JavaEE 架构及开发项目 SSM 学生管理系统 大学编程作业（TUST 天津科技大学 2023 年)

一、项目简介

本学生管理系统，我使用了 SSM（Spring+SpringMVC+MyBatis）架构来制作，使用了 Java 编程语言进行 Web 应用程序的开发，使用了 MySQL 数据库和 Maven 构建工具。实现了登

录、课程添加、课程删除、成绩查询、课程属性修改、成绩管理、已授课程成绩查询、培养方案管理、成绩管理、信息查询和成绩排名等功能，并且 UI 界面较为美观易用。通过这次 JavaEE 架构及开发项目的实践，我巩固了 Spring、SpringMVC 和 MyBatis 的知识，熟练应用了各个第三方开源库，为之后的 Java 后端开发的深入学习打下了基础。

这个项目是我大三写的，现在回顾已经非常粗糙，分享出来一方面希望可以帮助初学者，另一方面希望能让同学们可以从目前大学中普遍毫无价值的形式主义作业中解脱出来，更加高效地学习优质计算机知识和主流编程技术，一起发扬开源精神，感受互联网技术的美好愿景。

二、项目要求

请使用学过的 SSM 知识，实现一个小型学生管理系统，要求具有以下功能：

学生端：

1. 登录、登出、注册。
2. 选课。
3. 成绩查询。
4. 课表查询。

管理端：

1. 课程信息录入。
2. 学生信息修改。
3. 课程成绩录入。
4. 课程成绩统计（平均、方差、最高、最低、各个分数段人数）。

三、交流学习

互联网开源精神需要大家一起互相交流学习，互相支持奉献。欢迎大家与我友好交流。

加我 QQ 好友获取所有项目源码和项目文档，感谢大家的支持！

学生管理系统程序说明文档

1. 程序说明文档

1.1 系统目标

本学生管理系统旨在提供一种简单易用的方式来管理学生信息，让学生、教师和教务处能够快速进行课程管理和成绩管理，同时支持一些基本的扩展功能。应用还需要提供一个良好的用户界面，方便用户进行操作。

1.2 系统设计

1.2.1 系统架构

本学生管理系统采用了 SSM（Spring+SpringMVC+MyBatis）架构，将应用的业务逻辑、用户界面和数据管理分别封装在不同的组件中，使得应用更加易于维护和扩展。

- Model - 数据模型层：管理应用的数据，包括学生的学号、姓名、身份证号、手机号、入学年份、地址、专业、班级等信息。使用 MySql 数据库进行存储管理。
- View - 视图层：负责应用的用户界面设计和显示，包括登录界面、学生主界面、教师主界面、教务处主界面等。
- Controller - 控制器层：负责管理用户与应用之间的交互，接收用户的输入并处理对应的业务逻辑，包括登录验证、课程管理、成绩管理等。

1.2.2 功能实现

本学生管理系统主要实现以下功能：

1. 登录界面：用户可以输入用户名和密码进行登录，只有登录成功后才能进入主界面进行操作。
2. 学生主界面：学生可以通过该界面进行课程添加、删除和成绩查询等操作。
3. 教师主界面：教师可以通过该界面进行课程属性修改、成绩管理和已授课程成绩查询等操作。

4. 教务处主界面：教务处可以通过该界面进行培养方案管理、成绩管理、信息查询和成绩排名等操作。

1.2.3 系统实现

本学生管理系统使用了 IntelliJ IDEA 作为开发环境，采用了 Java 编程语言进行开发。应用的 UI 设计使用了 JSP 页面模板引擎和 Bootstrap 前端框架，使得应用的界面更加美观和易于维护。

在应用的设计和实现过程中，遵循了 SSM 的最佳实践，如尽量避免使用硬编码、将字符串和资源文件分离、使用多线程进行耗时操作等。同时，还使用了 MyBatis 数据库访问框架等相关技术。

1.3. 系统部署

1.3.1 建立数据库

首先，需要在 MySQL 中创建一个新的数据库。可以使用 MySQL Workbench 工具来创建数据库。具体步骤如下：

1. 打开 MySQL Workbench 工具。
2. 在管理器中，右键单击“数据导航器”，然后选择“新建连接”。
3. 在“新建连接”对话框中，输入连接名称（例如 MYXH），然后点击“测试连接”按钮，测试连接是否成功。
4. 在“新建连接”对话框中，点击“确定”按钮，完成连接的建立。
5. 在管理器中，右键单击“SCHEMAS”节点，然后选择“新建 SCHEMA”。
6. 在“新建 SCHEMA”对话框中，输入 SCHEMA 名称（例如 student_management_system），然后点击“应用”按钮。
7. 在“新建 SCHEMA”对话框中，点击“OK”按钮，完成新建 SCHEMA 的操作。

1.3.2 导入数据库表

导入数据库表需要执行以下步骤：

1. 将项目中的 student_management_system.sql 文件导入到 MySQL 数据库中，创建相应的数据表。

2. 修改 dbconfig.properties 文件中的数据库连接信息，包括用户名、密码、数据库名称、端口号等，以便应用能够连接到正确的数据库。

```
jdbc.username=MYXH
jdbc.password=520.ILY!
jdbc.driver=com.mysql.cj.jdbc.Driver
jdbc.url=jdbc:mysql:///student_management_system
```

1.3.2 登录账号

在应用部署完成后，在 Web 浏览器中访问 http://localhost:8080/StudentManagementSystem/index.jsp ，进入学生管理系统的登录界面，可以使用以下账号进行登录：

- 用户名：MYXH
  密码：520.ILY!
- 用户名：teacher
  密码：123456
- 用户名：admin
  密码：000000

可以使用以上账号在登录界面进行登录，成功后即可进入主界面进行操作。如果需要添加新的账号，可以在数据库中的用户表中进行添加。

2. 系统相关代码

以下是本学生管理系统的几个主要相关代码：

2.1 bean 包

2.1.1 ClassInfo.java

```java
package com.myxh.studentmanagementsystem.bean;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public class ClassInfo
{
    private String className;
    private Integer studentNum;
    private String monitor;
    private String studyCommittee;
    private String lifeCommittee;

    public String getClassName()
    {
        return className;
    }

    public void setClassName(String className)
    {
        this.className = className == null ? null : className.trim();
    }

    public Integer getStudentNum()
    {
        return studentNum;
    }

    public void setStudentNum(Integer studentNum)
    {
        this.studentNum = studentNum;
    }

    public String getMonitor()
    {
        return monitor;
    }

    public void setMonitor(String monitor)
    {
        this.monitor = monitor == null ? null : monitor.trim();
    }
```

```
    public String getStudyCommittee()
    {
        return studyCommittee;
    }

    public void setStudyCommittee(String studyCommittee)
    {
        this.studyCommittee = studyCommittee == null ? null : studyCommittee.trim();
    }

    public String getLifeCommittee()
    {
        return lifeCommittee;
    }

    public void setLifeCommittee(String lifeCommittee)
    {
        this.lifeCommittee = lifeCommittee == null ? null : lifeCommittee.trim();
    }
}
```

