

实验 5 学生管理系统实验程序

一、学生管理系统需求描述

需求：现有一个学生表t_students和一个班级表t_class，其中，班级表t_class和学生表t_students是一对多的关系。

学生表 t_students			
学生 id id	学生姓名 name	学生年龄 age	所属班级 cid
1	张三	18	1
2	李四	18	2
3	王五	19	2
4	赵六	20	1

班级表 t_class	
班级 id id	班级名称 classname
1	一班
2	二班

具体实现要求：

使用 MyBatis 注解完成以下几个要求：

1、MyBatis 注解实现查询操作：

根据表 t_student 和表 t_class 在数据库分别创建一个学生表 t_student 和一个班级表 t_class，并查询 id 为 2 的学生的信息。

2、MyBatis 注解实现修改操作

修改 id 为 4 的学生的姓名修改为李雷，年龄修改为 21。

3、MyBatis 注解实现一对多查询

查询出二班所有学生的信息。

二、开发步骤

第 1 步：数据库开发

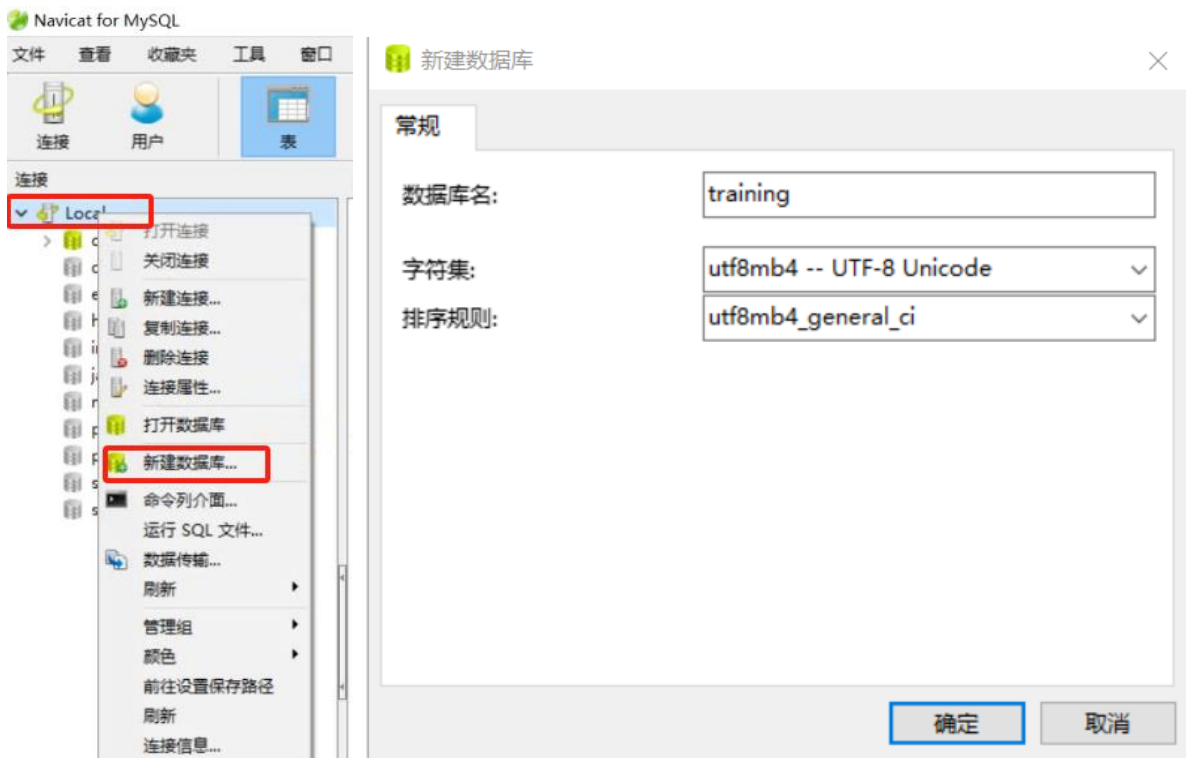
先使用 MySQL 数据库创建一个 training 数据库，然后创建一个学生表 t_students 和班级表 t_class，方法如下：

通过使用 Navicat 工具，连接上你安装的 MySQL 数据库，输入用户名和密码进行登录。

假设建立的 MySQL 连接名为 **Local**，数据库登录用户名和密码为：**root/123456**。

登录后，手动创建数据库，需要创建的数据库名为：**training**，如图：选中 **Local** 连接名，

右键选择【新建数据库】



数据库创建好后，创建学生表 **t_students** 和班级表 **t_class**，表结构如下。并按要求加入测试数据。

```
use training;
```

```
CREATE TABLE t_class (  
    id int(32) PRIMARY KEY AUTO_INCREMENT,  
    classname varchar(40)  
);
```

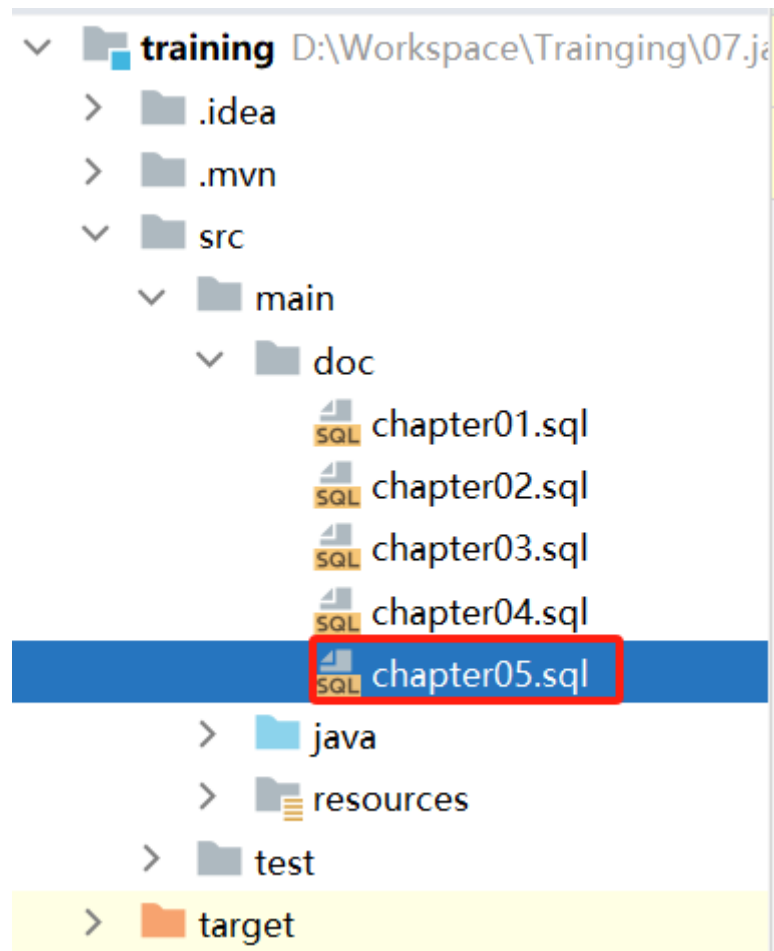
```
INSERT INTO t_class VALUES (1, '一班');  
INSERT INTO t_class VALUES (2, '二班');
```

```
CREATE TABLE t_students (  
    id int(32) PRIMARY KEY AUTO_INCREMENT,  
    name varchar(40),  
    age int,  
    cid int(32) NOT NULL,  
    FOREIGN KEY(cid) REFERENCES t_class(id)  
);
```

