**每日作业卷**

**MyBatis第1天**

极客营

# 关卡1

## 训练案例1

### 训练考核知识点

1. 使用pojo映射文件中的parameterType属性

2. 使用pojo映射文件中的resultType属性

3. 使用MyBatis动态sql标签

4. 使用MyBatis完成两张表一对一关系的查询

5. 使用MyBatis完成两张表一对多关系的查询

### 训练描述

使用课堂资料和已经提供的工程MyBatis02-study实现以下功能

1. 创建包装pojo，包含属性类型为user属性，使用包装pojo实现根据用户名模糊查询

2. 实现查询用户表的数据条数

3. 实现查询订单所有数据功能，并使用resultMap解决用户id没有获取的问题

4. 使用用户性别和名字查询用户，并使用if标签、where标签。

5. 使用sql标签提取Maper.xml中的相同的sql片段

6. 使用foreach标签实现使用多个id查询用户信息

7. 使用resultMap配置，实现订单-用户的一对一查询

8. 使用resultMap配置，实现用户-订单的一对多查询

### 操作步骤描述

1. 编写包装类QueryVo
2. 编写UserMapper接口
3. 编写OrderMapper接口
4. 编写UserMapper.xml
5. 编写OrderMapper.xml
6. 编写测试类UserMapperTest
7. 编写测试类OrderMapperTest

## 训练案例2

### 训练考核知识点

1. spring和MyBatis整合

### 训练描述

使用课堂的数据库表创建新工程，实现MyBatis和Spring的整合

并使用接口动态代理的方式实现DAO的开发，

1.1 根据id查询用户

1.2 根据用户名模糊查询用户

1.3 新增用户"

### 操作步骤描述

1. 创建工程mybatis-spring，导入课堂资料提供的整合jar包

2. 配置实现MyBatis和Spring的整合

编写配置文件SqlMapConfig.xml

编写配置文件applicationContext.xml

加入配置文件db.properties

加入配置文件log4j.properties

3. 使用接口动态代理的方式实现DAO的开发，

1.1 根据id查询用户

1.2 根据用户名模糊查询用户

1.3 新增用户"

编写User类

编写UserMapper接口

编写UserMapper.xml配置文件

修改配置文件applicationContext.xml，添加Mapper的配置

编写测试方法

# 关卡2

## 训练案例1

### 训练考核知识点

1. MyBatis和Spring整合使用

### 训练描述

在关卡1的基础上，

实现MyBatis和Spring的整合开发

要求：

1. 实现查询订单表的数据条数

2. 实现查询所有订单功能，并使用resultMap解决用户id没有获取的问题

3. 使用用户性别和名字查询用户，并使用if标签、where标签。

4. 使用sql标签提取Maper.xml中的相同的sql片段

5. 使用foreach标签实现使用多个id查询用户信息

6. 使用自定义pojo的方式，实现订单-用户的一对一查询

7. 使用resultMap配置，实现订单-用户的一对一查询

8. 使用resultMap配置，实现用户-订单的一对多查询"

### 操作步骤描述

1. 编写pojo
2. 编写UserMapper接口
3. 编写OrderMapper接口
4. 编写UserMapper.xml配置
5. 编写OrderMapper.xml配置
6. 编写UserMapperTest测试类
7. 编写OrderMapperTest测试类

## 训练案例2

### 训练考核知识点

1. MyBatis的逆向工程

### 训练描述

使用课堂提供的逆向工程生成代码

### 操作步骤描述

1. 导入逆向工程到工作空间

2. 修改逆向工程中的配置文件

3. 执行main函数生成代码

4. 把生成的代码复制到到MyBatis整合Spring的工程中。

# 关卡3

## 训练案例1

### 训练描述

使用前一天作业的tb\_order和user表，在mybatis-spring整合的工程中实现DAO的开发。

要求：

1. 实现查询tb\_order的数据条数

2. 使用包装对象，根据用户id查询该用户的所有已付款的订单

3. 根据订单金额payment查询订单，需要查询订单金额为某个范围的订单，需要一对一关联用户数据，需要使用自定义pojo和resultMap两种方式实现。

4. 根据id查询user，需要一对多关联tb\_order

5. 和spring整合使用直接配置Mapper代理对象的方式实现

6. 和spring整合使用Mapper扫描两种方式实现