2.1.2 ClassInfoExample.java

```java
package com.myxh.studentmanagementsystem.bean;

import java.util.ArrayList;
import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public class ClassInfoExample
{
    protected String orderByClause;
    protected boolean distinct;
    protected List<Criteria> oredCriteria;

    public ClassInfoExample()
    {
        oredCriteria = new ArrayList<>();
    }

    public void setOrderByClause(String orderByClause)
    {
        this.orderByClause = orderByClause;
    }

    public String getOrderByClause()
    {
        return orderByClause;
    }

    public void setDistinct(boolean distinct)
    {
        this.distinct = distinct;
    }

    public boolean isDistinct()
    {
        return distinct;
    }

    public List<Criteria> getOredCriteria()
    {
        return oredCriteria;
    }
```

```
    public void or(Criteria criteria)
    {
        oredCriteria.add(criteria);
    }

    public Criteria or()
    {
        Criteria criteria = createCriteriaInternal();
        oredCriteria.add(criteria);

        return criteria;
    }

    public Criteria createCriteria()
    {
        Criteria criteria = createCriteriaInternal();

        if (oredCriteria.size() == 0)
        {
            oredCriteria.add(criteria);
        }

        return criteria;
    }

    protected Criteria createCriteriaInternal()
    {
        return new Criteria();
    }

    public void clear()
    {
        oredCriteria.clear();
        orderByClause = null;
        distinct = false;
    }

    protected abstract static class GeneratedCriteria
    {
        protected List<Criterion> criteria;

        protected GeneratedCriteria()
        {
```

```
        super();
        criteria = new ArrayList<>();
    }

    public boolean isValid()
    {
        return criteria.size() > 0;
    }

    public List<Criterion> getAllCriteria()
    {
        return criteria;
    }

    public List<Criterion> getCriteria()
    {
        return criteria;
    }

    protected void addCriterion(String condition)
    {
        if (condition == null)
        {
            throw new RuntimeException("Value for condition cannot be null");
        }

        criteria.add(new Criterion(condition));
    }

    protected void addCriterion(String condition, Object value, String property)
    {
        if (value == null)
        {
            throw new RuntimeException("Value for " + property + " cannot be null");
        }

        criteria.add(new Criterion(condition, value));
    }

    protected void addCriterion(String condition, Object value1, Object value2, String propert
    {
        if (value1 == null || value2 == null)
        {
            throw new RuntimeException("Between values for " + property + " cannot be null");
```

```
            }

            criteria.add(new Criterion(condition, value1, value2));
        }

        public Criteria andClassNameIsNull()
        {
            addCriterion("class_name is null");

            return (Criteria) this;
        }

        public Criteria andClassNameIsNotNull()
        {
            addCriterion("class_name is not null");

            return (Criteria) this;
        }

        public Criteria andClassNameEqualTo(String value)
        {
            addCriterion("class_name =", value, "className");

            return (Criteria) this;
        }

        public Criteria andClassNameNotEqualTo(String value)
        {
            addCriterion("class_name <>", value, "className");

            return (Criteria) this;
        }

        public Criteria andClassNameGreaterThan(String value)
        {
            addCriterion("class_name >", value, "className");

            return (Criteria) this;
        }

        public Criteria andClassNameGreaterThanOrEqualTo(String value)
        {
            addCriterion("class_name >=", value, "className");
```

```
            return (Criteria) this;
        }

        public Criteria andClassNameLessThan(String value)
        {
            addCriterion("class_name <", value, "className");

            return (Criteria) this;
        }

        public Criteria andClassNameLessThanOrEqualTo(String value)
        {
            addCriterion("class_name <=", value, "className");

            return (Criteria) this;
        }

        public Criteria andClassNameLike(String value)
        {
            addCriterion("class_name like", value, "className");

            return (Criteria) this;
        }

        public Criteria andClassNameNotLike(String value)
        {
            addCriterion("class_name not like", value, "className");

            return (Criteria) this;
        }

        public Criteria andClassNameIn(List<String> values)
        {
            addCriterion("class_name in", values, "className");

            return (Criteria) this;
        }

        public Criteria andClassNameNotIn(List<String> values)
        {
            addCriterion("class_name not in", values, "className");

            return (Criteria) this;
        }
```

```
        public Criteria andClassNameBetween(String value1, String value2)
        {
            addCriterion("class_name between", value1, value2, "className");

            return (Criteria) this;
        }

        public Criteria andClassNameNotBetween(String value1, String value2)
        {
            addCriterion("class_name not between", value1, value2, "className");

            return (Criteria) this;
        }

        public Criteria andStudentNumIsNull()
        {
            addCriterion("student_num is null");

            return (Criteria) this;
        }

        public Criteria andStudentNumIsNotNull()
        {
            addCriterion("student_num is not null");

            return (Criteria) this;
        }

        public Criteria andStudentNumEqualTo(Integer value)
        {
            addCriterion("student_num =", value, "studentNum");

            return (Criteria) this;
        }

        public Criteria andStudentNumNotEqualTo(Integer value)
        {
            addCriterion("student_num <>", value, "studentNum");

            return (Criteria) this;
        }

        public Criteria andStudentNumGreaterThan(Integer value)
```

```
        {
            addCriterion("student_num >", value, "studentNum");

            return (Criteria) this;
        }

        public Criteria andStudentNumGreaterThanOrEqualTo(Integer value)
        {
            addCriterion("student_num >=", value, "studentNum");

            return (Criteria) this;
        }

        public Criteria andStudentNumLessThan(Integer value)
        {
            addCriterion("student_num <", value, "studentNum");

            return (Criteria) this;
        }

        public Criteria andStudentNumLessThanOrEqualTo(Integer value)
        {
            addCriterion("student_num <=", value, "studentNum");

            return (Criteria) this;
        }

        public Criteria andStudentNumIn(List<Integer> values)
        {
            addCriterion("student_num in", values, "studentNum");

            return (Criteria) this;
        }

        public Criteria andStudentNumNotIn(List<Integer> values)
        {
            addCriterion("student_num not in", values, "studentNum");

            return (Criteria) this;
        }

        public Criteria andStudentNumBetween(Integer value1, Integer value2)
        {
            addCriterion("student_num between", value1, value2, "studentNum");
```

```
            return (Criteria) this;
        }

        public Criteria andStudentNumNotBetween(Integer value1, Integer value2)
        {
            addCriterion("student_num not between", value1, value2, "studentNum");

            return (Criteria) this;
        }

        public Criteria andMonitorIsNull()
        {
            addCriterion("monitor is null");

            return (Criteria) this;
        }

        public Criteria andMonitorIsNotNull()
        {
            addCriterion("monitor is not null");

            return (Criteria) this;
        }

        public Criteria andMonitorEqualTo(String value)
        {
            addCriterion("monitor =", value, "monitor");

            return (Criteria) this;
        }

        public Criteria andMonitorNotEqualTo(String value)
        {
            addCriterion("monitor <>", value, "monitor");

            return (Criteria) this;
        }

        public Criteria andMonitorGreaterThan(String value)
        {
            addCriterion("monitor >", value, "monitor");

            return (Criteria) this;
```

```
        }

        public Criteria andMonitorGreaterThanOrEqualTo(String value)
        {
            addCriterion("monitor >=", value, "monitor");

            return (Criteria) this;
        }

        public Criteria andMonitorLessThan(String value)
        {
            addCriterion("monitor <", value, "monitor");

            return (Criteria) this;
        }

        public Criteria andMonitorLessThanOrEqualTo(String value)
        {
            addCriterion("monitor <=", value, "monitor");

            return (Criteria) this;
        }

        public Criteria andMonitorLike(String value)
        {
            addCriterion("monitor like", value, "monitor");

            return (Criteria) this;
        }

        public Criteria andMonitorNotLike(String value)
        {
            addCriterion("monitor not like", value, "monitor");

            return (Criteria) this;
        }

        public Criteria andMonitorIn(List<String> values)
        {
            addCriterion("monitor in", values, "monitor");

            return (Criteria) this;
        }
```

```
        public Criteria andMonitorNotIn(List<String> values)
        {
            addCriterion("monitor not in", values, "monitor");

            return (Criteria) this;
        }

        public Criteria andMonitorBetween(String value1, String value2)
        {
            addCriterion("monitor between", value1, value2, "monitor");

            return (Criteria) this;
        }

        public Criteria andMonitorNotBetween(String value1, String value2)
        {
            addCriterion("monitor not between", value1, value2, "monitor");

            return (Criteria) this;
        }

        public Criteria andStudyCommitteeIsNull()
        {
            addCriterion("study_committee is null");

            return (Criteria) this;
        }

        public Criteria andStudyCommitteeIsNotNull()
        {
            addCriterion("study_committee is not null");

            return (Criteria) this;
        }

        public Criteria andStudyCommitteeEqualTo(String value)
        {
            addCriterion("study_committee =", value, "studyCommittee");

            return (Criteria) this;
        }

        public Criteria andStudyCommitteeNotEqualTo(String value)
        {
```

```
            addCriterion("study_committee <>", value, "studyCommittee");

            return (Criteria) this;
        }

        public Criteria andStudyCommitteeGreaterThan(String value)
        {
            addCriterion("study_committee >", value, "studyCommittee");

            return (Criteria) this;
        }

        public Criteria andStudyCommitteeGreaterThanOrEqualTo(String value)
        {
            addCriterion("study_committee >=", value, "studyCommittee");

            return (Criteria) this;
        }

        public Criteria andStudyCommitteeLessThan(String value)
        {
            addCriterion("study_committee <", value, "studyCommittee");

            return (Criteria) this;
        }

        public Criteria andStudyCommitteeLessThanOrEqualTo(String value)
        {
            addCriterion("study_committee <=", value, "studyCommittee");

            return (Criteria) this;
        }

        public Criteria andStudyCommitteeLike(String value)
        {
            addCriterion("study_committee like", value, "studyCommittee");

            return (Criteria) this;
        }

        public Criteria andStudyCommitteeNotLike(String value)
        {
            addCriterion("study_committee not like", value, "studyCommittee");
```

```
            return (Criteria) this;
        }

        public Criteria andStudyCommitteeIn(List<String> values)
        {
            addCriterion("study_committee in", values, "studyCommittee");

            return (Criteria) this;
        }

        public Criteria andStudyCommitteeNotIn(List<String> values)
        {
            addCriterion("study_committee not in", values, "studyCommittee");

            return (Criteria) this;
        }

        public Criteria andStudyCommitteeBetween(String value1, String value2)
        {
            addCriterion("study_committee between", value1, value2, "studyCommittee");

            return (Criteria) this;
        }

        public Criteria andStudyCommitteeNotBetween(String value1, String value2)
        {
            addCriterion("study_committee not between", value1, value2, "studyCommittee");

            return (Criteria) this;
        }

        public Criteria andLifeCommitteeIsNull()
        {
            addCriterion("life_committee is null");

            return (Criteria) this;
        }

        public Criteria andLifeCommitteeIsNotNull()
        {
            addCriterion("life_committee is not null");

            return (Criteria) this;
        }
```

```
        public Criteria andLifeCommitteeEqualTo(String value)
        {
            addCriterion("life_committee =", value, "lifeCommittee");

            return (Criteria) this;
        }

        public Criteria andLifeCommitteeNotEqualTo(String value)
        {
            addCriterion("life_committee <>", value, "lifeCommittee");

            return (Criteria) this;
        }

        public Criteria andLifeCommitteeGreaterThan(String value)
        {
            addCriterion("life_committee >", value, "lifeCommittee");

            return (Criteria) this;
        }

        public Criteria andLifeCommitteeGreaterThanOrEqualTo(String value)
        {
            addCriterion("life_committee >=", value, "lifeCommittee");

            return (Criteria) this;
        }

        public Criteria andLifeCommitteeLessThan(String value)
        {
            addCriterion("life_committee <", value, "lifeCommittee");

            return (Criteria) this;
        }

        public Criteria andLifeCommitteeLessThanOrEqualTo(String value)
        {
            addCriterion("life_committee <=", value, "lifeCommittee");

            return (Criteria) this;
        }

        public Criteria andLifeCommitteeLike(String value)
```

```
        {
            addCriterion("life_committee like", value, "lifeCommittee");

            return (Criteria) this;
        }

        public Criteria andLifeCommitteeNotLike(String value)
        {
            addCriterion("life_committee not like", value, "lifeCommittee");

            return (Criteria) this;
        }

        public Criteria andLifeCommitteeIn(List<String> values)
        {
            addCriterion("life_committee in", values, "lifeCommittee");

            return (Criteria) this;
        }

        public Criteria andLifeCommitteeNotIn(List<String> values)
        {
            addCriterion("life_committee not in", values, "lifeCommittee");

            return (Criteria) this;
        }

        public Criteria andLifeCommitteeBetween(String value1, String value2)
        {
            addCriterion("life_committee between", value1, value2, "lifeCommittee");

            return (Criteria) this;
        }

        public Criteria andLifeCommitteeNotBetween(String value1, String value2)
        {
            addCriterion("life_committee not between", value1, value2, "lifeCommittee");

            return (Criteria) this;
        }
    }

    public static class Criteria extends GeneratedCriteria
    {
```

```
        protected Criteria()
        {
            super();
        }
    }

    public static class Criterion
    {
        private final String condition;
        private Object value;
        private Object secondValue;
        private boolean noValue;
        private boolean singleValue;
        private boolean betweenValue;
        private boolean listValue;
        private final String typeHandler;

        public String getCondition()
        {
            return condition;
        }

        public Object getValue()
        {
            return value;
        }

        public Object getSecondValue()
        {
            return secondValue;
        }

        public boolean isNoValue()
        {
            return noValue;
        }

        public boolean isSingleValue()
        {
            return singleValue;
        }

        public boolean isBetweenValue()
        {
```

```
            return betweenValue;
        }

        public boolean isListValue()
        {
            return listValue;
        }

        public String getTypeHandler()
        {
            return typeHandler;
        }

        protected Criterion(String condition)
        {
            super();
            this.condition = condition;
            this.typeHandler = null;
            this.noValue = true;
        }

        protected Criterion(String condition, Object value, String typeHandler)
        {
            super();
            this.condition = condition;
            this.value = value;
            this.typeHandler = typeHandler;

            if (value instanceof List<?>)
            {
                this.listValue = true;
            }
            else
            {
                this.singleValue = true;
            }
        }

        protected Criterion(String condition, Object value)
        {
            this(condition, value, null);
        }

        protected Criterion(String condition, Object value, Object secondValue, String typeHandler
```

```
        {
            super();
            this.condition = condition;
            this.value = value;
            this.secondValue = secondValue;
            this.typeHandler = typeHandler;
            this.betweenValue = true;
        }

        protected Criterion(String condition, Object value, Object secondValue)
        {
            this(condition, value, secondValue, null);
        }
    }
}
```

