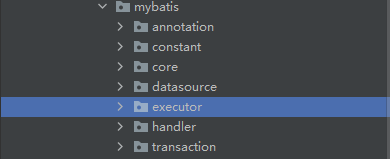
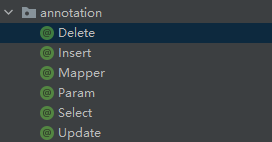
模块总览



1.annotation: 自定义的注解



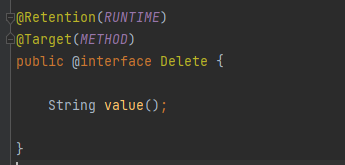
定义方式

@Retention(RUNTIME) //@Retention 用来确定这个注解的生命周期  
@Target(METHOD) //@Target 指定注解使用的目标范围（类、方法、字段等）

public @Interface {name} {

}

Example:

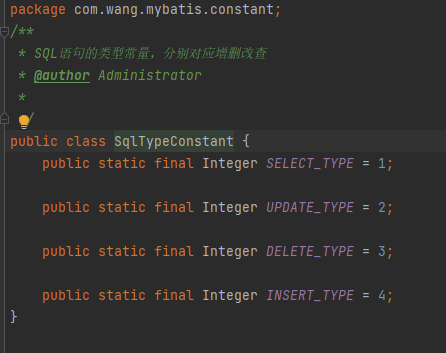


String value();//该注解需要输入String类型的参数

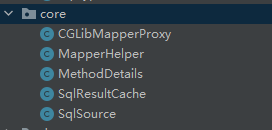
2.constant:常量定义

(1):