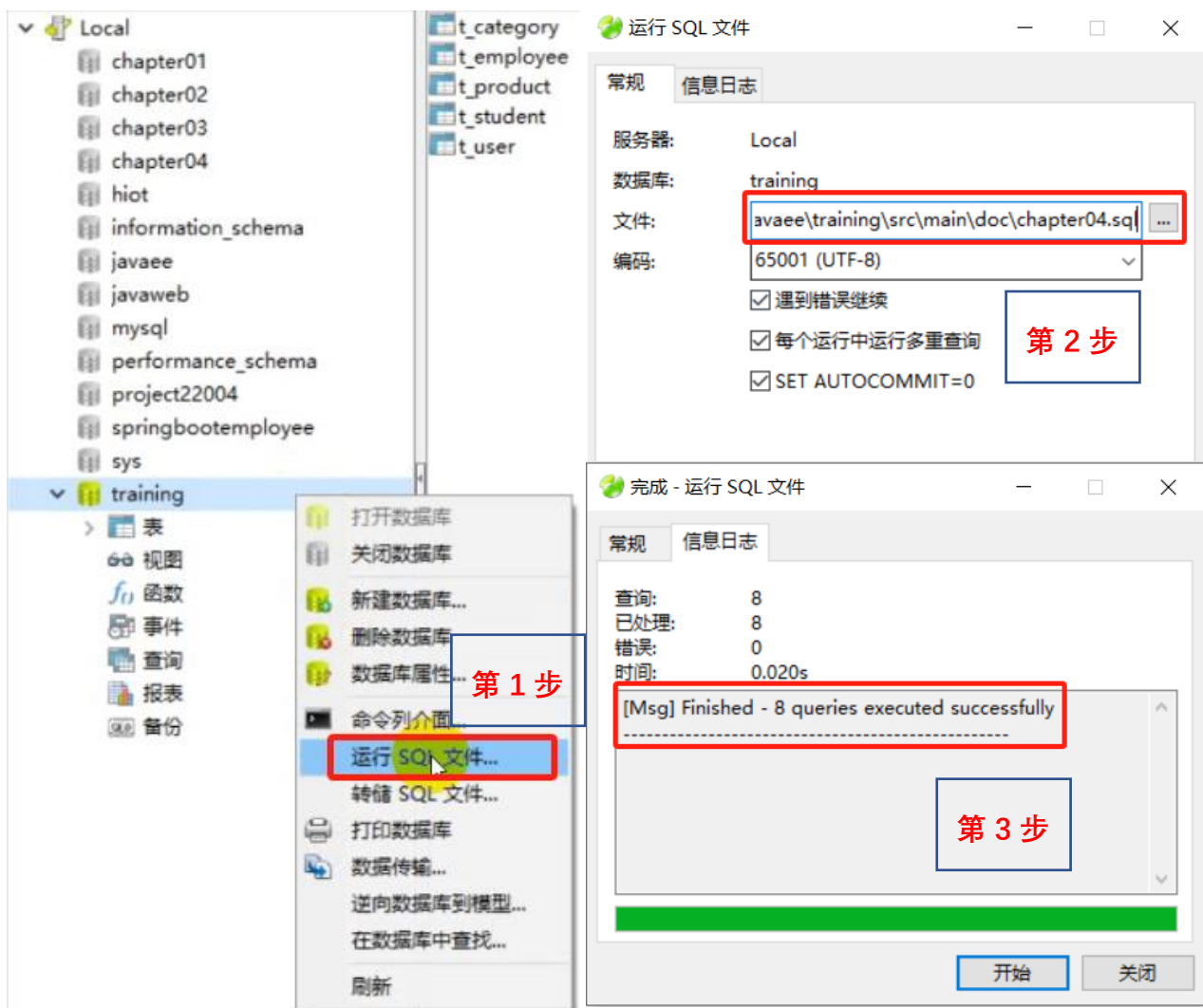
```
INSERT INTO t_students VALUES (1, '张三', 18, 1);  
INSERT INTO t_students VALUES (2, '李四', 18, 2);  
INSERT INTO t_students VALUES (3, '王五', 19, 2);  
INSERT INTO t_students VALUES (4, '赵六', 20, 1);
```

至此，数据库部分开发完毕。

注：以上建表数据库脚本已放置工程的 **doc** 目录，文件路径如下图所示：



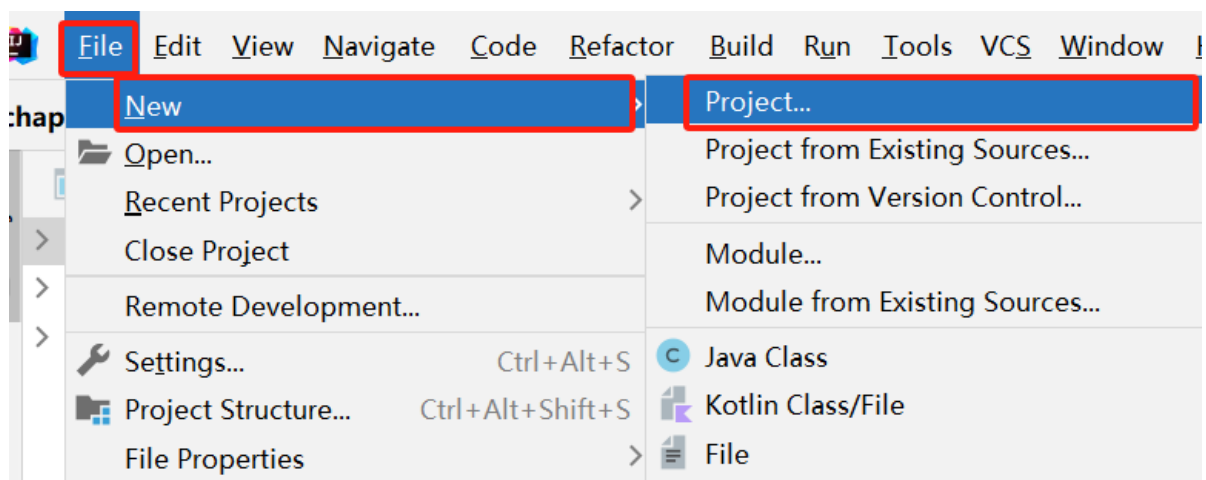
可以通过 Navicat 工具直接读取 chapter05.sql 脚本文件，以完成数据表的创建。



第 2 步：创建 Maven 工程

方式一：直接创建新的 Maven 工程

按如下示例的步骤，进行工程创建（基于 IDEA 2022）



New Project

×

?