2.1.3 Msg.java

```java
package com.myxh.studentmanagementsystem.bean;

import java.util.HashMap;
import java.util.Map;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public class Msg
{
    private Integer code;
    private String msg;
    private Map<String, Object> extend = new HashMap<>();

    public static Msg success()
    {
        Msg msg = new Msg();

        msg.setCode(100);
        msg.setMsg("处理成功! ");

        return msg;
    }

    public static Msg fail()
    {
        Msg msg = new Msg();

        msg.setCode(200);
        msg.setMsg("处理失败! ");

        return msg;
    }

    public Msg add(String key, Object value)
    {
        this.getExtend().put(key, value);

        return this;
    }

    public Integer getCode()
    {
```

```
        return code;
    }

    public void setCode(Integer code)
    {
        this.code = code;
    }

    public String getMsg()
    {
        return msg;
    }

    public void setMsg(String msg)
    {
        this.msg = msg;
    }

    public Map<String, Object> getExtend()
    {
        return extend;
    }

    public void setExtend(Map<String, Object> extend)
    {
        this.extend = extend;
    }
}
```

2.1.4 StuCurriculum.java

```java
package com.myxh.studentmanagementsystem.bean;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public class StuCurriculum
{
    private Integer id;
    private Integer courseId;
    private String courseName;
    private String courseNature;
    private String courseSelectedId;
    private String courseSelectedStu;
    private String courseStatus;
    private Integer grade;

    public StuCurriculum()
    {

    }

    public StuCurriculum(Integer id, Integer courseId, String courseName, String courseNature, Str
    {
        this.id = id;
        this.courseId = courseId;
        this.courseName = courseName;
        this.courseNature = courseNature;
        this.courseSelectedId = courseSelectedId;
        this.courseSelectedStu = courseSelectedStu;
        this.courseStatus = courseStatus;
        this.grade = grade;
    }

    public Integer getId()
    {
        return id;
    }

    public void setId(Integer id)
    {
        this.id = id;
    }
```

```
    public Integer getCourseId()
    {
        return courseId;
    }

    public void setCourseId(Integer courseId)
    {
        this.courseId = courseId;
    }

    public String getCourseName()
    {
        return courseName;
    }

    public void setCourseName(String courseName)
    {
        this.courseName = courseName == null ? null : courseName.trim();
    }

    public String getCourseNature()
    {
        return courseNature;
    }

    public void setCourseNature(String courseNature)
    {
        this.courseNature = courseNature == null ? null : courseNature.trim();
    }

    public String getCourseSelectedId()
    {
        return courseSelectedId;
    }

    public void setCourseSelectedId(String courseSelectedId)
    {
        this.courseSelectedId = courseSelectedId == null ? null : courseSelectedId.trim();
    }

    public String getCourseSelectedStu()
    {
        return courseSelectedStu;
    }
```

```
    public void setCourseSelectedStu(String courseSelectedStu)
    {
        this.courseSelectedStu = courseSelectedStu == null ? null : courseSelectedStu.trim();
    }

    public String getCourseStatus()
    {
        return courseStatus;
    }

    public void setCourseStatus(String courseStatus)
    {
        this.courseStatus = courseStatus == null ? null : courseStatus.trim();
    }

    public Integer getGrade()
    {
        return grade;
    }

    public void setGrade(Integer grade)
    {
        this.grade = grade;
    }

    @Override
    public String toString()
    {
        return "StuCurriculum{" +
                "id=" + id +
                ", courseId=" + courseId +
                ", courseName='" + courseName + '\'' +
                ", courseNature='" + courseNature + '\'' +
                ", courseSelectedId='" + courseSelectedId + '\'' +
                ", courseSelectedStu='" + courseSelectedStu + '\'' +
                ", courseStatus='" + courseStatus + '\'' +
                ", grade=" + grade +
                '}';
    }
}
```