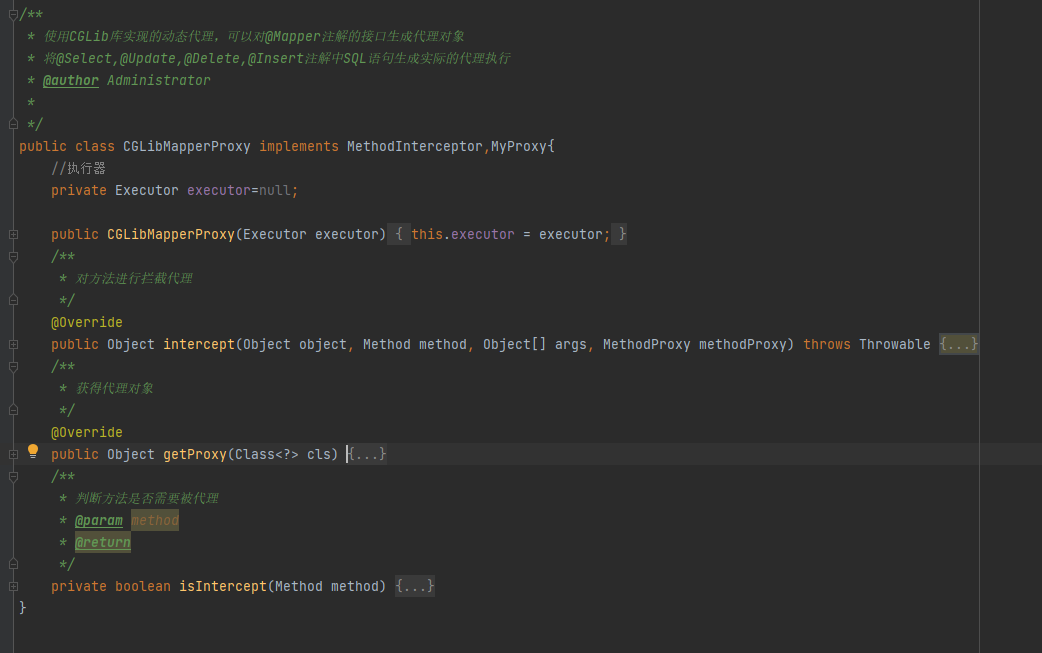




3.core:核心功能

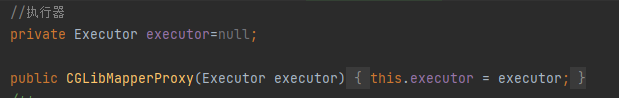


1. CGLibMapperProxy:基于CGLib的mapper映射

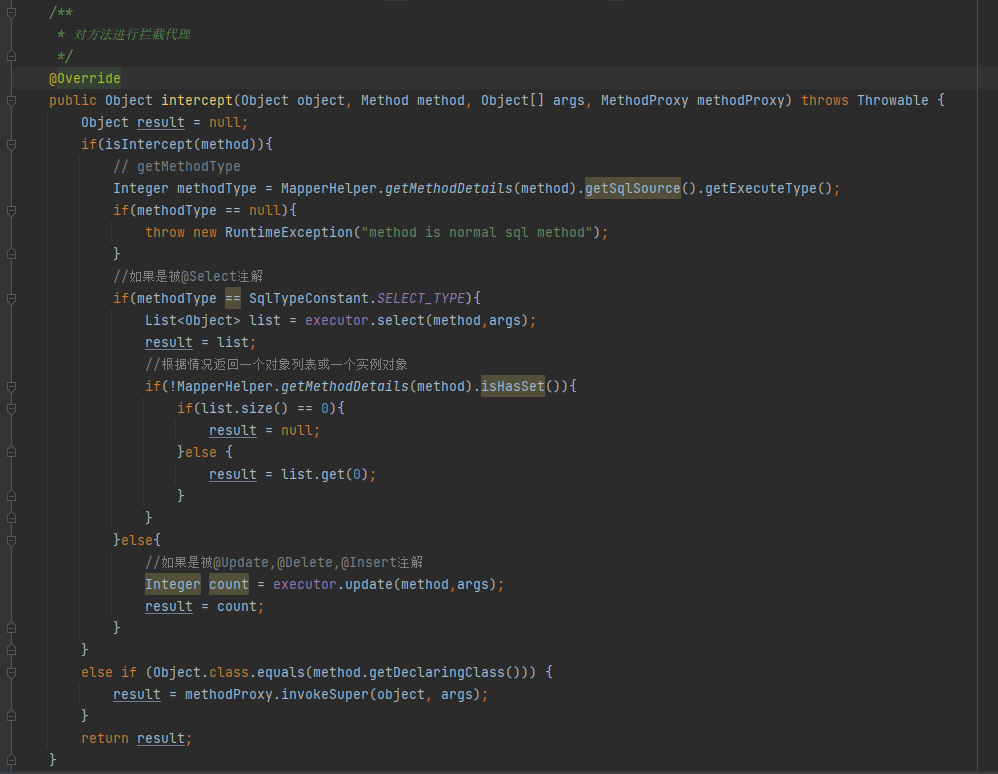


实现接口:MethodInterceptor(intercept)和MyProxy(getProxy)

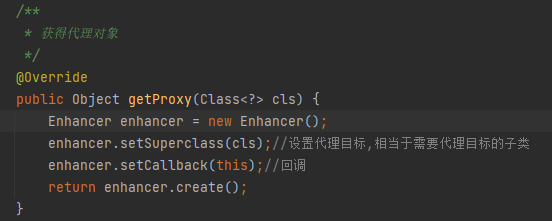
A.类构造函数（初始化一个代理执行器Executor)



1. 重载拦截器，根据注解类型拦截方法



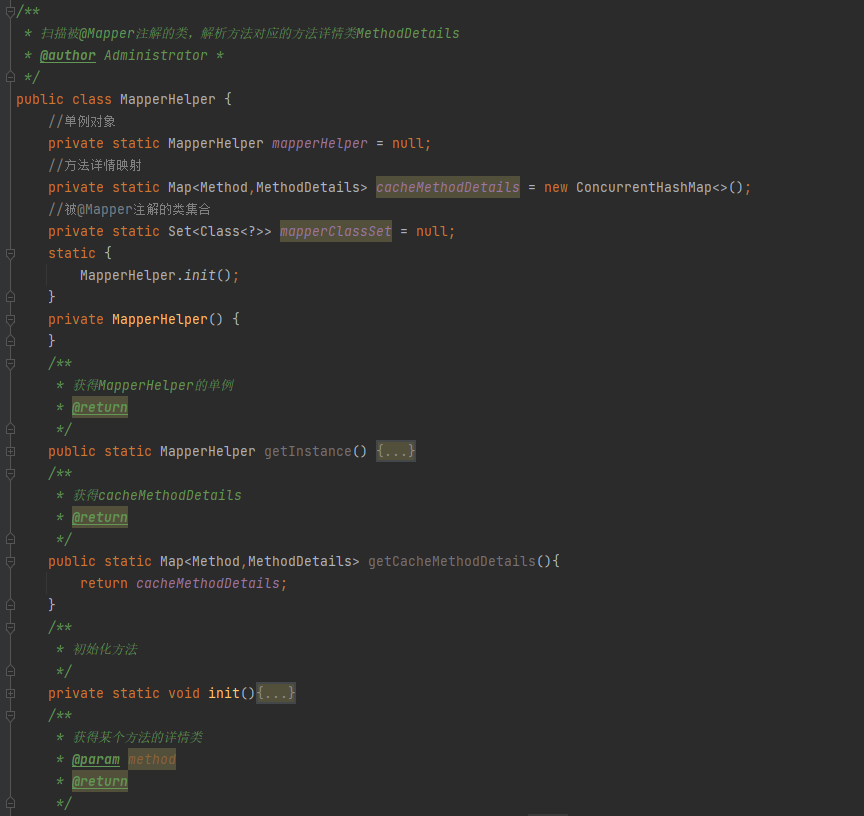
C.获得代理对象（cglib代理方式的类不需要实现接口，而jdk代理方式则需要)