Generators

Maven Archetype

Jakarta EE

Spring Initializr

JavaFX

Quarkus

Micronaut

Ktor

Kotlin Multiplatform

Compose Multiplatform

HTML

React

Express

Angular CLI

IDE Plugin

Android

Vue.js

Vite

Name:

training

项目名称

Location:

D:\Workspace\Trainging\07.javaee\training

项目存放的目录

Project will be created in: D:\Workspace\Trainging\07.javaee\chapter01\demo

☐ Create Git repository

Template:

Library

Dependencies only

Application server:

Tomcat 9.0.44

New...

Language:

Java

Kotlin

Groovy

Build system:

Maven

Gradle

Group:

com.xhxc

项目源代码包命名路径

Artifact:

training

与项目名称保持一致即可

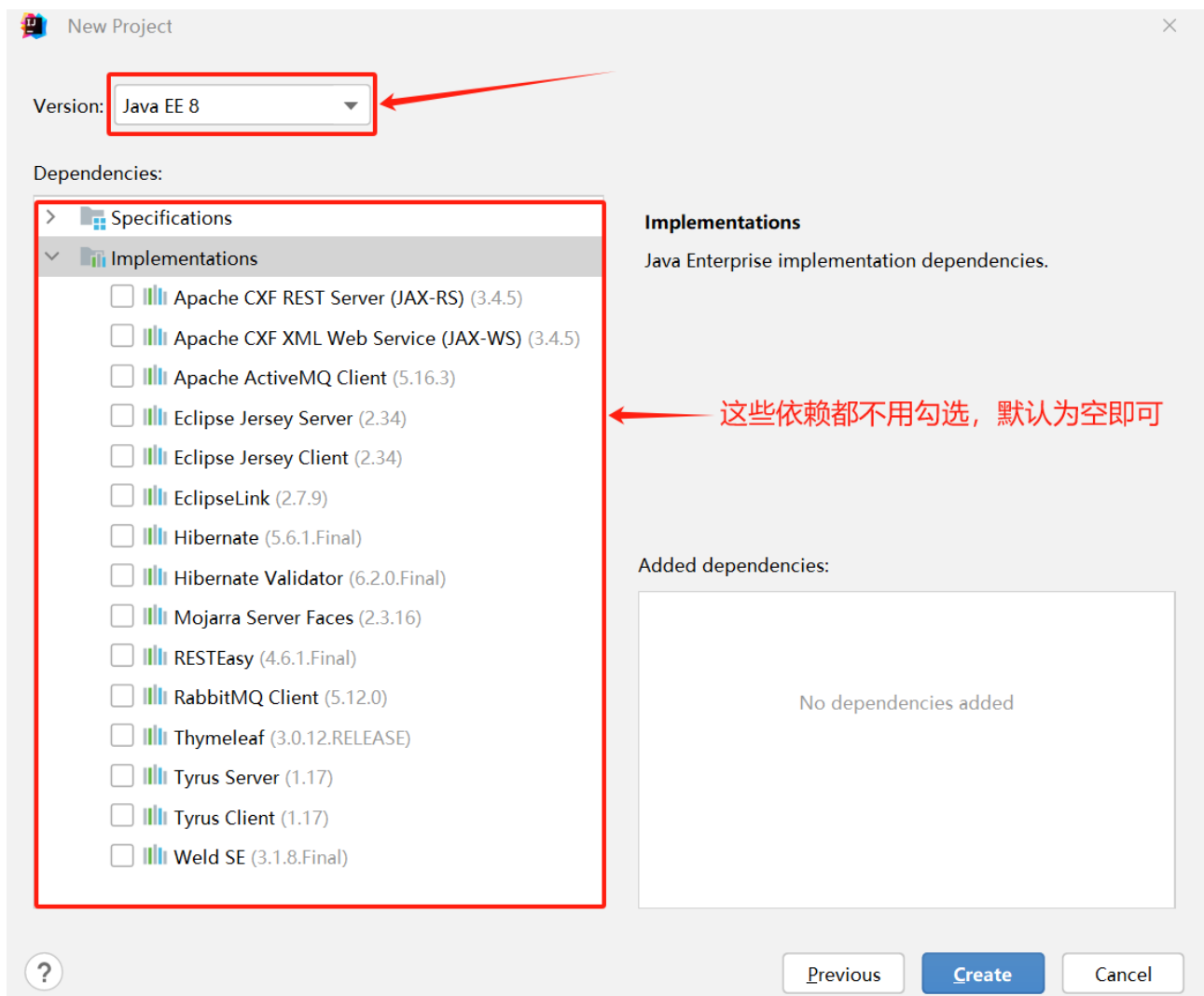
JDK:

1.8 Oracle OpenJDK version 1.8.0_191

JDK版本

Next

Cancel




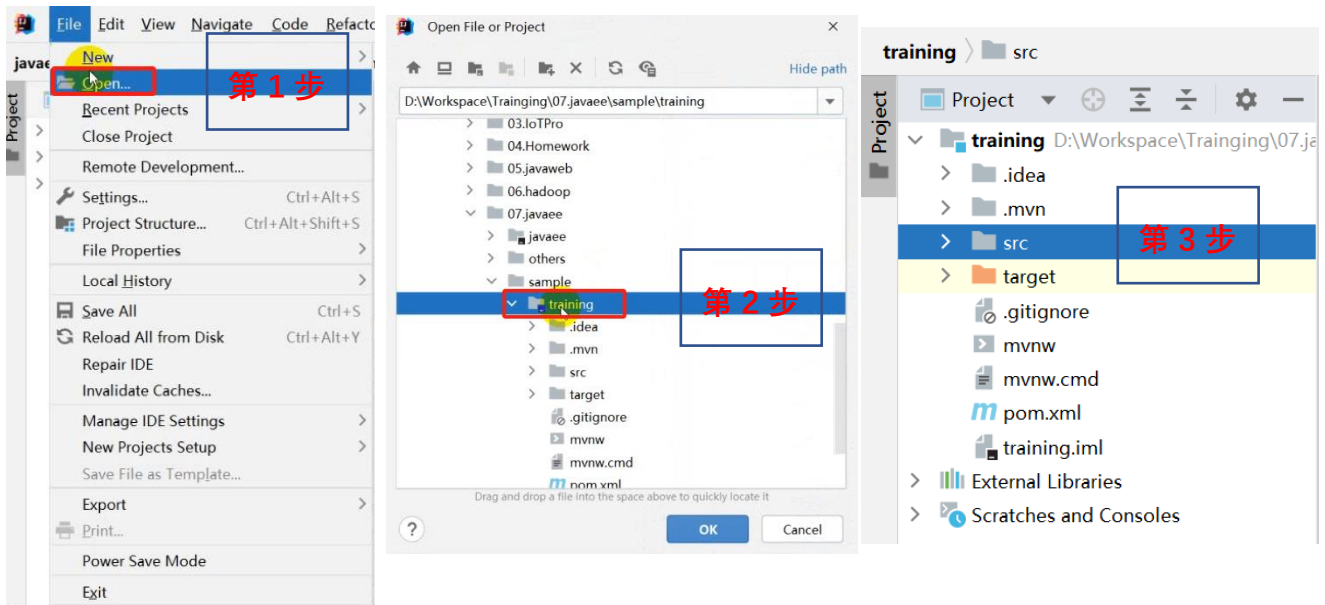
最后一步，点击【**Create**】即可完成工程创建。

方式二：导入已存在的 Maven 工程（推荐）

Step1，下载并解压 training 工程。从【学习通】→【章节】→【第五章…】→【上机实验 5…】，找到 training.zip 压缩包，下载到本地，并解压成 training 目录。

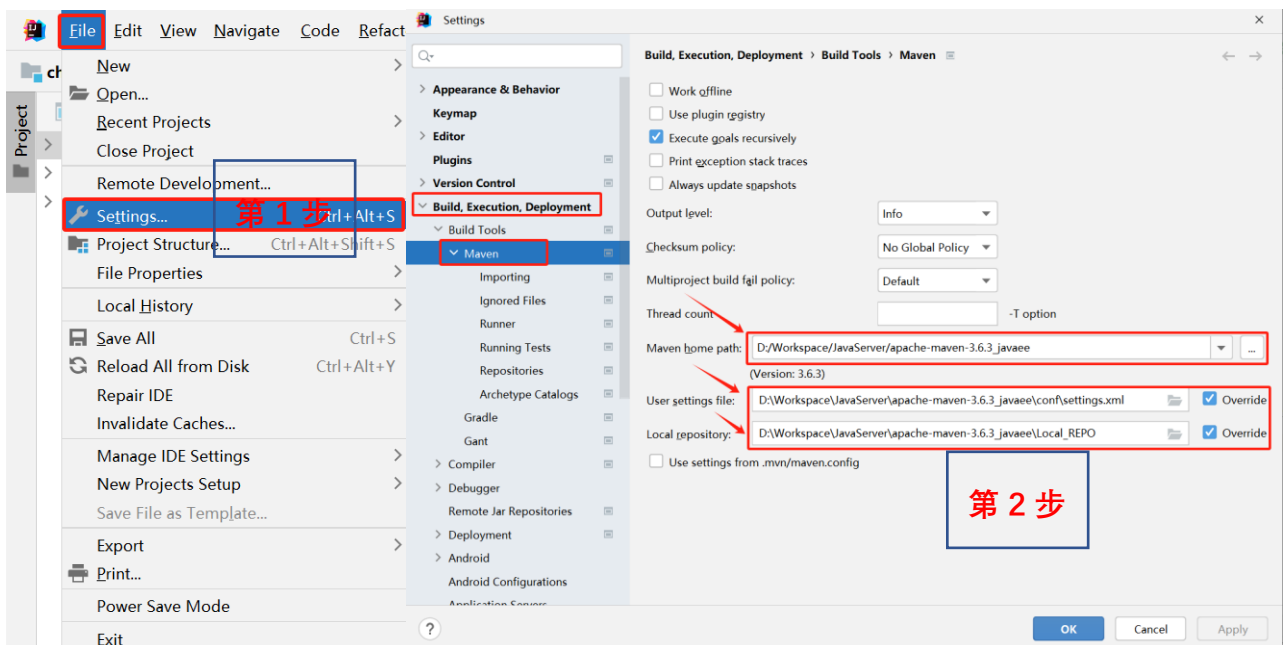
Step2，导入 training 工程。打开 IDEA 工具，点击【File】→【Open…】→选中上一步解压的 training 工程，如下图，确定即可导入。