2.1.5 StuCurriculumExample.java

```java
package com.myxh.studentmanagementsystem.bean;

import java.util.ArrayList;
import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public class StuCurriculumExample
{
    protected String orderByClause;
    protected boolean distinct;
    protected List<Criteria> oredCriteria;

    public StuCurriculumExample()
    {
        oredCriteria = new ArrayList<>();
    }

    public void setOrderByClause(String orderByClause)
    {
        this.orderByClause = orderByClause;
    }

    public String getOrderByClause()
    {
        return orderByClause;
    }

    public void setDistinct(boolean distinct)
    {
        this.distinct = distinct;
    }

    public boolean isDistinct()
    {
        return distinct;
    }

    public List<Criteria> getOredCriteria()
    {
        return oredCriteria;
    }
```

```
    public void or(Criteria criteria)
    {
        oredCriteria.add(criteria);
    }

    public Criteria or()
    {
        Criteria criteria = createCriteriaInternal();
        oredCriteria.add(criteria);

        return criteria;
    }

    public Criteria createCriteria()
    {
        Criteria criteria = createCriteriaInternal();

        if (oredCriteria.size() == 0)
        {
            oredCriteria.add(criteria);
        }

        return criteria;
    }

    protected Criteria createCriteriaInternal()
    {

        return new Criteria();
    }

    public void clear()
    {
        oredCriteria.clear();
        orderByClause = null;
        distinct = false;
    }

    protected abstract static class GeneratedCriteria
    {
        protected List<Criterion> criteria;

        protected GeneratedCriteria()
```

```
        {
            super();
            criteria = new ArrayList<>();
        }

        public boolean isValid()
        {
            return criteria.size() > 0;
        }

        public List<Criterion> getAllCriteria()
        {
            return criteria;
        }

        public List<Criterion> getCriteria()
        {
            return criteria;
        }

        protected void addCriterion(String condition)
        {
            if (condition == null)
            {
                throw new RuntimeException("Value for condition cannot be null");
            }

            criteria.add(new Criterion(condition));
        }

        protected void addCriterion(String condition, Object value, String property)
        {
            if (value == null)
            {
                throw new RuntimeException("Value for " + property + " cannot be null");
            }

            criteria.add(new Criterion(condition, value));
        }

        protected void addCriterion(String condition, Object value1, Object value2, String propert
        {
            if (value1 == null || value2 == null)
            {
```

```
            throw new RuntimeException("Between values for " + property + " cannot be null");
        }

        criteria.add(new Criterion(condition, value1, value2));
    }

    public Criteria andIdIsNull()
    {
        addCriterion("id is null");

        return (Criteria) this;
    }

    public Criteria andIdIsNotNull()
    {
        addCriterion("id is not null");

        return (Criteria) this;
    }

    public Criteria andIdEqualTo(Integer value)
    {
        addCriterion("id =", value, "id");

        return (Criteria) this;
    }

    public Criteria andIdNotEqualTo(Integer value)
    {
        addCriterion("id <>", value, "id");

        return (Criteria) this;
    }

    public Criteria andIdGreaterThan(Integer value)
    {
        addCriterion("id >", value, "id");

        return (Criteria) this;
    }

    public Criteria andIdGreaterThanOrEqualTo(Integer value)
    {
        addCriterion("id >=", value, "id");
```

```
            return (Criteria) this;
        }

        public Criteria andIdLessThan(Integer value)
        {
            addCriterion("id <", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdLessThanOrEqualTo(Integer value)
        {
            addCriterion("id <=", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdIn(List<Integer> values)
        {
            addCriterion("id in", values, "id");

            return (Criteria) this;
        }

        public Criteria andIdNotIn(List<Integer> values)
        {
            addCriterion("id not in", values, "id");

            return (Criteria) this;
        }

        public Criteria andIdBetween(Integer value1, Integer value2)
        {
            addCriterion("id between", value1, value2, "id");

            return (Criteria) this;
        }

        public Criteria andIdNotBetween(Integer value1, Integer value2)
        {
            addCriterion("id not between", value1, value2, "id");

            return (Criteria) this;
```

```
        }

        public Criteria andCourseIdIsNull()
        {
            addCriterion("course_id is null");

            return (Criteria) this;
        }

        public Criteria andCourseIdIsNotNull()
        {
            addCriterion("course_id is not null");

            return (Criteria) this;
        }

        public Criteria andCourseIdEqualTo(Integer value)
        {
            addCriterion("course_id =", value, "courseId");

            return (Criteria) this;
        }

        public Criteria andCourseIdNotEqualTo(Integer value)
        {
            addCriterion("course_id <>", value, "courseId");

            return (Criteria) this;
        }

        public Criteria andCourseIdGreaterThan(Integer value)
        {
            addCriterion("course_id >", value, "courseId");

            return (Criteria) this;
        }

        public Criteria andCourseIdGreaterThanOrEqualTo(Integer value)
        {
            addCriterion("course_id >=", value, "courseId");

            return (Criteria) this;
        }
```

```
        public Criteria andCourseIdLessThan(Integer value)
        {
            addCriterion("course_id <", value, "courseId");

            return (Criteria) this;
        }

        public Criteria andCourseIdLessThanOrEqualTo(Integer value)
        {
            addCriterion("course_id <=", value, "courseId");

            return (Criteria) this;
        }

        public Criteria andCourseIdIn(List<Integer> values)
        {
            addCriterion("course_id in", values, "courseId");

            return (Criteria) this;
        }

        public Criteria andCourseIdNotIn(List<Integer> values)
        {
            addCriterion("course_id not in", values, "courseId");

            return (Criteria) this;
        }

        public Criteria andCourseIdBetween(Integer value1, Integer value2)
        {
            addCriterion("course_id between", value1, value2, "courseId");

            return (Criteria) this;
        }

        public Criteria andCourseIdNotBetween(Integer value1, Integer value2)
        {
            addCriterion("course_id not between", value1, value2, "courseId");

            return (Criteria) this;
        }

        public Criteria andCourseNameIsNull()
        {
```

```
            addCriterion("course_name is null");

            return (Criteria) this;
        }

        public Criteria andCourseNameIsNotNull()
        {
            addCriterion("course_name is not null");

            return (Criteria) this;
        }

        public Criteria andCourseNameEqualTo(String value)
        {
            addCriterion("course_name =", value, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNameNotEqualTo(String value)
        {
            addCriterion("course_name <>", value, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNameGreaterThan(String value)
        {
            addCriterion("course_name >", value, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNameGreaterThanOrEqualTo(String value)
        {
            addCriterion("course_name >=", value, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNameLessThan(String value)
        {
            addCriterion("course_name <", value, "courseName");
```

```
            return (Criteria) this;
        }

        public Criteria andCourseNameLessThanOrEqualTo(String value)
        {
            addCriterion("course_name <=", value, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNameLike(String value)
        {
            addCriterion("course_name like", value, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNameNotLike(String value)
        {
            addCriterion("course_name not like", value, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNameIn(List<String> values)
        {
            addCriterion("course_name in", values, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNameNotIn(List<String> values)
        {
            addCriterion("course_name not in", values, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNameBetween(String value1, String value2)
        {
            addCriterion("course_name between", value1, value2, "courseName");

            return (Criteria) this;
        }
```

```
        public Criteria andCourseNameNotBetween(String value1, String value2)
        {
            addCriterion("course_name not between", value1, value2, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNatureIsNull()
        {
            addCriterion("course_nature is null");

            return (Criteria) this;
        }

        public Criteria andCourseNatureIsNotNull()
        {
            addCriterion("course_nature is not null");

            return (Criteria) this;
        }

        public Criteria andCourseNatureEqualTo(String value)
        {
            addCriterion("course_nature =", value, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureNotEqualTo(String value)
        {
            addCriterion("course_nature <>", value, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureGreaterThan(String value)
        {
            addCriterion("course_nature >", value, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureGreaterThanOrEqualTo(String value)
```

```
        {
            addCriterion("course_nature >=", value, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureLessThan(String value)
        {
            addCriterion("course_nature <", value, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureLessThanOrEqualTo(String value)
        {
            addCriterion("course_nature <=", value, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureLike(String value)
        {
            addCriterion("course_nature like", value, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureNotLike(String value)
        {
            addCriterion("course_nature not like", value, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureIn(List<String> values)
        {
            addCriterion("course_nature in", values, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureNotIn(List<String> values)
        {
            addCriterion("course_nature not in", values, "courseNature");
```

```
            return (Criteria) this;
        }

        public Criteria andCourseNatureBetween(String value1, String value2)
        {
            addCriterion("course_nature between", value1, value2, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureNotBetween(String value1, String value2)
        {
            addCriterion("course_nature not between", value1, value2, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedIdIsNull()
        {
            addCriterion("course_selected_ID is null");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedIdIsNotNull()
        {
            addCriterion("course_selected_ID is not null");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedIdEqualTo(String value)
        {
            addCriterion("course_selected_ID =", value, "courseSelectedId");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedIdNotEqualTo(String value)
        {
            addCriterion("course_selected_ID <>", value, "courseSelectedId");

            return (Criteria) this;
```

```
        }

        public Criteria andCourseSelectedIdGreaterThan(String value)
        {
            addCriterion("course_selected_ID >", value, "courseSelectedId");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedIdGreaterThanOrEqualTo(String value)
        {
            addCriterion("course_selected_ID >=", value, "courseSelectedId");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedIdLessThan(String value)
        {
            addCriterion("course_selected_ID <", value, "courseSelectedId");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedIdLessThanOrEqualTo(String value)
        {
            addCriterion("course_selected_ID <=", value, "courseSelectedId");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedIdLike(String value)
        {
            addCriterion("course_selected_ID like", value, "courseSelectedId");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedIdNotLike(String value)
        {
            addCriterion("course_selected_ID not like", value, "courseSelectedId");

            return (Criteria) this;
        }
```

```
        public Criteria andCourseSelectedIdIn(List<String> values)
        {
            addCriterion("course_selected_ID in", values, "courseSelectedId");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedIdNotIn(List<String> values)
        {
            addCriterion("course_selected_ID not in", values, "courseSelectedId");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedIdBetween(String value1, String value2)
        {
            addCriterion("course_selected_ID between", value1, value2, "courseSelectedId");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedIdNotBetween(String value1, String value2)
        {
            addCriterion("course_selected_ID not between", value1, value2, "courseSelectedId");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedStuIsNull()
        {
            addCriterion("course_selected_stu is null");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedStuIsNotNull()
        {
            addCriterion("course_selected_stu is not null");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedStuEqualTo(String value)
        {
```

```
            addCriterion("course_selected_stu =", value, "courseSelectedStu");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedStuNotEqualTo(String value)
        {
            addCriterion("course_selected_stu <>", value, "courseSelectedStu");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedStuGreaterThan(String value)
        {
            addCriterion("course_selected_stu >", value, "courseSelectedStu");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedStuGreaterThanOrEqualTo(String value)
        {
            addCriterion("course_selected_stu >=", value, "courseSelectedStu");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedStuLessThan(String value)
        {
            addCriterion("course_selected_stu <", value, "courseSelectedStu");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedStuLessThanOrEqualTo(String value)
        {
            addCriterion("course_selected_stu <=", value, "courseSelectedStu");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedStuLike(String value)
        {
            addCriterion("course_selected_stu like", value, "courseSelectedStu");
```

```
            return (Criteria) this;
        }

        public Criteria andCourseSelectedStuNotLike(String value)
        {
            addCriterion("course_selected_stu not like", value, "courseSelectedStu");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedStuIn(List<String> values)
        {
            addCriterion("course_selected_stu in", values, "courseSelectedStu");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedStuNotIn(List<String> values)
        {
            addCriterion("course_selected_stu not in", values, "courseSelectedStu");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedStuBetween(String value1, String value2)
        {
            addCriterion("course_selected_stu between", value1, value2, "courseSelectedStu");

            return (Criteria) this;
        }

        public Criteria andCourseSelectedStuNotBetween(String value1, String value2)
        {
            addCriterion("course_selected_stu not between", value1, value2, "courseSelectedStu");

            return (Criteria) this;
        }

        public Criteria andCourseStatusIsNull()
        {
            addCriterion("course_status is null");

            return (Criteria) this;
        }
```

```
        public Criteria andCourseStatusIsNotNull()
        {
            addCriterion("course_status is not null");

            return (Criteria) this;
        }

        public Criteria andCourseStatusEqualTo(String value)
        {
            addCriterion("course_status =", value, "courseStatus");

            return (Criteria) this;
        }

        public Criteria andCourseStatusNotEqualTo(String value)
        {
            addCriterion("course_status <>", value, "courseStatus");

            return (Criteria) this;
        }

        public Criteria andCourseStatusGreaterThan(String value)
        {
            addCriterion("course_status >", value, "courseStatus");

            return (Criteria) this;
        }

        public Criteria andCourseStatusGreaterThanOrEqualTo(String value)
        {
            addCriterion("course_status >=", value, "courseStatus");

            return (Criteria) this;
        }

        public Criteria andCourseStatusLessThan(String value)
        {
            addCriterion("course_status <", value, "courseStatus");

            return (Criteria) this;
        }

        public Criteria andCourseStatusLessThanOrEqualTo(String value)
```

```
        {
            addCriterion("course_status <=", value, "courseStatus");

            return (Criteria) this;
        }

        public Criteria andCourseStatusLike(String value)
        {
            addCriterion("course_status like", value, "courseStatus");

            return (Criteria) this;
        }

        public Criteria andCourseStatusNotLike(String value)
        {
            addCriterion("course_status not like", value, "courseStatus");

            return (Criteria) this;
        }

        public Criteria andCourseStatusIn(List<String> values)
        {
            addCriterion("course_status in", values, "courseStatus");

            return (Criteria) this;
        }

        public Criteria andCourseStatusNotIn(List<String> values)
        {
            addCriterion("course_status not in", values, "courseStatus");

            return (Criteria) this;
        }

        public Criteria andCourseStatusBetween(String value1, String value2)
        {
            addCriterion("course_status between", value1, value2, "courseStatus");

            return (Criteria) this;
        }

        public Criteria andCourseStatusNotBetween(String value1, String value2)
        {
            addCriterion("course_status not between", value1, value2, "courseStatus");
```

```
            return (Criteria) this;
        }

        public Criteria andGradeIsNull()
        {
            addCriterion("grade is null");

            return (Criteria) this;
        }

        public Criteria andGradeIsNotNull()
        {
            addCriterion("grade is not null");

            return (Criteria) this;
        }

        public Criteria andGradeEqualTo(Integer value)
        {
            addCriterion("grade =", value, "grade");

            return (Criteria) this;
        }

        public Criteria andGradeNotEqualTo(Integer value)
        {
            addCriterion("grade <>", value, "grade");

            return (Criteria) this;
        }

        public Criteria andGradeGreaterThan(Integer value)
        {
            addCriterion("grade >", value, "grade");

            return (Criteria) this;
        }

        public Criteria andGradeGreaterThanOrEqualTo(Integer value)
        {
            addCriterion("grade >=", value, "grade");

            return (Criteria) this;
```

```
        }

        public Criteria andGradeLessThan(Integer value)
        {
            addCriterion("grade <", value, "grade");

            return (Criteria) this;
        }

        public Criteria andGradeLessThanOrEqualTo(Integer value)
        {
            addCriterion("grade <=", value, "grade");

            return (Criteria) this;
        }

        public Criteria andGradeIn(List<Integer> values)
        {
            addCriterion("grade in", values, "grade");

            return (Criteria) this;
        }

        public Criteria andGradeNotIn(List<Integer> values)
        {
            addCriterion("grade not in", values, "grade");

            return (Criteria) this;
        }

        public Criteria andGradeBetween(Integer value1, Integer value2)
        {
            addCriterion("grade between", value1, value2, "grade");

            return (Criteria) this;
        }

        public Criteria andGradeNotBetween(Integer value1, Integer value2)
        {
            addCriterion("grade not between", value1, value2, "grade");

            return (Criteria) this;
        }
    }
```

```
public static class Criteria extends GeneratedCriteria
{
    protected Criteria()
    {
        super();
    }
}

public static class Criterion
{
    private final String condition;
    private Object value;
    private Object secondValue;
    private boolean noValue;
    private boolean singleValue;
    private boolean betweenValue;
    private boolean listValue;
    private final String typeHandler;

    public String getCondition()
    {
        return condition;
    }

    public Object getValue()
    {
        return value;
    }

    public Object getSecondValue()
    {
        return secondValue;
    }

    public boolean isNoValue()
    {
        return noValue;
    }

    public boolean isSingleValue()
    {
        return singleValue;
    }
```

```
        public boolean isBetweenValue()
        {
            return betweenValue;
        }

        public boolean isListValue()
        {
            return listValue;
        }

        public String getTypeHandler()
        {
            return typeHandler;
        }

        protected Criterion(String condition)
        {
            super();
            this.condition = condition;
            this.typeHandler = null;
            this.noValue = true;
        }

        protected Criterion(String condition, Object value, String typeHandler)
        {
            super();
            this.condition = condition;
            this.value = value;
            this.typeHandler = typeHandler;

            if (value instanceof List<?>)
            {
                this.listValue = true;
            }
            else
            {
                this.singleValue = true;
            }
        }

        protected Criterion(String condition, Object value)
        {
            this(condition, value, null);
```

```
        }

        protected Criterion(String condition, Object value, Object secondValue, String typeHandler
        {
            super();
            this.condition = condition;
            this.value = value;
            this.secondValue = secondValue;
            this.typeHandler = typeHandler;
            this.betweenValue = true;
        }

        protected Criterion(String condition, Object value, Object secondValue)
        {
            this(condition, value, secondValue, null);
        }
    }
}
```