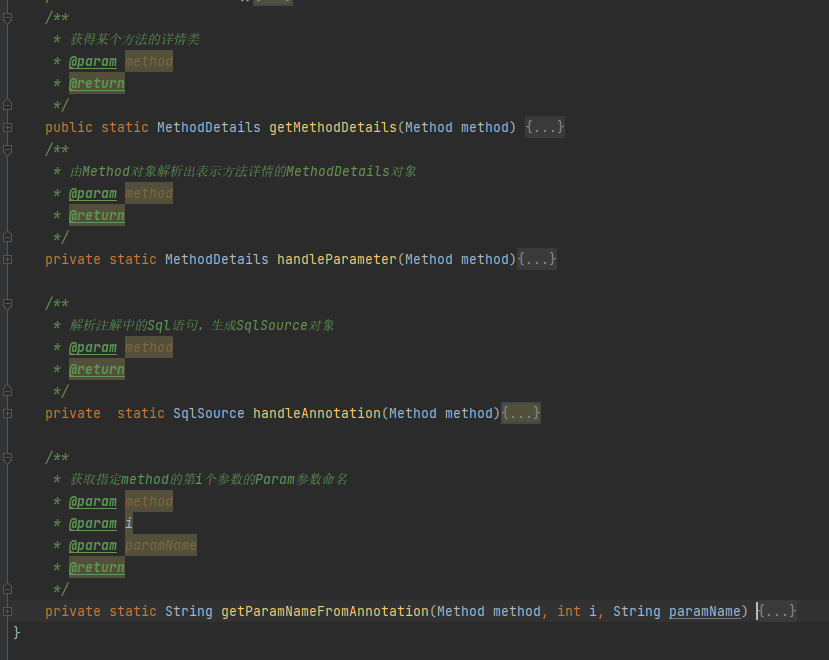


D.判断方法是否需要被代理

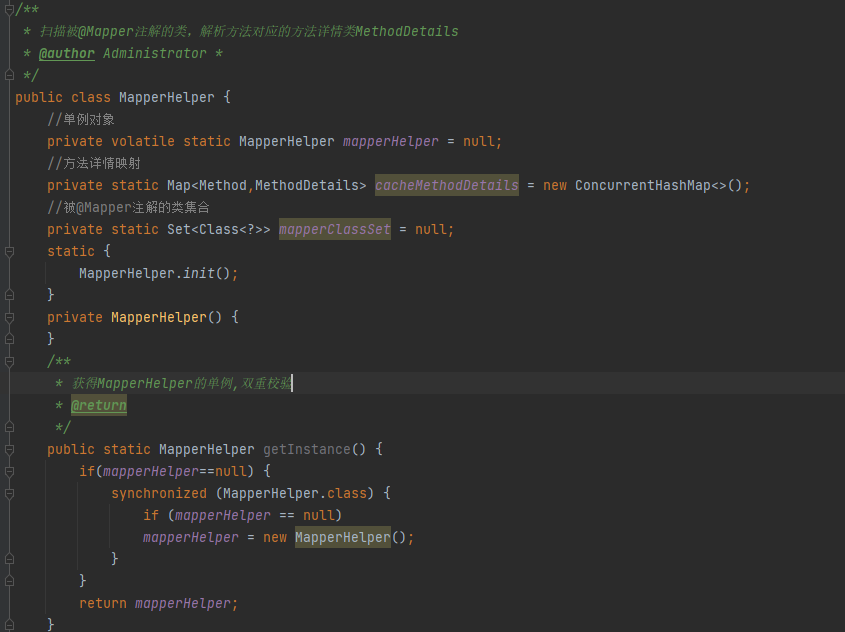


1. 工具类，负责处理被@Mapper注解的类





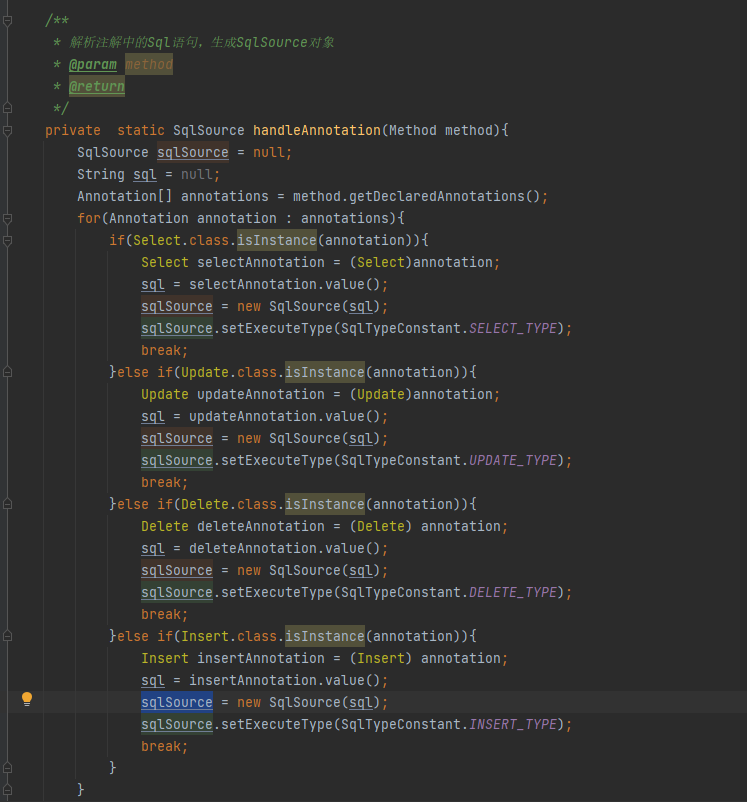
1. 单例构造(双重校验+懒加载）



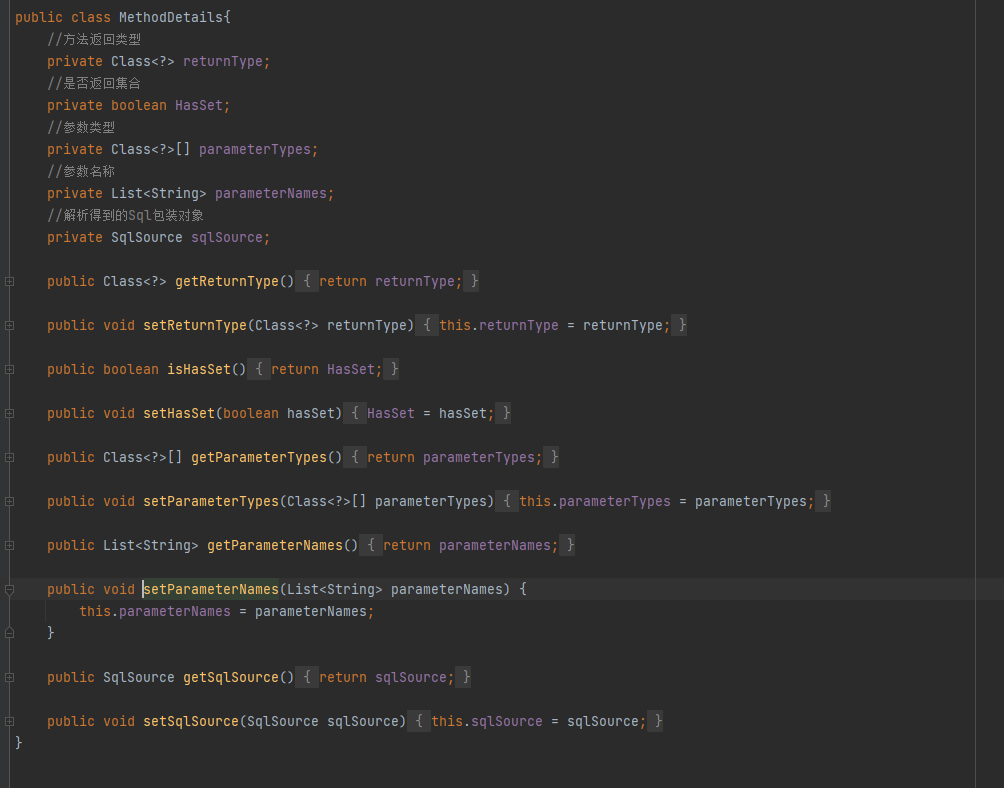
1. 初始化，将被@Mapper注解过的类加入到容器中，并设置代理



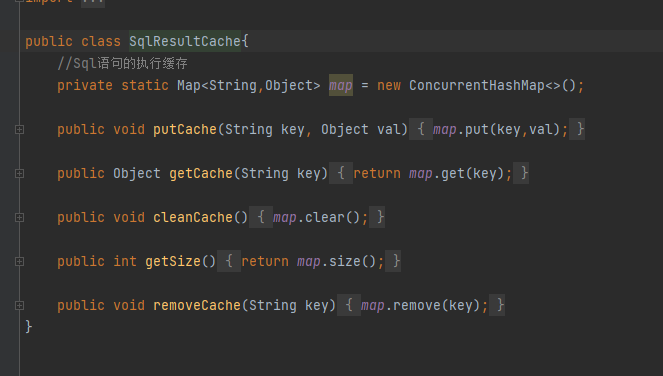