注意，导入工程后，对于 IDEA 2023 的版本，要点击左上角边的  文件夹小图标，即可查看到工程中的文件。



第 3 步：配置本地 Maven 路径

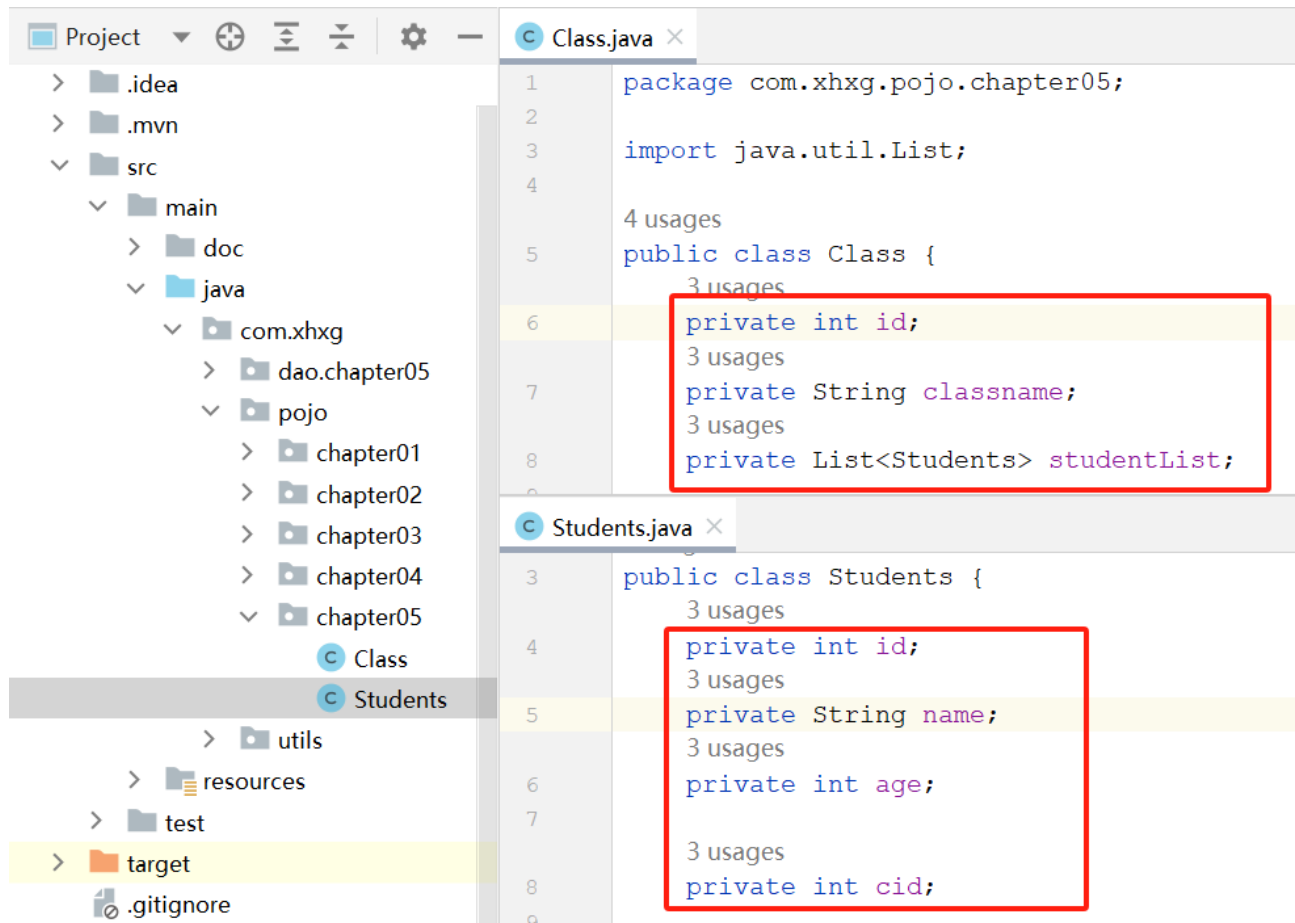
选择【File】→【Setting...】然后找到菜单【Build, Execution, Deployment】→【Build Tools】→【Maven】，按如下图所示，将你的本地 Maven 路径和配置文件设置好即可。



