引发运行时错误。

寻找其中的动词和名词：

动词：创建、定义、添加、选择、删除、修改、调用、动态加载

名词： `CtClass` 实例， `ClassPool` 实例，类结构，成员（方法、字段），方法，字段，构造函数，动态加载类

可以抽象出类和方法：

【类】： `ClassPool`

- 【方法】：创建，添加，获取
- 【属性】：类路径

【类】： `CtClass`

- 【方法】: 定义, 修改, 生成字节码
- 【属性】: 类名, 方法列表, 字段列表

【类】: `CtMethod`

- 【方法】: 添加, 改, 删除
- 【属性】: 方法名, 返回类型, 参数类型

【类】: `CtField`

- 【方法】: 添加, 修改, 删除
- 【属性】: 字段名, 类型

总结

对 `ClassPool` 和 `CtClass` 的阅读让我了解到了 Javassist 的强大功能, 其优秀的集成使得程序可以高效地进行字节码的生成和操作, 实现动态类生成和修改, 满足运行时的需求。