1. 通过方法注解参数，执行对应语句



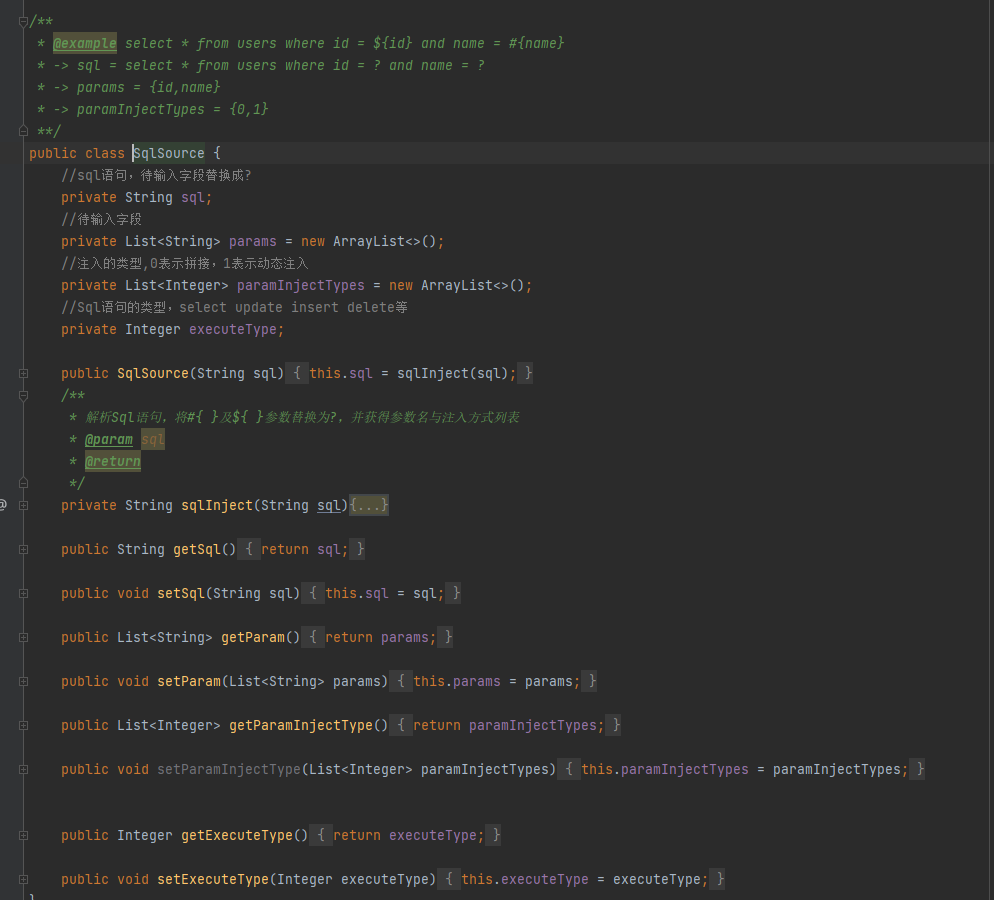
1. 属性类，定义方法所需的各种参数



1. 缓存所需要执行的sql语句，使用concurrentHashMap，线程安全



1. 解析对应的sql语句，映射

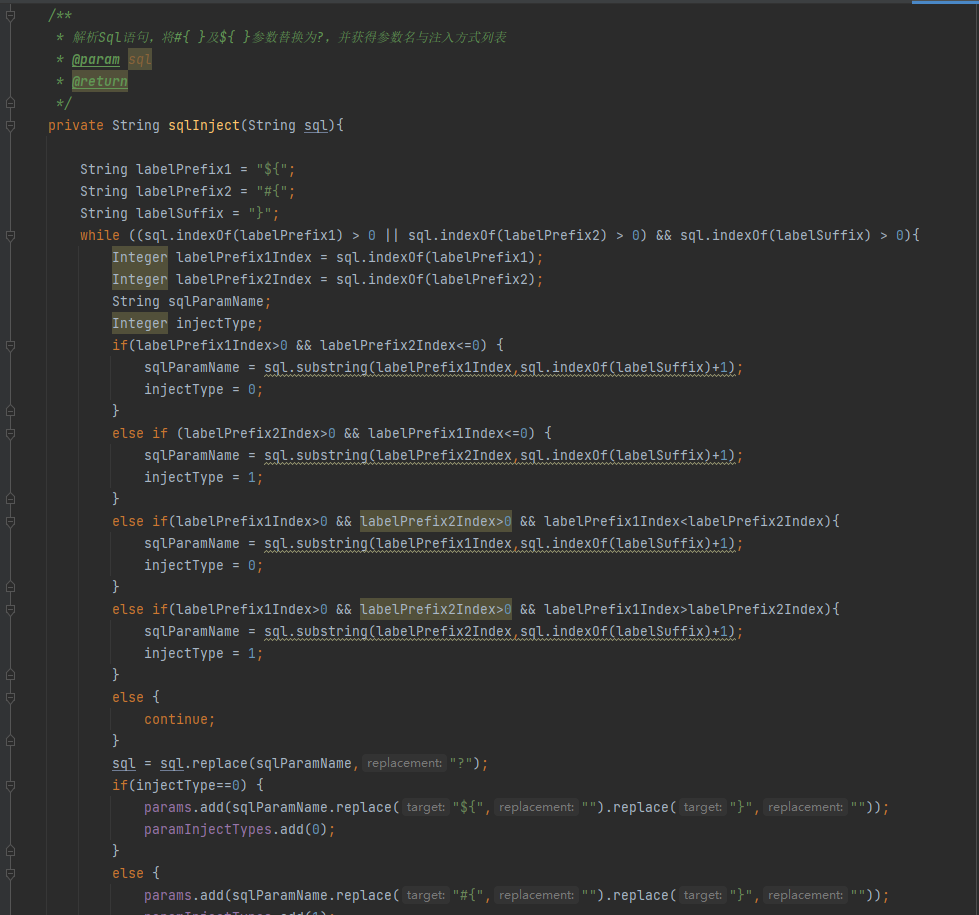


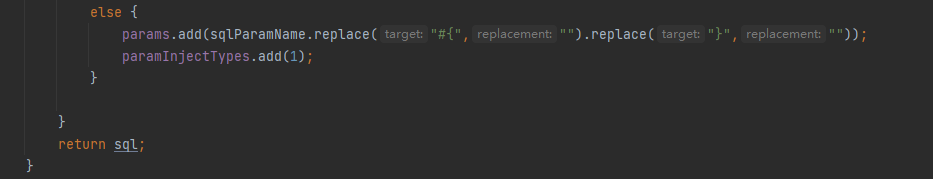
1. ${}和#{} <https://blog.csdn.net/weixin_57529171/article/details/121297305>

#{}：占位符号，可以防止[sql注入](https://so.csdn.net/so/search?q=sql%E6%B3%A8%E5%85%A5&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/weixin_57529171/article/details/_blank)，自己会带有双引号；(#{id} -> “id”}

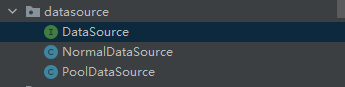
${}：sql拼接符号，存在sql注入问题，需要在代码中过滤以避免注入，不会带有双引号；(${id}-> id)