第 4 步：程序代码开发

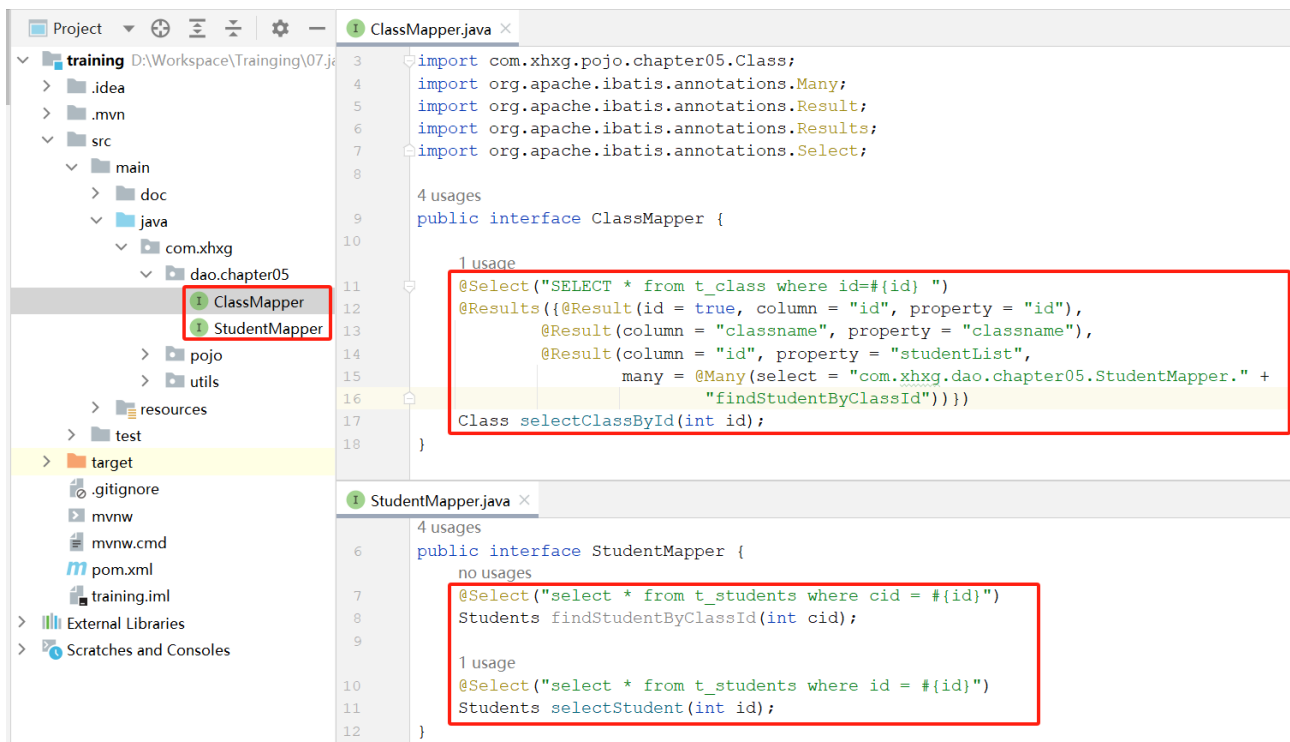
1、创建 POJO 实体

打开 training 项目工程。创建代码包为：com.xhcg.pojo.chapter05，并且创建 Students 和 Class 类



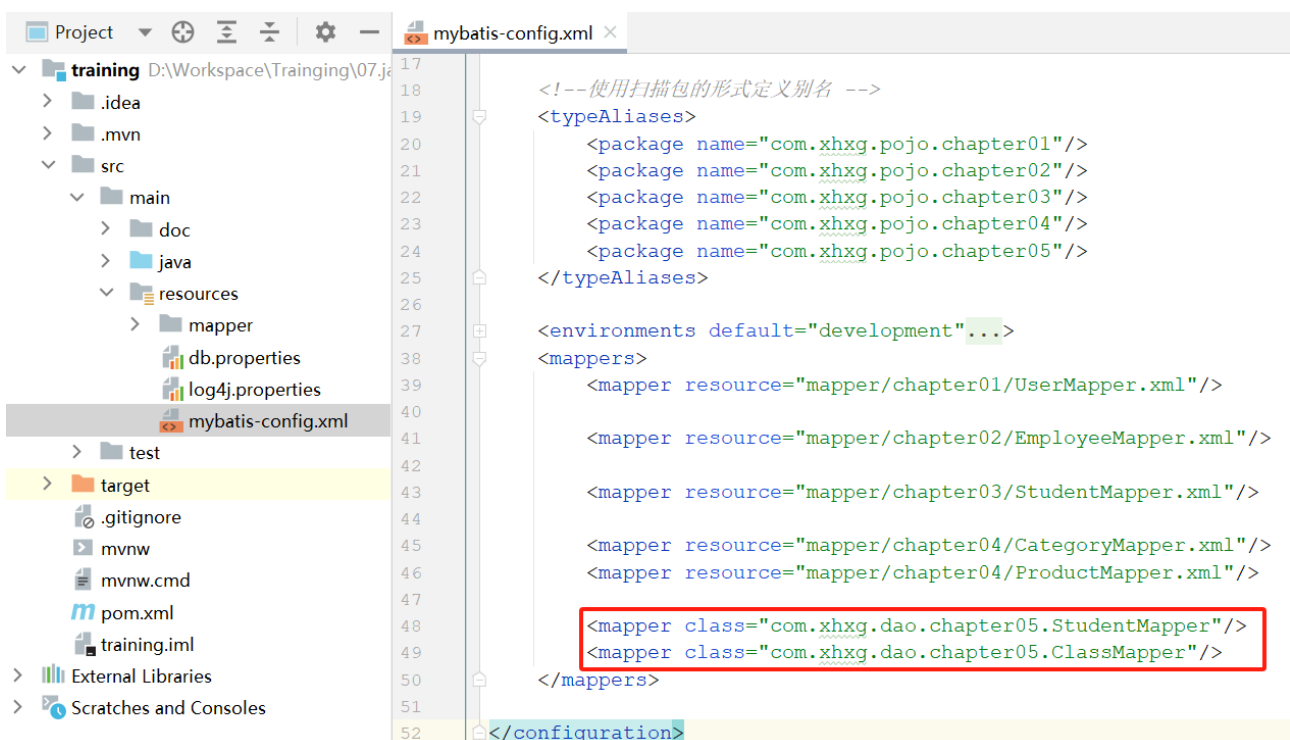
2、创建 Mapper 接口类

创建代码包为：com.xhcg.dao.chapter05，并且创建 ClassMapper 接口类和 StudentMapper 接口类，内容如下。



3、创建 mybatis-config.xml 配置文件

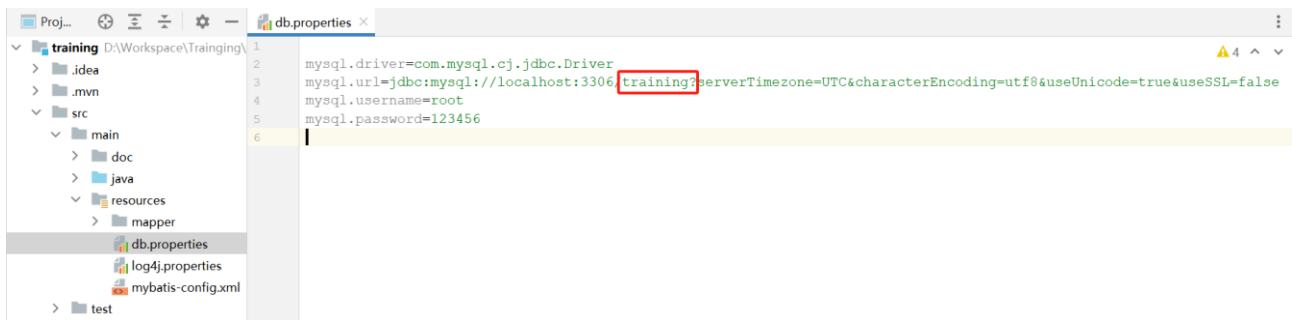
在 src/main/resources 目录下创建一个 mybatis-config.xml 文件，内容如下：



里面的配置项，除了数据库的连接信息外，还需要将前面的创建的 StudentMapper 和 ClassMapper 接口类引入，另外还需要创建另一个配置文件 db.properties