2.1.6 Student.java

```java
package com.myxh.studentmanagementsystem.bean;

import java.util.Date;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public class Student
{
    private String id;
    private String studentName;
    private String idCard;
    private String phoneNumber;
    private Date startYear;
    private String studentSource;
    private String major;
    private String className;

    public String getId()
    {
        return id;
    }

    public void setId(String id)
    {
        this.id = id == null ? null : id.trim();
    }

    public String getStudentName()
    {
        return studentName;
    }

    public void setStudentName(String studentName)
    {
        this.studentName = studentName == null ? null : studentName.trim();
    }

    public String getIdCard()
    {
        return idCard;
    }
```

```java
    public void setIdCard(String idCard)
    {
        this.idCard = idCard == null ? null : idCard.trim();
    }

    public String getPhoneNumber()
    {
        return phoneNumber;
    }

    public void setPhoneNumber(String phoneNumber)
    {
        this.phoneNumber = phoneNumber == null ? null : phoneNumber.trim();
    }

    public Date getStartYear()
    {
        return startYear;
    }

    public void setStartYear(Date startYear)
    {
        this.startYear = startYear;
    }

    public String getStudentSource()
    {
        return studentSource;
    }

    public void setStudentSource(String studentSource)
    {
        this.studentSource = studentSource == null ? null : studentSource.trim();
    }

    public String getMajor()
    {
        return major;
    }

    public void setMajor(String major)
    {
        this.major = major == null ? null : major.trim();
    }
```

```
    public String getClassName()
    {
        return className;
    }

    public void setClassName(String className)
    {
        this.className = className == null ? null : className.trim();
    }
}
```