1. sql解析

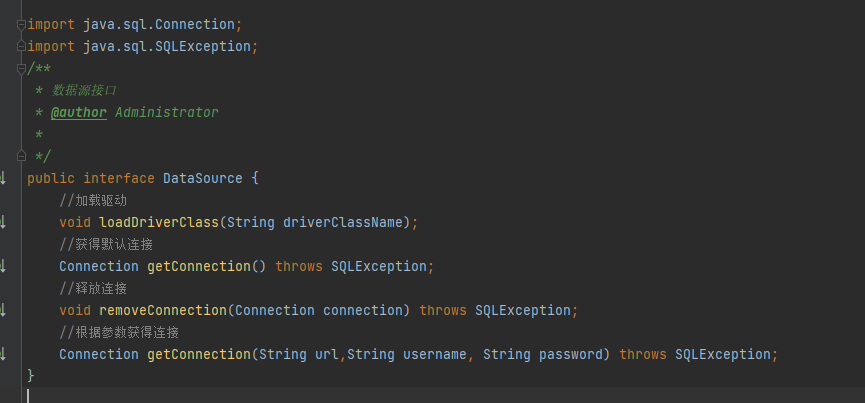




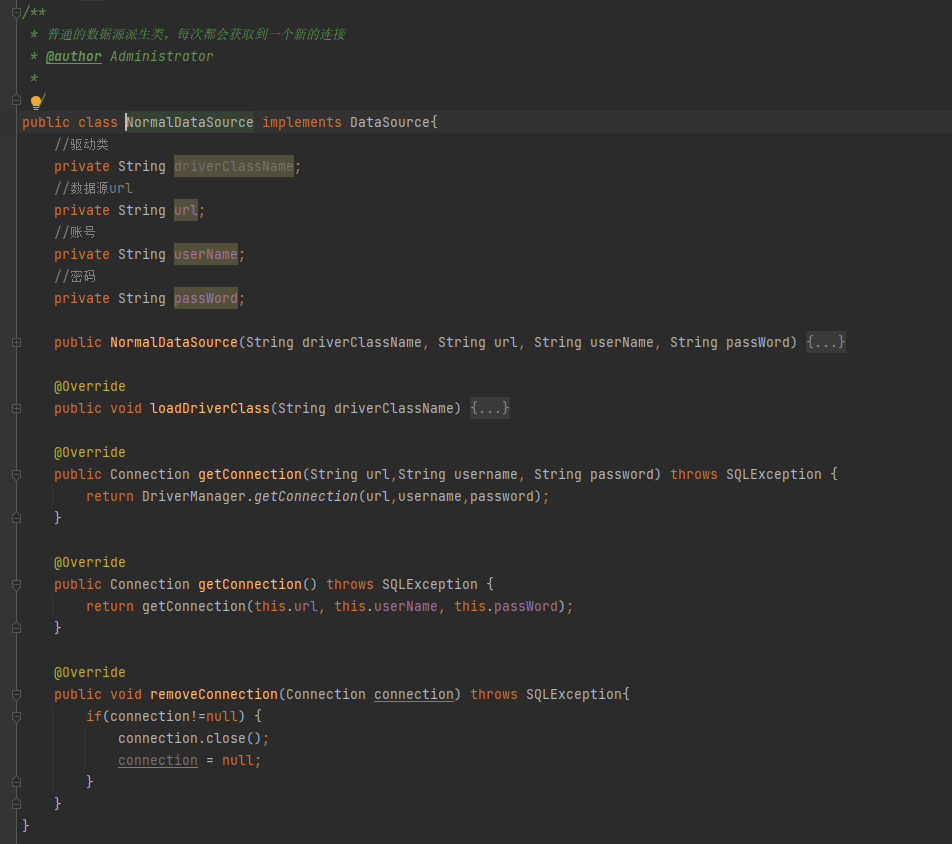
1. 管理数据源



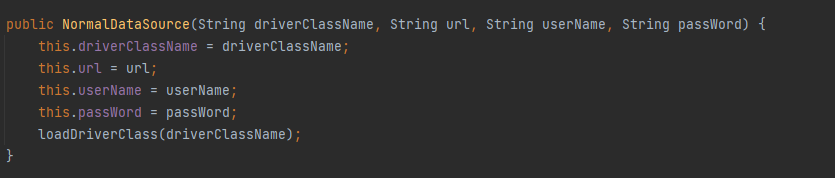
1. .定义功能接口



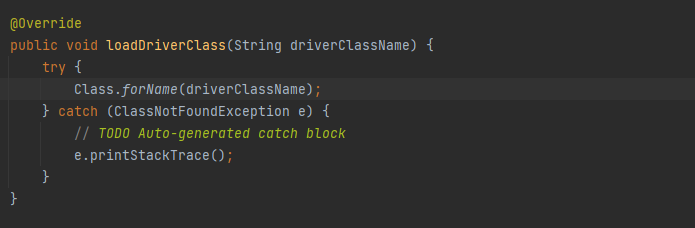
