Atitit NoSuchMethodException: No such accessible method: main() on class:

MethodUtils.*invokeStaticMethod*(cls, "main", (Object)new String[]{});

m.invoke(null, (Object)new String[]{});//静态方法不用谢调用的对象

//加Object强制转换的原因

//由于可变参数是JDK5.0之后才有　m.invoke(null, new String[]{"23","34"});

//编译器会把它编译成m.invoke(null,"23","34");的格式,会发生参数不匹配的问题

//带数组的参数都这样做

动态编译虽然是很好的工具，让我们可以更加自如地控制编译过程，但是在我目前所接触的项目中还是使用得较少。原因很简单，静态编译已经能够帮我们处理大部分的工作，甚至是全部的工作，即使真的需要动态编译，也有很好的替代方案，比如JRuby、Groovy 等无缝的脚本语言。

Java Compiler

Java Compiler API，这是JDK6开始提供的标准API，提供了与javac对等的编译功能，即动态编译，文档地址

步骤

通过 Controller 接口，提交Java代码，代码统一实现Callable接口；

动态编译Java代码为字节码；

使用类加载器加载编译的字节码；

使用反射拿到类的元数据信息，执行call方法，完成对应的功能。

作者：废物点心

链接：https://juejin.cn/post/6844903748389584909

来源：掘金

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。