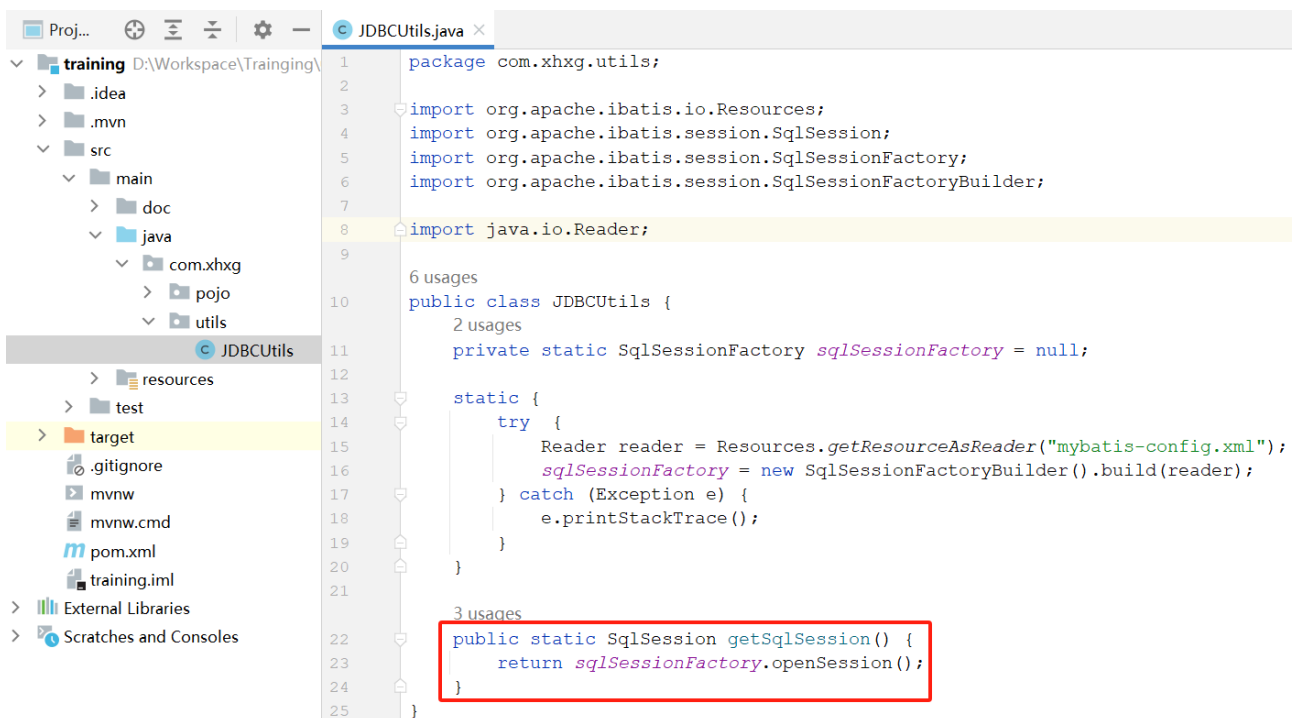
4、创建 db.properties 配置文件

在 src/main/resources 目录下创建一个 **db.properties** 文件，内容如下：



5、创建一个 Mybatis 连接工具类 JDBCUtils

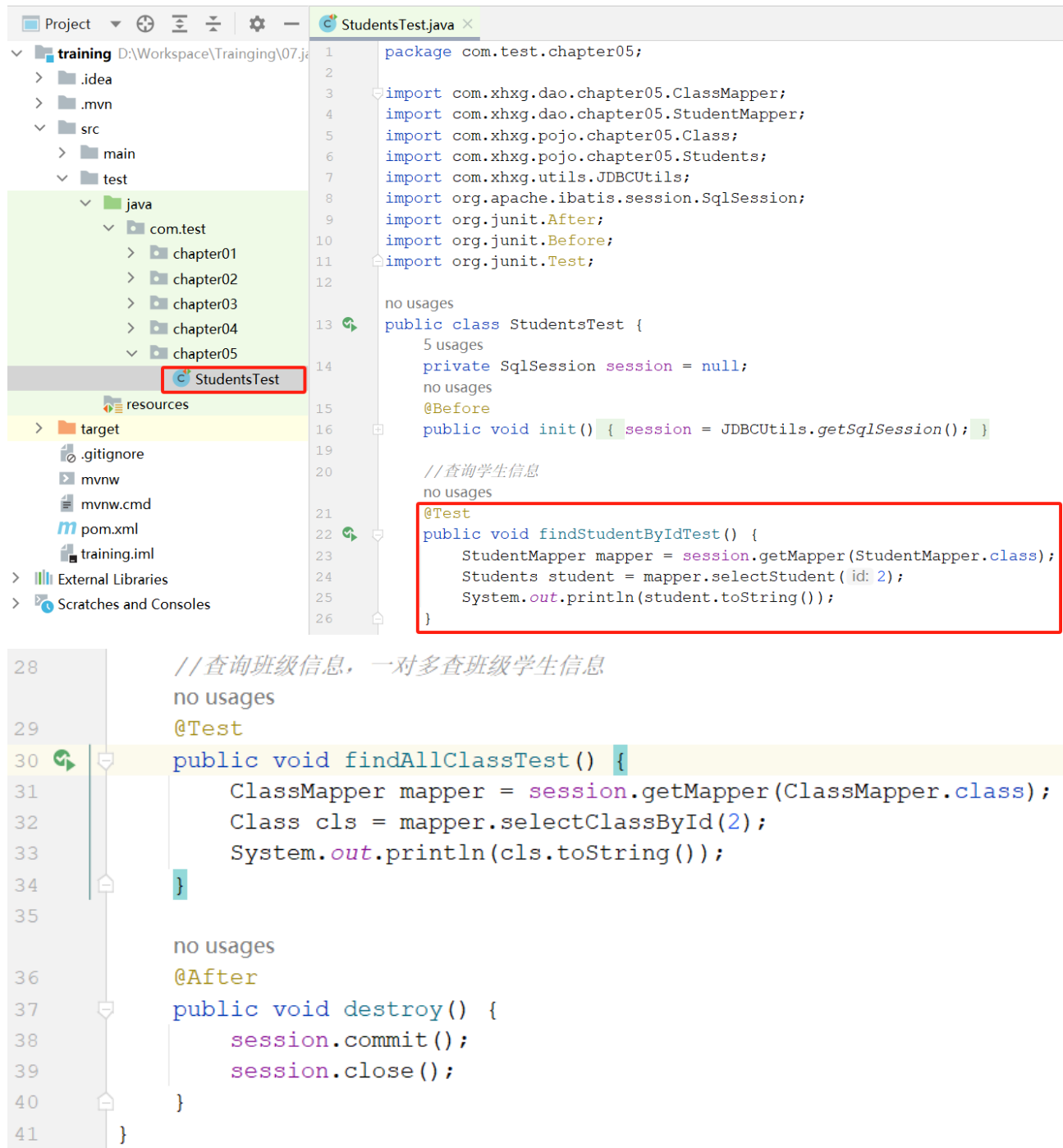
由于执行测试时，每个方法执行都需要读取配置文件，并根据配置文件的信息构建 `SqlSessionFactory` 对象、创建 `SqlSession` 对象、释放资源，这导致了大量的重复代码。为了简化开发，我们可以将读取配置文件和释放资源的代码封装到一个工具类 `JDBCUtils` 中（需要创建一个 `utils` 的 package 包，类路径：`com.xhxcg.utils.JDBCUtils`），然后通过工具类（`JDBCUtils`）创建 `SqlSession` 对象。代码如下：



6、编写测试类

以上代码和配置创建好后，就可以编写测试类进行验证测试。

我们在 src/test/java 目录下，建议测试包，路径为：com.test.chapter04，然后创建测类 StudentsTest，测试的方法如下：



```
1 package com.test.chapter05;
2
3 import com.xhxc.dao.chapter05.ClassMapper;
4 import com.xhxc.dao.chapter05.StudentMapper;
5 import com.xhxc.pojo.chapter05.Class;
6 import com.xhxc.pojo.chapter05.Students;
7 import com.xhxc.utils.JDBCUtils;
8 import org.apache.ibatis.session.SqlSession;
9 import org.junit.After;
10 import org.junit.Before;
11 import org.junit.Test;
12
13 no usages
14 public class StudentsTest {
15     5 usages
16     private SqlSession session = null;
17     no usages
18     @Before
19     public void init() { session = JDBCUtils.getSqlSession(); }
20
21     // 查询学生信息
22     no usages
23     @Test
24     public void findStudentByIdTest() {
25         StudentMapper mapper = session.getMapper(StudentMapper.class);
26         Students student = mapper.selectStudent(id: 2);
27         System.out.println(student.toString());
28     }
29
30     // 查询班级信息，一对多查班级学生信息
31     no usages
32     @Test
33     public void findAllClassTest() {
34         ClassMapper mapper = session.getMapper(ClassMapper.class);
35         Class cls = mapper.selectClassById(2);
36         System.out.println(cls.toString());
37     }
38
39     no usages
40     @After
41     public void destroy() {
42         session.commit();
43         session.close();
44     }
45 }
```