1. 普通数据源



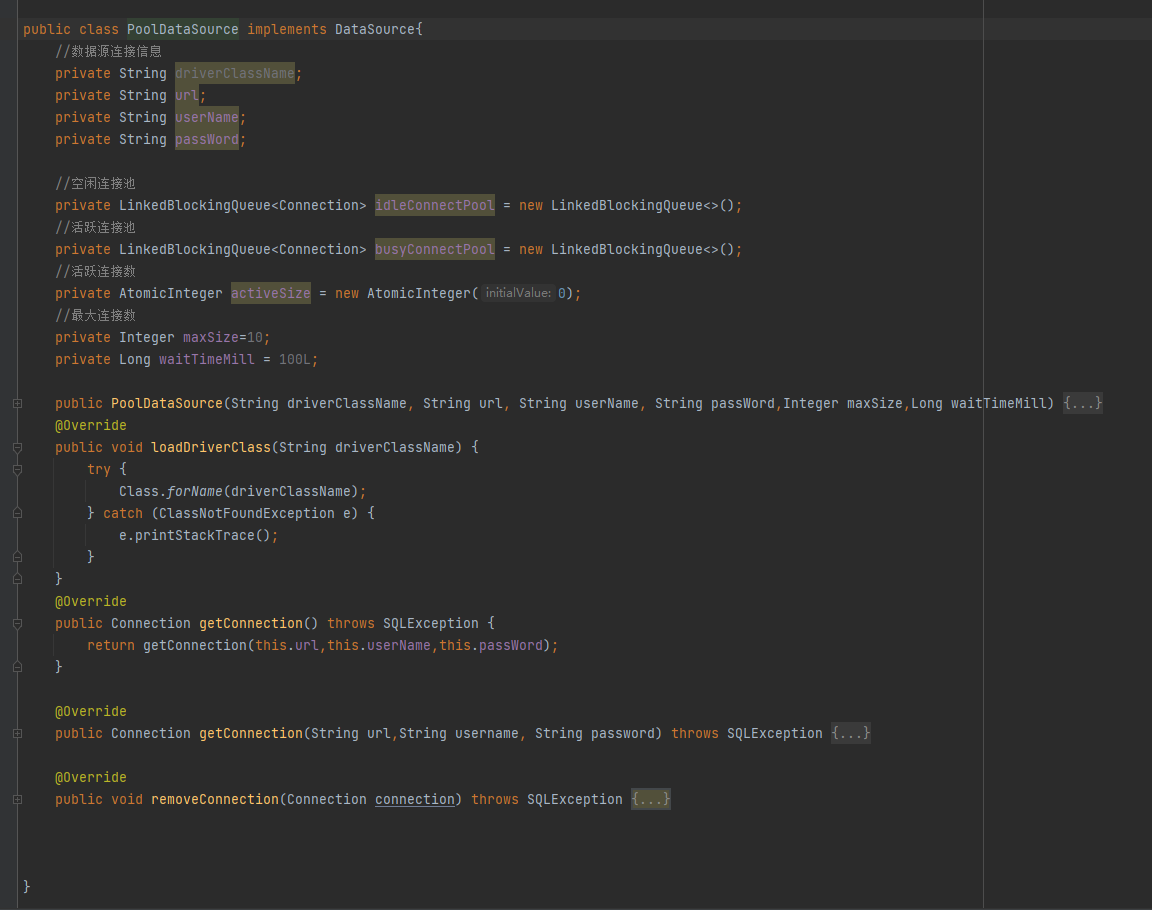
1. 赋值



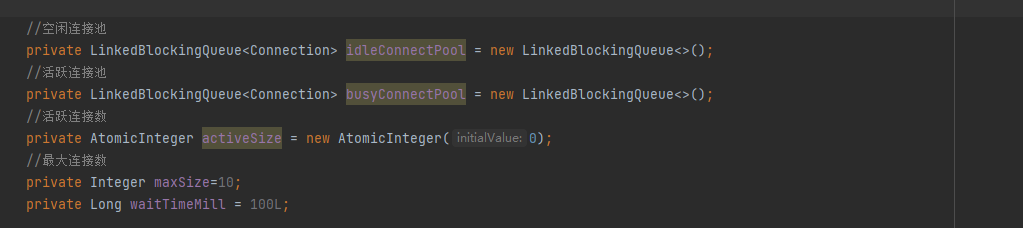
1. 判断对应驱动是否存在



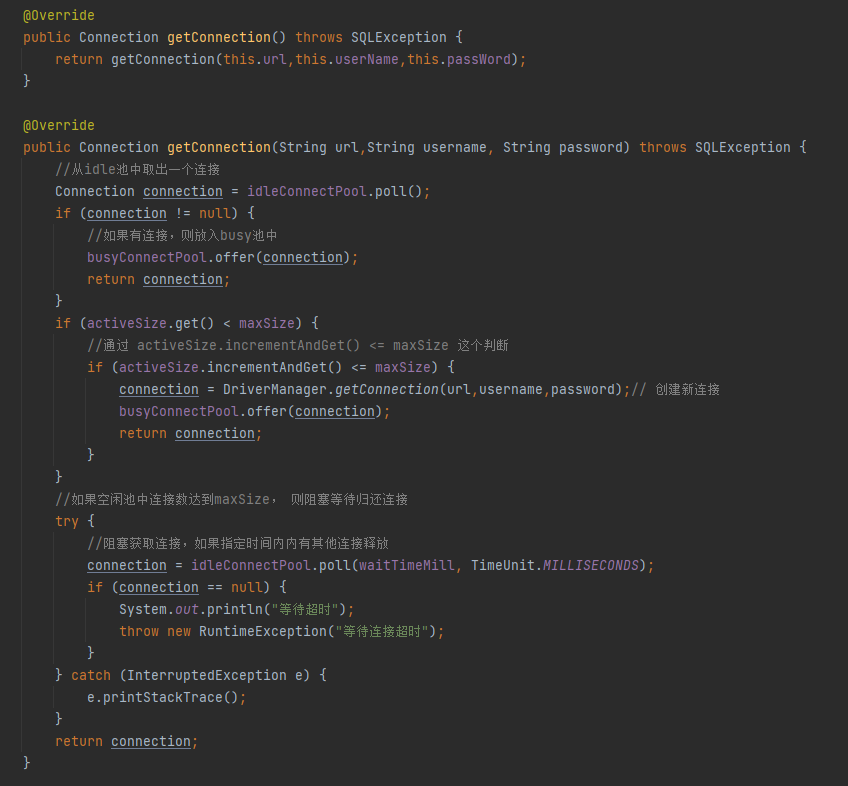
1. 池化数据源



1. 使用阻塞队列保证链接有序