2.1.7 StudentExample.java

```java
package com.myxh.studentmanagementsystem.bean;

import java.util.ArrayList;
import java.util.Date;
import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public class StudentExample
{
    protected String orderByClause;
    protected boolean distinct;
    protected List<Criteria> oredCriteria;

    public StudentExample()
    {
        oredCriteria = new ArrayList<>();
    }

    public void setOrderByClause(String orderByClause)
    {
        this.orderByClause = orderByClause;
    }

    public String getOrderByClause()
    {
        return orderByClause;
    }

    public void setDistinct(boolean distinct)
    {
        this.distinct = distinct;
    }

    public boolean isDistinct()
    {
        return distinct;
    }

    public List<Criteria> getOredCriteria()
    {
        return oredCriteria;
```

```
    }

    public void or(Criteria criteria)
    {
        oredCriteria.add(criteria);
    }

    public Criteria or()
    {
        Criteria criteria = createCriteriaInternal();
        oredCriteria.add(criteria);

        return criteria;
    }

    public Criteria createCriteria()
    {
        Criteria criteria = createCriteriaInternal();

        if (oredCriteria.size() == 0)
        {
            oredCriteria.add(criteria);
        }

        return criteria;
    }

    protected Criteria createCriteriaInternal()
    {
        return new Criteria();
    }

    public void clear()
    {
        oredCriteria.clear();
        orderByClause = null;
        distinct = false;
    }

    protected abstract static class GeneratedCriteria
    {
        protected List<Criterion> criteria;

        protected GeneratedCriteria()
```

```
        {
            super();
            criteria = new ArrayList<>();
        }

        public boolean isValid()
        {
            return criteria.size() > 0;
        }

        public List<Criterion> getAllCriteria()
        {
            return criteria;
        }

        public List<Criterion> getCriteria()
        {
            return criteria;
        }

        protected void addCriterion(String condition)
        {
            if (condition == null)
            {
                throw new RuntimeException("Value for condition cannot be null");
            }

            criteria.add(new Criterion(condition));
        }

        protected void addCriterion(String condition, Object value, String property)
        {
            if (value == null)
            {
                throw new RuntimeException("Value for " + property + " cannot be null");
            }

            criteria.add(new Criterion(condition, value));
        }

        protected void addCriterion(String condition, Object value1, Object value2, String propert
        {
            if (value1 == null || value2 == null)
            {
```

```
            throw new RuntimeException("Between values for " + property + " cannot be null");
        }

        criteria.add(new Criterion(condition, value1, value2));
    }

    protected void addCriterionForJDBCDate(String condition, Date value, String property)
    {
        if (value == null)
        {
            throw new RuntimeException("Value for " + property + " cannot be null");
        }

        addCriterion(condition, new java.sql.Date(value.getTime()), property);
    }

    protected void addCriterionForJDBCDate(String condition, List<Date> values, String propert
    {
        if (values == null || values.size() == 0)
        {
            throw new RuntimeException("Value list for " + property + " cannot be null or empt
        }

        List<java.sql.Date> dateList = new ArrayList<>();

        for (Date value : values)
        {
            dateList.add(new java.sql.Date(value.getTime()));
        }

        addCriterion(condition, dateList, property);
    }

    protected void addCriterionForJDBCDate(String condition, Date value1, Date value2, String
    {
        if (value1 == null || value2 == null)
        {
            throw new RuntimeException("Between values for " + property + " cannot be null");
        }

        addCriterion(condition, new java.sql.Date(value1.getTime()), new java.sql.Date(value2.
    }

    public Criteria andIdIsNull()
```

```
        {
            addCriterion("ID is null");

            return (Criteria) this;
        }

        public Criteria andIdIsNotNull()
        {
            addCriterion("ID is not null");

            return (Criteria) this;
        }

        public Criteria andIdEqualTo(String value)
        {
            addCriterion("ID =", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdNotEqualTo(String value)
        {
            addCriterion("ID <>", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdGreaterThan(String value)
        {
            addCriterion("ID >", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdGreaterThanOrEqualTo(String value)
        {
            addCriterion("ID >=", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdLessThan(String value)
        {
            addCriterion("ID <", value, "id");
```

```
            return (Criteria) this;
        }

        public Criteria andIdLessThanOrEqualTo(String value)
        {
            addCriterion("ID <=", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdLike(String value)
        {
            addCriterion("ID like", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdNotLike(String value)
        {
            addCriterion("ID not like", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdIn(List<String> values)
        {
            addCriterion("ID in", values, "id");

            return (Criteria) this;
        }

        public Criteria andIdNotIn(List<String> values)
        {
            addCriterion("ID not in", values, "id");

            return (Criteria) this;
        }

        public Criteria andIdBetween(String value1, String value2)
        {
            addCriterion("ID between", value1, value2, "id");

            return (Criteria) this;
```

```
        }

        public Criteria andIdNotBetween(String value1, String value2)
        {
            addCriterion("ID not between", value1, value2, "id");

            return (Criteria) this;
        }

        public Criteria andStudentNameIsNull()
        {
            addCriterion("student_name is null");

            return (Criteria) this;
        }

        public Criteria andStudentNameIsNotNull()
        {
            addCriterion("student_name is not null");

            return (Criteria) this;
        }

        public Criteria andStudentNameEqualTo(String value)
        {
            addCriterion("student_name =", value, "studentName");

            return (Criteria) this;
        }

        public Criteria andStudentNameNotEqualTo(String value)
        {
            addCriterion("student_name <>", value, "studentName");

            return (Criteria) this;
        }

        public Criteria andStudentNameGreaterThan(String value)
        {
            addCriterion("student_name >", value, "studentName");

            return (Criteria) this;
        }
```

```
        public Criteria andStudentNameGreaterThanOrEqualTo(String value)
        {
            addCriterion("student_name >=", value, "studentName");

            return (Criteria) this;
        }

        public Criteria andStudentNameLessThan(String value)
        {
            addCriterion("student_name <", value, "studentName");

            return (Criteria) this;
        }

        public Criteria andStudentNameLessThanOrEqualTo(String value)
        {
            addCriterion("student_name <=", value, "studentName");

            return (Criteria) this;
        }

        public Criteria andStudentNameLike(String value)
        {
            addCriterion("student_name like", value, "studentName");

            return (Criteria) this;
        }

        public Criteria andStudentNameNotLike(String value)
        {
            addCriterion("student_name not like", value, "studentName");

            return (Criteria) this;
        }

        public Criteria andStudentNameIn(List<String> values)
        {
            addCriterion("student_name in", values, "studentName");

            return (Criteria) this;
        }

        public Criteria andStudentNameNotIn(List<String> values)
        {
```

```
            addCriterion("student_name not in", values, "studentName");

            return (Criteria) this;
        }

        public Criteria andStudentNameBetween(String value1, String value2)
        {
            addCriterion("student_name between", value1, value2, "studentName");

            return (Criteria) this;
        }

        public Criteria andStudentNameNotBetween(String value1, String value2)
        {
            addCriterion("student_name not between", value1, value2, "studentName");

            return (Criteria) this;
        }

        public Criteria andIdCardIsNull()
        {
            addCriterion("ID_card is null");

            return (Criteria) this;
        }

        public Criteria andIdCardIsNotNull()
        {
            addCriterion("ID_card is not null");

            return (Criteria) this;
        }

        public Criteria andIdCardEqualTo(String value)
        {
            addCriterion("ID_card =", value, "idCard");

            return (Criteria) this;
        }

        public Criteria andIdCardNotEqualTo(String value)
        {
            addCriterion("ID_card <>", value, "idCard");
```

```
            return (Criteria) this;
        }

        public Criteria andIdCardGreaterThan(String value)
        {
            addCriterion("ID_card >", value, "idCard");

            return (Criteria) this;
        }

        public Criteria andIdCardGreaterThanOrEqualTo(String value)
        {
            addCriterion("ID_card >=", value, "idCard");

            return (Criteria) this;
        }

        public Criteria andIdCardLessThan(String value)
        {
            addCriterion("ID_card <", value, "idCard");

            return (Criteria) this;
        }

        public Criteria andIdCardLessThanOrEqualTo(String value)
        {
            addCriterion("ID_card <=", value, "idCard");

            return (Criteria) this;
        }

        public Criteria andIdCardLike(String value)
        {
            addCriterion("ID_card like", value, "idCard");

            return (Criteria) this;
        }

        public Criteria andIdCardNotLike(String value)
        {
            addCriterion("ID_card not like", value, "idCard");

            return (Criteria) this;
        }
```

```
        public Criteria andIdCardIn(List<String> values)
        {
            addCriterion("ID_card in", values, "idCard");

            return (Criteria) this;
        }

        public Criteria andIdCardNotIn(List<String> values)
        {
            addCriterion("ID_card not in", values, "idCard");

            return (Criteria) this;
        }

        public Criteria andIdCardBetween(String value1, String value2)
        {
            addCriterion("ID_card between", value1, value2, "idCard");

            return (Criteria) this;
        }

        public Criteria andIdCardNotBetween(String value1, String value2)
        {
            addCriterion("ID_card not between", value1, value2, "idCard");

            return (Criteria) this;
        }

        public Criteria andPhoneNumberIsNull()
        {
            addCriterion("phone_number is null");

            return (Criteria) this;
        }

        public Criteria andPhoneNumberIsNotNull()
        {
            addCriterion("phone_number is not null");

            return (Criteria) this;
        }

        public Criteria andPhoneNumberEqualTo(String value)
```

```
        {
            addCriterion("phone_number =", value, "phoneNumber");

            return (Criteria) this;
        }

        public Criteria andPhoneNumberNotEqualTo(String value)
        {
            addCriterion("phone_number <>", value, "phoneNumber");

            return (Criteria) this;
        }

        public Criteria andPhoneNumberGreaterThan(String value)
        {
            addCriterion("phone_number >", value, "phoneNumber");

            return (Criteria) this;
        }

        public Criteria andPhoneNumberGreaterThanOrEqualTo(String value)
        {
            addCriterion("phone_number >=", value, "phoneNumber");

            return (Criteria) this;
        }

        public Criteria andPhoneNumberLessThan(String value)
        {
            addCriterion("phone_number <", value, "phoneNumber");

            return (Criteria) this;
        }

        public Criteria andPhoneNumberLessThanOrEqualTo(String value)
        {
            addCriterion("phone_number <=", value, "phoneNumber");

            return (Criteria) this;
        }

        public Criteria andPhoneNumberLike(String value)
        {
            addCriterion("phone_number like", value, "phoneNumber");
```

```
            return (Criteria) this;
        }

        public Criteria andPhoneNumberNotLike(String value)
        {
            addCriterion("phone_number not like", value, "phoneNumber");

            return (Criteria) this;
        }

        public Criteria andPhoneNumberIn(List<String> values)
        {
            addCriterion("phone_number in", values, "phoneNumber");

            return (Criteria) this;
        }

        public Criteria andPhoneNumberNotIn(List<String> values)
        {
            addCriterion("phone_number not in", values, "phoneNumber");

            return (Criteria) this;
        }

        public Criteria andPhoneNumberBetween(String value1, String value2)
        {
            addCriterion("phone_number between", value1, value2, "phoneNumber");

            return (Criteria) this;
        }

        public Criteria andPhoneNumberNotBetween(String value1, String value2)
        {
            addCriterion("phone_number not between", value1, value2, "phoneNumber");

            return (Criteria) this;
        }

        public Criteria andStartYearIsNull()
        {
            addCriterion("start_year is null");

            return (Criteria) this;
```

```
        }

        public Criteria andStartYearIsNotNull()
        {
            addCriterion("start_year is not null");

            return (Criteria) this;
        }

        public Criteria andStartYearEqualTo(Date value)
        {
            addCriterionForJDBCDate("start_year =", value, "startYear");

            return (Criteria) this;
        }

        public Criteria andStartYearNotEqualTo(Date value)
        {
            addCriterionForJDBCDate("start_year <>", value, "startYear");

            return (Criteria) this;
        }

        public Criteria andStartYearGreaterThan(Date value)
        {
            addCriterionForJDBCDate("start_year >", value, "startYear");

            return (Criteria) this;
        }

        public Criteria andStartYearGreaterThanOrEqualTo(Date value)
        {
            addCriterionForJDBCDate("start_year >=", value, "startYear");

            return (Criteria) this;
        }

        public Criteria andStartYearLessThan(Date value)
        {
            addCriterionForJDBCDate("start_year <", value, "startYear");

            return (Criteria) this;
        }
```

```
        public Criteria andStartYearLessThanOrEqualTo(Date value)
        {
            addCriterionForJDBCDate("start_year <=", value, "startYear");

            return (Criteria) this;
        }

        public Criteria andStartYearIn(List<Date> values)
        {
            addCriterionForJDBCDate("start_year in", values, "startYear");

            return (Criteria) this;
        }

        public Criteria andStartYearNotIn(List<Date> values)
        {
            addCriterionForJDBCDate("start_year not in", values, "startYear");

            return (Criteria) this;
        }

        public Criteria andStartYearBetween(Date value1, Date value2)
        {
            addCriterionForJDBCDate("start_year between", value1, value2, "startYear");

            return (Criteria) this;
        }

        public Criteria andStartYearNotBetween(Date value1, Date value2)
        {
            addCriterionForJDBCDate("start_year not between", value1, value2, "startYear");

            return (Criteria) this;
        }

        public Criteria andStudentSourceIsNull()
        {
            addCriterion("student_source is null");

            return (Criteria) this;
        }

        public Criteria andStudentSourceIsNotNull()
        {
```

```
            addCriterion("student_source is not null");

            return (Criteria) this;
        }

        public Criteria andStudentSourceEqualTo(String value)
        {
            addCriterion("student_source =", value, "studentSource");

            return (Criteria) this;
        }

        public Criteria andStudentSourceNotEqualTo(String value)
        {
            addCriterion("student_source <>", value, "studentSource");

            return (Criteria) this;
        }

        public Criteria andStudentSourceGreaterThan(String value)
        {
            addCriterion("student_source >", value, "studentSource");

            return (Criteria) this;
        }

        public Criteria andStudentSourceGreaterThanOrEqualTo(String value)
        {
            addCriterion("student_source >=", value, "studentSource");

            return (Criteria) this;
        }

        public Criteria andStudentSourceLessThan(String value)
        {
            addCriterion("student_source <", value, "studentSource");

            return (Criteria) this;
        }

        public Criteria andStudentSourceLessThanOrEqualTo(String value)
        {
            addCriterion("student_source <=", value, "studentSource");
```

```
            return (Criteria) this;
        }

        public Criteria andStudentSourceLike(String value)
        {
            addCriterion("student_source like", value, "studentSource");

            return (Criteria) this;
        }

        public Criteria andStudentSourceNotLike(String value)
        {
            addCriterion("student_source not like", value, "studentSource");

            return (Criteria) this;
        }

        public Criteria andStudentSourceIn(List<String> values)
        {
            addCriterion("student_source in", values, "studentSource");

            return (Criteria) this;
        }

        public Criteria andStudentSourceNotIn(List<String> values)
        {
            addCriterion("student_source not in", values, "studentSource");

            return (Criteria) this;
        }

        public Criteria andStudentSourceBetween(String value1, String value2)
        {
            addCriterion("student_source between", value1, value2, "studentSource");

            return (Criteria) this;
        }

        public Criteria andStudentSourceNotBetween(String value1, String value2)
        {
            addCriterion("student_source not between", value1, value2, "studentSource");

            return (Criteria) this;
        }
```

```
        public Criteria andMajorIsNull()
        {
            addCriterion("major is null");

            return (Criteria) this;
        }

        public Criteria andMajorIsNotNull()
        {
            addCriterion("major is not null");

            return (Criteria) this;
        }

        public Criteria andMajorEqualTo(String value)
        {
            addCriterion("major =", value, "major");

            return (Criteria) this;
        }

        public Criteria andMajorNotEqualTo(String value)
        {
            addCriterion("major <>", value, "major");

            return (Criteria) this;
        }

        public Criteria andMajorGreaterThan(String value)
        {
            addCriterion("major >", value, "major");

            return (Criteria) this;
        }

        public Criteria andMajorGreaterThanOrEqualTo(String value)
        {
            addCriterion("major >=", value, "major");

            return (Criteria) this;
        }

        public Criteria andMajorLessThan(String value)
```

```
        {
            addCriterion("major <", value, "major");

            return (Criteria) this;
        }

        public Criteria andMajorLessThanOrEqualTo(String value)
        {
            addCriterion("major <=", value, "major");

            return (Criteria) this;
        }

        public Criteria andMajorLike(String value)
        {
            addCriterion("major like", value, "major");

            return (Criteria) this;
        }

        public Criteria andMajorNotLike(String value)
        {
            addCriterion("major not like", value, "major");

            return (Criteria) this;
        }

        public Criteria andMajorIn(List<String> values)
        {
            addCriterion("major in", values, "major");

            return (Criteria) this;
        }

        public Criteria andMajorNotIn(List<String> values)
        {
            addCriterion("major not in", values, "major");

            return (Criteria) this;
        }

        public Criteria andMajorBetween(String value1, String value2)
        {
            addCriterion("major between", value1, value2, "major");
```

```
            return (Criteria) this;
        }

        public Criteria andMajorNotBetween(String value1, String value2)
        {
            addCriterion("major not between", value1, value2, "major");

            return (Criteria) this;
        }

        public Criteria andClassNameIsNull()
        {
            addCriterion("class_name is null");

            return (Criteria) this;
        }

        public Criteria andClassNameIsNotNull()
        {
            addCriterion("class_name is not null");

            return (Criteria) this;
        }

        public Criteria andClassNameEqualTo(String value)
        {
            addCriterion("class_name =", value, "className");

            return (Criteria) this;
        }

        public Criteria andClassNameNotEqualTo(String value)
        {
            addCriterion("class_name <>", value, "className");

            return (Criteria) this;
        }

        public Criteria andClassNameGreaterThan(String value)
        {
            addCriterion("class_name >", value, "className");

            return (Criteria) this;
```

```
        }

        public Criteria andClassNameGreaterThanOrEqualTo(String value)
        {
            addCriterion("class_name >=", value, "className");

            return (Criteria) this;
        }

        public Criteria andClassNameLessThan(String value)
        {
            addCriterion("class_name <", value, "className");

            return (Criteria) this;
        }

        public Criteria andClassNameLessThanOrEqualTo(String value)
        {
            addCriterion("class_name <=", value, "className");

            return (Criteria) this;
        }

        public Criteria andClassNameLike(String value)
        {
            addCriterion("class_name like", value, "className");

            return (Criteria) this;
        }

        public Criteria andClassNameNotLike(String value)
        {
            addCriterion("class_name not like", value, "className");

            return (Criteria) this;
        }

        public Criteria andClassNameIn(List<String> values)
        {
            addCriterion("class_name in", values, "className");

            return (Criteria) this;
        }
```

```
        public Criteria andClassNameNotIn(List<String> values)
        {
            addCriterion("class_name not in", values, "className");

            return (Criteria) this;
        }

        public Criteria andClassNameBetween(String value1, String value2)
        {
            addCriterion("class_name between", value1, value2, "className");

            return (Criteria) this;
        }

        public Criteria andClassNameNotBetween(String value1, String value2)
        {
            addCriterion("class_name not between", value1, value2, "className");

            return (Criteria) this;
        }
    }

    public static class Criteria extends GeneratedCriteria
    {
        protected Criteria()
        {
            super();
        }
    }

    public static class Criterion
    {
        private final String condition;
        private Object value;
        private Object secondValue;
        private boolean noValue;
        private boolean singleValue;
        private boolean betweenValue;
        private boolean listValue;
        private final String typeHandler;

        public String getCondition()
        {
            return condition;
```

```
    }

    public Object getValue()
    {
        return value;
    }

    public Object getSecondValue()
    {
        return secondValue;
    }

    public boolean isNoValue()
    {
        return noValue;
    }

    public boolean isSingleValue()
    {
        return singleValue;
    }

    public boolean isBetweenValue()
    {
        return betweenValue;
    }

    public boolean isListValue()
    {
        return listValue;
    }

    public String getTypeHandler()
    {
        return typeHandler;
    }

    protected Criterion(String condition)
    {
        super();
        this.condition = condition;
        this.typeHandler = null;
        this.noValue = true;
    }
```

```
        protected Criterion(String condition, Object value, String typeHandler)
        {
            super();
            this.condition = condition;
            this.value = value;
            this.typeHandler = typeHandler;

            if (value instanceof List<?>)
            {
                this.listValue = true;
            }
            else
            {
                this.singleValue = true;
            }
        }

        protected Criterion(String condition, Object value)
        {
            this(condition, value, null);
        }

        protected Criterion(String condition, Object value, Object secondValue, String typeHandler
        {
            super();
            this.condition = condition;
            this.value = value;
            this.secondValue = secondValue;
            this.typeHandler = typeHandler;
            this.betweenValue = true;
        }

        protected Criterion(String condition, Object value, Object secondValue)
        {
            this(condition, value, secondValue, null);
        }
    }
}
```

