**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. MyBatis的原始方式实现根据id查询
2. MyBatis的原始方式实现模糊查询
3. MyBatis的原始方式实现新增
4. MyBatis的原始方式实现修改
5. MyBatis的原始方式实现删除

### 训练描述

使用课堂的数据库表和已经提供的工程MyBatis01-study实现以下五个功能

1. 实现根据id查询用户，使用#{}
2. 实现根据用户名模糊查询用户，使用${}
3. 实现新增用户
4. 实现更新用户
5. 实现删除用户
6. 编写接口的实现类，使用SqlSession的用户方法完成

### 操作步骤描述

1. 在MySQL客户端中编写以上五个功能的sql语句
2. 在已经准备好的User.xml页面中编写对应的sql配置
3. 在UserDaoImpl中使用SqlSession实现接口方法
4. 在SqlMapConfig.xml中配置User.xml的加载

<mapper resource=" " />

1. 在UserDaoTest编写测试类，测试五个方法

## 训练案例2

### 训练考核知识点

1. MyBatis的接口动态代理方式实现根据id查询
2. MyBatis的接口动态代理方式实现模糊查询
3. MyBatis的接口动态代理方式实现新增用户
4. MyBatis的接口动态代理方式实现修改用户
5. MyBatis的接口动态代理方式实现删除用户

### 训练描述

使用课堂的数据库表和已经提供的工程MyBatis01-study实现以下五个功能

1. 实现根据id查询用户，使用#{}
2. 实现根据用户名模糊查询用户，使用${}
3. 实现新增用户，并能返回自增的id
4. 实现更新用户
5. 实现删除用户

### 操作步骤描述

1. 在MySQL客户端中编写以上五个功能的sql语句
2. 在已经准备好的UserMapper.xml页面中编写对应的sql配置
3. 在UserDao中编写接口方法
4. 在SqlMapConfig.xml中配置UserMapper.xml的加载

<mapper resource=" " />

1. 在UserMapperTest编写测试类，测试五个方法

# 关卡2

## 训练案例1

### 训练考核知识点

1. MyBatis的接口动态代理方式实现根据id查询
2. MyBatis的接口动态代理方式实现模糊查询
3. MyBatis的接口动态代理方式实现新增并返回主键
4. MyBatis的接口动态代理方式实现修改
5. MyBatis的接口动态代理方式实现删除

### 训练描述

使用课堂的数据库表和已经提供的工程MyBatis01-study实现以下五个功能

1. 实现根据id查询订单，使用${}
2. 实现根据备注note模糊查询订单，使用#{}
3. 实现新增订单，并能返回自增的id
4. 实现根据id更新订单note
5. 实现根据id删除订单

### 操作步骤描述

1. 在MySQL客户端中编写以上五个功能的sql语句
2. 在OrderMapper.xml页面中编写对应的sql配置
3. 在OrderMapper中编写接口方法
4. 在SqlMapConfig.xml中配置OrderMapper.xml的加载

<mapper resource=" " />

1. 在OrderMapperTest编写测试类，测试五个方法

## 训练案例2

### 训练考核知识点

1. MyBatis的Properties
2. MyBatis的别名
3. MyBatis的Mapper三种加载方式

### 训练描述

在之前实现的功能上，实现以下操作：

1. 配置properties，加载db.properties
2. 配置使用别名
3. 使用三种方式加载Mapper.xml配置文件

### 操作步骤描述

1. 在SqlMapConfig.xml中进行配置，使之加载db.properties，并使用db.properties的配置信息
2. 在SqlMapConfig.xml中配置别名扫描包
3. 在Mapper.xml中配置使用别名
4. 在Mapper.xml中配置使用类路径的方式加载Mapper.xml
5. 在Mapper.xml中配置使用包扫描的方式加载Mapper.xml

# 关卡3

## 训练案例1

### 训练描述

先根据以下sql创建数据库表：

CREATE TABLE `tb\_order` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT COMMENT '订单id',

`payment` varchar(50) COLLATE utf8\_bin DEFAULT NULL COMMENT '实付金额。精确到2位小数;单位:元。如:200.07，表示:200元7分',

`paymentType` int(2) DEFAULT NULL COMMENT '支付类型，1、在线支付，2、货到付款',

`postFee` varchar(50) COLLATE utf8\_bin DEFAULT NULL COMMENT '邮费。精确到2位小数;单位:元。如:200.07，表示:200元7分',

`status` int(10) DEFAULT NULL COMMENT '状态：1、未付款，2、已付款，3、未发货，4、已发货，5、交易成功，6、交易关闭',

`createTime` datetime DEFAULT NULL COMMENT '订单创建时间',

`updateTime` datetime DEFAULT NULL COMMENT '订单更新时间',

`paymentTime` datetime DEFAULT NULL COMMENT '付款时间',

`consignTime` datetime DEFAULT NULL COMMENT '发货时间',

`endTime` datetime DEFAULT NULL COMMENT '交易完成时间',

`closeTime` datetime DEFAULT NULL COMMENT '交易关闭时间',

`userId` bigint(20) DEFAULT NULL COMMENT '用户id',

`buyerMessage` varchar(100) COLLATE utf8\_bin DEFAULT NULL COMMENT '买家留言',

`buyerNick` varchar(50) COLLATE utf8\_bin DEFAULT NULL COMMENT '买家昵称',

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_bin;

使用以上数据库表，使用接口动态代理的方式实现以下功能:

1. 创建pojo
2. 实现创建订单，并返回订单id
3. 实现根据订单id查询订单
4. 实现根据订单创建的时间范围(例如查询3月26日这一天的)，查询所有支付类型为在线支付，而且已付款的订单

*查询范围可以使用BETWEEN关键词，例如以下sql是查询金额为11和31之间的订单*

*SELECT \* FROM tb\_order WHERE payment BETWEEN 10 AND 31*

1. 实现根据买家留言模糊查询订单
2. 实现根据订单id修改订单
3. 实现根据订单创建的时间范围(例如查询3月26日这一天的)，修改所有支付类型是在线支付的，而且未付款的，把订单状态修改为交易关闭
4. 实现根据id删除订单