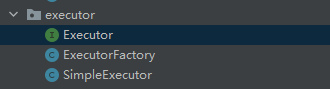
1. 获取链接



1. 移除链接



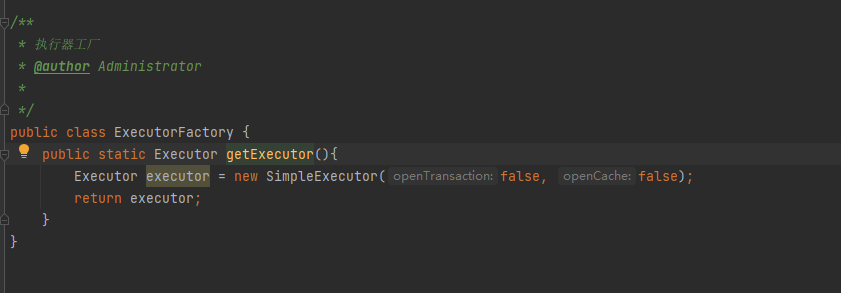
1. 执行器



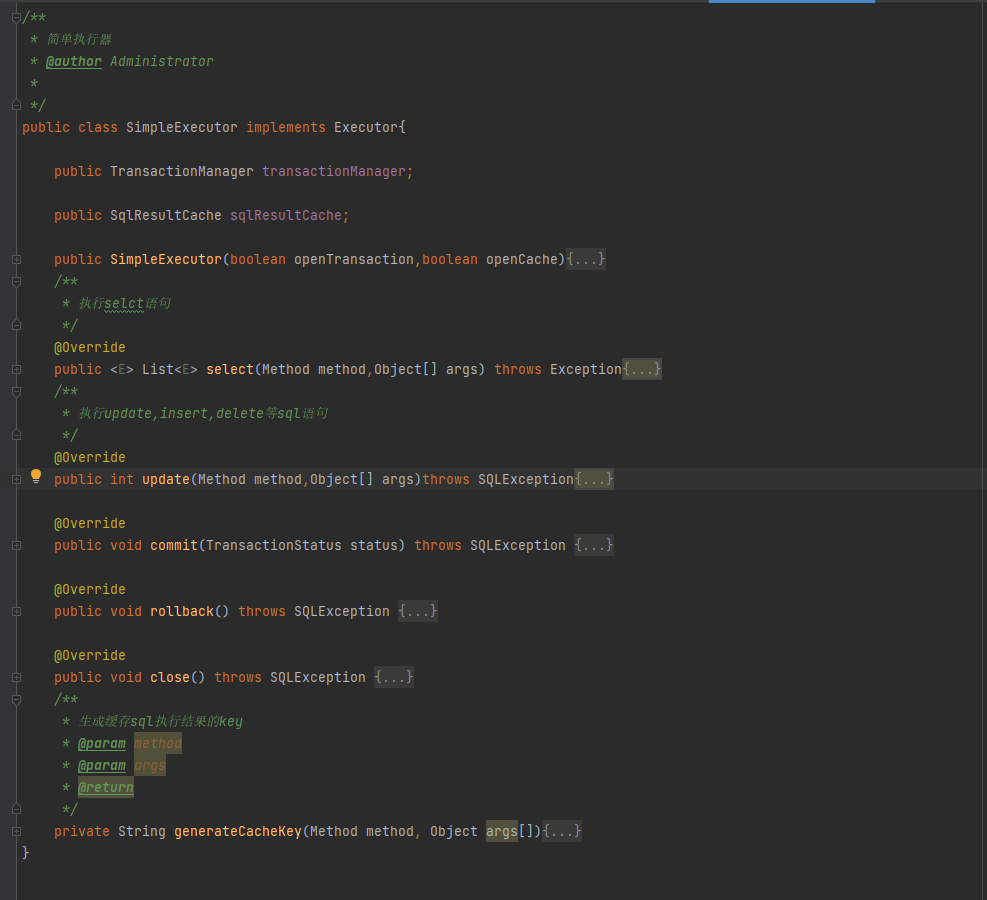
1. 接口定义



1. 工厂模式创建对象



1. 执行器实现



A.