2.1.8 TrainingProgram.java

```java
package com.myxh.studentmanagementsystem.bean;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public class TrainingProgram
{
    private Integer courseId;
    private String courseName;
    private String courseNature;
    private String courseMajor;
    private String courseTeacher;
    private String courseAttributes;
    private String preliminaryCourse;
    private String followupCourse;

    public TrainingProgram()
    {

    }

    public TrainingProgram(Integer courseId, String courseName, String courseNature, String course
    {
        this.courseId = courseId;
        this.courseName = courseName;
        this.courseNature = courseNature;
        this.courseMajor = courseMajor;
        this.courseTeacher = courseTeacher;
        this.courseAttributes = courseAttributes;
        this.preliminaryCourse = preliminaryCourse;
        this.followupCourse = followupCourse;
    }

    public Integer getCourseId()
    {
        return courseId;
    }

    public void setCourseId(Integer courseId)
    {
        this.courseId = courseId;
    }
```

```
    public String getCourseName()
    {
        return courseName;
    }

    public void setCourseName(String courseName)
    {
        this.courseName = courseName == null ? null : courseName.trim();
    }

    public String getCourseNature()
    {
        return courseNature;
    }

    public void setCourseNature(String courseNature)
    {
        this.courseNature = courseNature == null ? null : courseNature.trim();
    }

    public String getCourseMajor()
    {
        return courseMajor;
    }

    public void setCourseMajor(String courseMajor)
    {
        this.courseMajor = courseMajor == null ? null : courseMajor.trim();
    }

    public String getCourseTeacher()
    {
        return courseTeacher;
    }

    public void setCourseTeacher(String courseTeacher)
    {
        this.courseTeacher = courseTeacher == null ? null : courseTeacher.trim();
    }

    public String getCourseAttributes()
    {
        return courseAttributes;
    }
```

```
    public void setCourseAttributes(String courseAttributes)
    {
        this.courseAttributes = courseAttributes == null ? null : courseAttributes.trim();
    }

    public String getPreliminaryCourse()
    {
        return preliminaryCourse;
    }

    public void setPreliminaryCourse(String preliminaryCourse)
    {
        this.preliminaryCourse = preliminaryCourse == null ? null : preliminaryCourse.trim();
    }

    public String getFollowupCourse()
    {
        return followupCourse;
    }

    public void setFollowupCourse(String followupCourse)
    {
        this.followupCourse = followupCourse == null ? null : followupCourse.trim();
    }

    @Override
    public String toString()
    {
        return "TrainingProgram{" +
                "courseId=" + courseId +
                ", courseName='" + courseName + '\'' +
                ", courseNature='" + courseNature + '\'' +
                ", courseMajor='" + courseMajor + '\'' +
                ", courseTeacher='" + courseTeacher + '\'' +
                ", courseAttributes='" + courseAttributes + '\'' +
                ", preliminaryCourse='" + preliminaryCourse + '\'' +
                ", followupCourse='" + followupCourse + '\'' +
                '}';
    }
}
```

