**今天完成的事情：**

**第一部分：JAVA基础**

**学习预览：**

**第二部分：修真任务**

**学习预览：**

* Mybatis

**第二部分**

1. **Mybatis**

**MyBatis特点：**

* MyBatis 是一款优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。
* MyBatis 避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集
* MyBatis 可以使用简单的 XML 或注解来配置和映射原生信息，将接口和 Java 的 POJOs(Plain Ordinary Java Object,普通的 Java对象)映射成数据库中的记录。

1. **Mybatis功能架构**

**Mybatis的功能架构分为三层：**

* **API接口层**

提供给外部使用的接口API，开发人员通过这些本地API来操纵数据库。接口层接收到调用请求就会调用数据处理层来完成具体的数据处理

* **数据处理层**

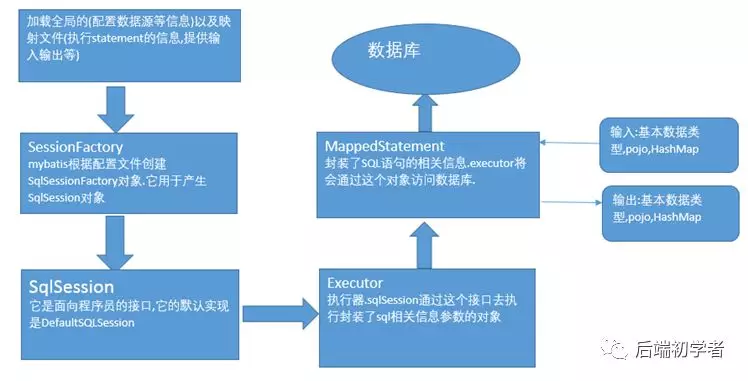
负责具体的SQL查找、SQL解析、SQL执行和执行结果映射处理等。它主要的目的是根据调用的请求完成一次数据库操作

* **基础支撑层**

负责最基础的功能支撑，包括连接管理、事务管理、配置加载和缓存处理，这些都是共用的东西，将他们抽取出来作为最基础的组件。为上层的数据处理层提供最基础的支撑

1. **Mybatis操作流程**

基本流程如下图所示，并配以分步解释：



1. **加载配置并初始化**

**1# 触发条件：**

加载配置文件

**2# 处理过程：**

将SQL的配置信息加载成为一个MappedStatement对象（包括了传入参数映射配置、执行的SQL语句、结果映射配置），存储在内存中

1. **接受调用请求**

**触发条件：**

调用Mybatis提供的API

**传入参数：**

SQL的ID和传入参数对象

**处理过程：**

将请求传递给下层的请求处理层进行处理。

1. **处理操作请求（处理层）**

**触发条件：**

API接口层传递请求过来

**传入参数：**

SQL的ID和传入参数对象

处理过程：

1. 根据SQL的ID查找对应的MappedStatement对象
2. 根据传入参数对象解析MappedStatement对象，得到最终要执行的SQL和执行传入参数
3. 获取数据库连接，根据得到的最终SQL语句和执行传入参数到数据库执行，并得到执行结果
4. 根据MappedStatement对象中的结果映射配置对得到的执行结果进行转换处理，并得到最终的处理结果
5. 释放连接资源
6. 返回处理结果
7. **返回处理结果**
8. **框架架构解析**
9. **加载配置**

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，将SQL的配置信息加载成为一个MappedStatement对象（包括了传入参数映射配置、执行的SQL语句、结果映射配置），存储在内存中

1. **SQL解析（API层）**

当API接口层接收到调用请求时，会接收到传入SQL的ID和传入对象（可以是Map、JavaBean或者基本数据类型），Mybatis会根据SQL的ID找到对应的MappedStatement，然后根据传入参数对象对MappedStatement进行解析，解析后可以得到最终要执行的SQL语句和参数

1. **SQL执行**

将最终得到的SQL和参数拿到数据库进行执行，得到操作数据库的结果

1. **结果映射**

将操作数据库的结果按照映射的配置进行转换，可以转换成HashMap、JavaBean或者基本数据类型，并将最终结果返回

**明天计划完成的事情：**

周末去驾校报名学车，所以计划从下周开始

下周计划完成Mybatis与Spring框架的整合

**遇到的问题：**

除了昨天遇到的JDBCtemplate调试的问题，今天没有遇到别的问题

**收获：**

开始新任务的进程，开始学习Mybatis相关概念