## 什么是mybatis以及优缺点

什么是Mybatis？其实MyBatis 本是apache的一个开源项目iBatis, 2010年这个项目由apache software foundation 迁移到了google code，并且改名为MyBatis 。iBATIS一词来源于“internet”和“abatis”的组合，是一个基于Java的持久层框架。iBATIS提供的持久层框架包括SQL Maps和Data Access Objects（DAO）。

下面来看一下官网对Mybatis的介绍吧：MyBatis 是支持定制化 SQL、存储过程以及高级映射的优秀的持久层框架。MyBatis 避免了几乎所有的 JDBC 代码和手工设置参数以及抽取结果集。MyBatis 使用简单的 XML 或注解来配置和映射基本体，将接口和 Java 的POJOs(Plain Old 映射成数据库中的记录。

MyBatis是iBatis的升级版，用法有很多的相似之处，但是MyBatis进行了重要的改进。例如：

1、Mybatis实现了接口绑定，使用更加方便。

在ibatis2.x中我们需要在DAO的实现类中指定具体对应哪个xml映射文件，而Mybatis实现了DAO接口与xml映射文件的绑定，自动为我们生成接口的具体实现，使用起来变得更加省事和方便。

2、对象关系映射的改进，效率更高

3、MyBatis采用功能强大的基于OGNL的表达式来消除其他元素。

对于IBatis与MyBatis的区别就不再多说了，感兴趣的朋友可以研究一下。

MyBatis的框架架构

看到Mybatis的框架图，可以清晰的看到Mybatis的整体核心对象，我更喜欢用自己的图来表达Mybatis的整个的执行流程。

原理详解：

MyBatis应用程序根据XML配置文件创建SqlSessionFactory，SqlSessionFactory在根据配置，配置来源于两个地方，一处是配置文件，一处是Java代码的注解，获取一个SqlSession。SqlSession包含了执行sql所需要的所有方法，可以通过SqlSession实例直接运行映射的sql语句，完成对数据的增删改查和事务提交等，用完之后关闭SqlSession。

MyBatis的优缺点

优点：

1、简单易学

mybatis本身就很小且简单。没有任何第三方依赖，最简单安装只要两个jar文件+配置几个sql映射文件易于学习，易于使用，通过文档和源代码，可以比较完全的掌握它的设计思路和实现。

2、灵活

mybatis不会对应用程序或者数据库的现有设计强加任何影响。sql写在xml里，便于统一管理和优化。通过sql基本上可以实现我们不使用数据访问框架可以实现的所有功能，或许更多。

3、解除sql与程序代码的耦合

通过提供DAL层，将业务逻辑和数据访问逻辑分离，使系统的设计更清晰，更易维护，更易单元测试。sql和代码的分离，提高了可维护性。

4、提供映射标签，支持对象与数据库的orm字段关系映射

5、提供对象关系映射标签，支持对象关系组建维护

6、提供xml标签，支持编写动态sql。

缺点：

1、编写SQL语句时工作量很大，尤其是字段多、关联表多时，更是如此。

2、SQL语句依赖于数据库，导致数据库移植性差，不能更换数据库。

3、框架还是比较简陋，功能尚有缺失，虽然简化了数据绑定代码，但是整个底层数据库查询实际还是要自己写的，工作量也比较大，而且不太容易适应快速数据库修改。

4、二级缓存机制不佳。

总结

mybatis的优点同样是mybatis的缺点，正因为mybatis使用简单，数据的可靠性、完整性的瓶颈便更多依赖于程序员对sql的使用水平上了。sql写在xml里，虽然方便了修改、优化和统一浏览，但可读性很低，调试也非常困难，也非常受限。mybatis没有hibernate那么强大，但是mybatis最大的优点就是简单小巧易于上手，方便浏览修改sql语句。