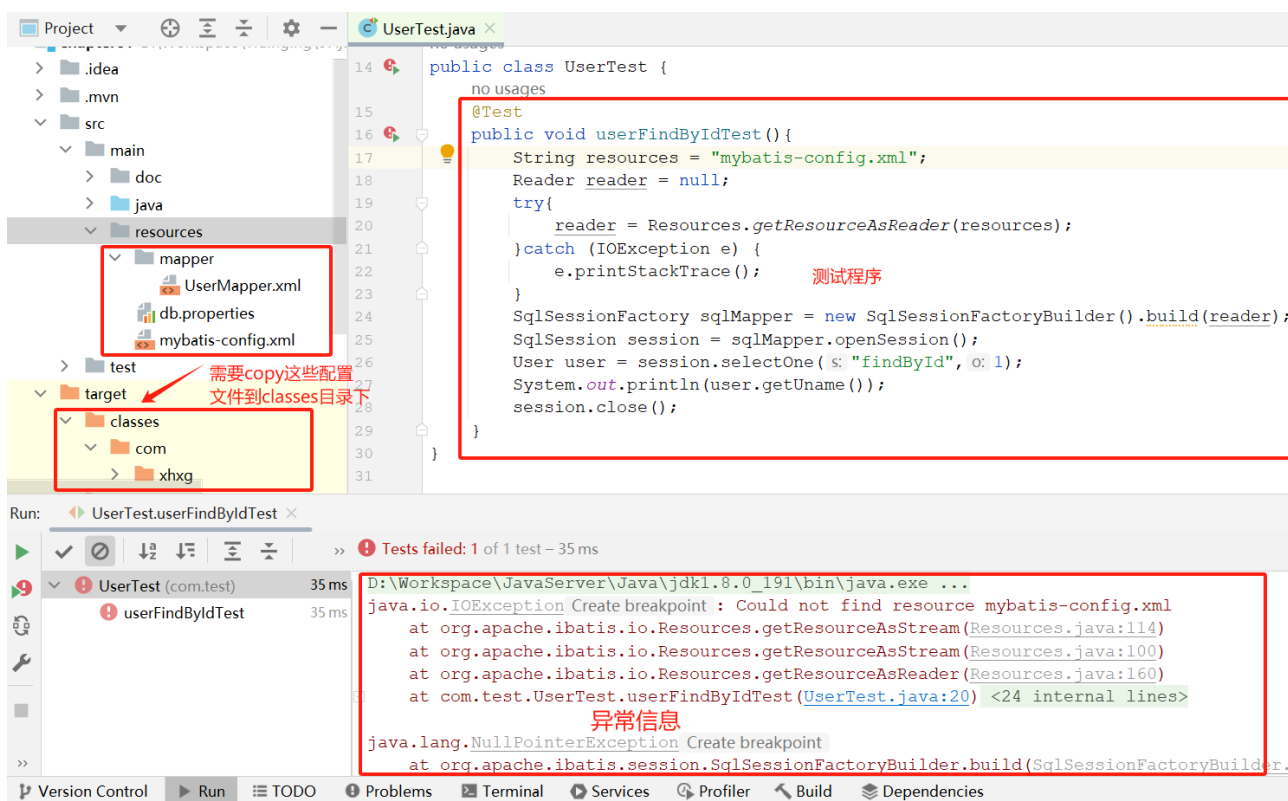
在不同的单元测试方法中，执行不同的单元测试，并输出相应的结果。

至此，本实验圆满成功，建立了 Java 代码与数据库的关联，验证了通过 Java 调用，正确获取数据库中的记录。

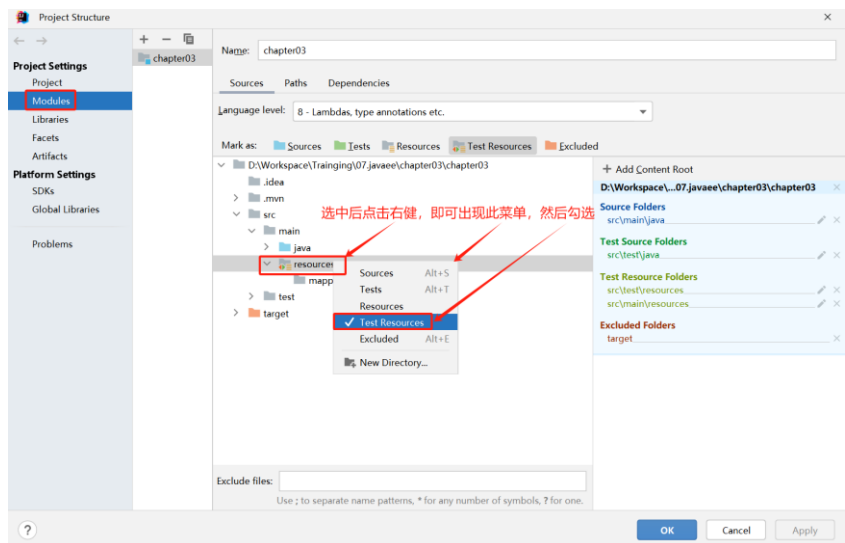
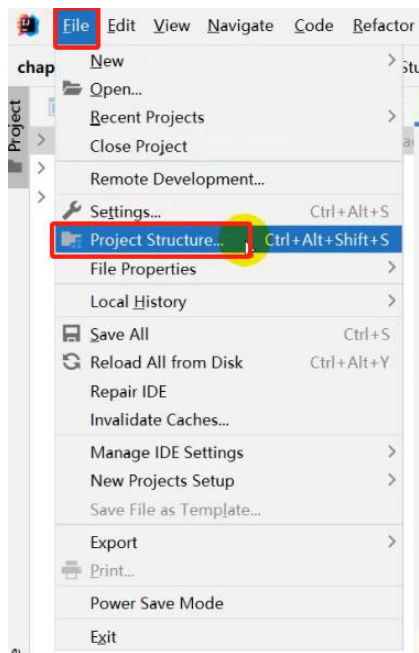
三、常见问题

有时候 Maven 不能正确的复制配置文件到 target 目录中, 此时运行测试程序就会报如下异常: Could not find resource mybatis-config.xml

解决办法 1, 就是手动将 src/main/resources/ 目录下的所有配置文件 (包含目录) 复制到 target/classes/ 目录下即可解决。(初学者建议采用此方法, 增加你对项目工程的熟悉度)



解决办法 2, 让 IDEA 自动编译复制, 步骤如下截图。配置好后, IDEA 编译时会自动复制 resource 目录中配置文件到编译后的目录, 就不需要再手工复制了。



注：若 Test Resources 不生效，可以换成勾选 Resources