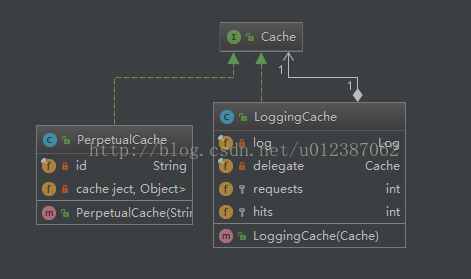
## Mybatis所使用的设计模式

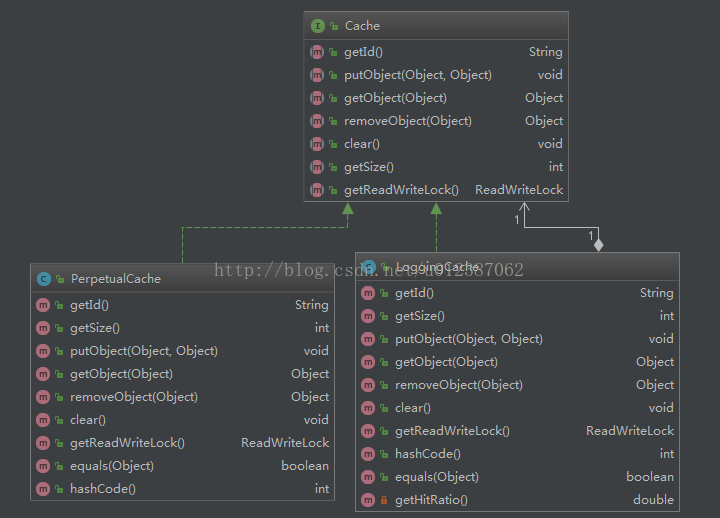
一、装饰模式

最明显的就是cache包下面的实现

Cahe、LoggingCache、LruCache、TransactionalCahe...等

以LoggingCache为例，UML图





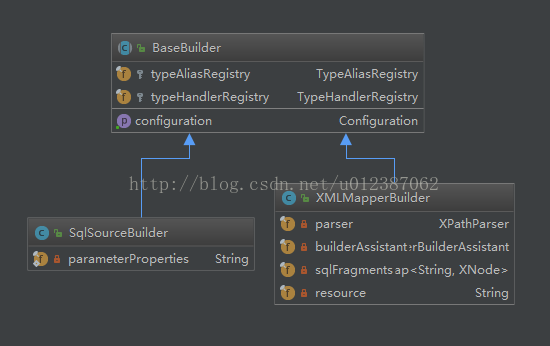
**[html]** [view plain](https://blog.csdn.net/u012387062/article/details/54719114" \o "view plain) [copy](https://blog.csdn.net/u012387062/article/details/54719114" \o "copy)

1. Cache cache  = new LoggingCache(new PerpetualCache("cacheid"));

一层层包装就使得默认cache实现PerpetualCache具有附加的功能，比如上面的log功能。

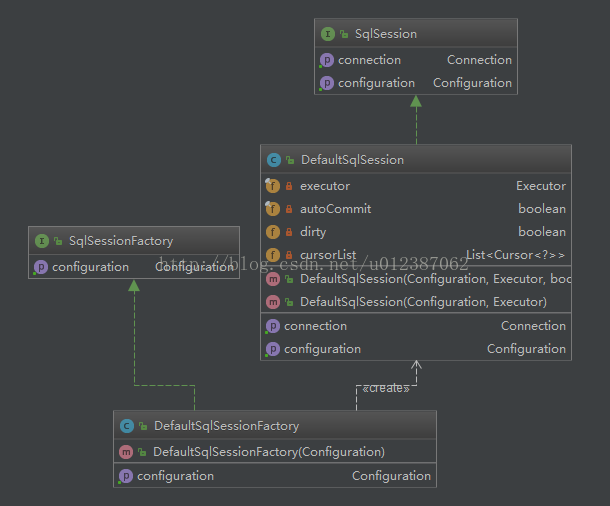
二、建造者模式

BaseBuilder、XMLMapperBuilder



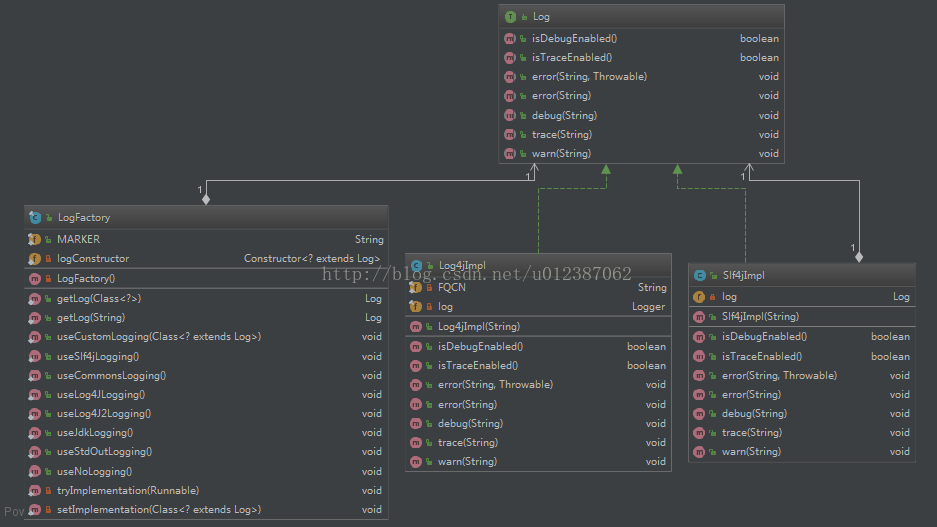
三、工厂方法

SqlSessionFactory



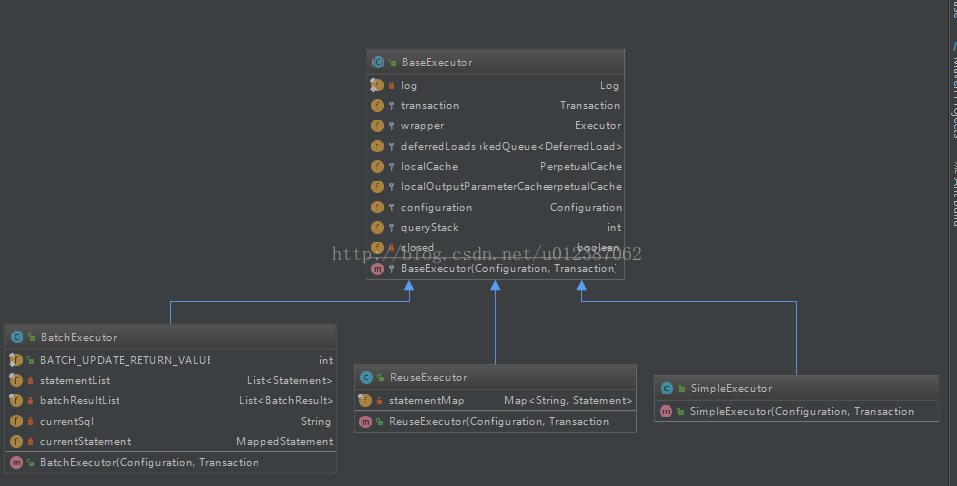
四、适配器模式

Log、LogFactory



五、模板方法

BaseExecutor、SimpleExecutor



六、动态代理