2.1.9 TrainingProgramExample.java

```java
package com.myxh.studentmanagementsystem.bean;

import java.util.ArrayList;
import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public class TrainingProgramExample
{
    protected String orderByClause;
    protected boolean distinct;
    protected List<Criteria> oredCriteria;

    public TrainingProgramExample()
    {
        oredCriteria = new ArrayList<>();
    }

    public void setOrderByClause(String orderByClause)
    {
        this.orderByClause = orderByClause;
    }

    public String getOrderByClause()
    {
        return orderByClause;
    }

    public void setDistinct(boolean distinct)
    {
        this.distinct = distinct;
    }

    public boolean isDistinct()
    {
        return distinct;
    }

    public List<Criteria> getOredCriteria()
    {
        return oredCriteria;
    }
```

```
    public void or(Criteria criteria)
    {
        oredCriteria.add(criteria);
    }

    public Criteria or()
    {
        Criteria criteria = createCriteriaInternal();
        oredCriteria.add(criteria);
        return criteria;
    }

    public Criteria createCriteria()
    {
        Criteria criteria = createCriteriaInternal();

        if (oredCriteria.size() == 0)
        {
            oredCriteria.add(criteria);
        }

        return criteria;
    }

    protected Criteria createCriteriaInternal()
    {
        return new Criteria();
    }

    public void clear()
    {
        oredCriteria.clear();
        orderByClause = null;
        distinct = false;
    }

    protected abstract static class GeneratedCriteria
    {
        protected List<Criterion> criteria;

        protected GeneratedCriteria()
        {
            super();
```

```
        criteria = new ArrayList<>();
    }

    public boolean isValid()
    {
        return criteria.size() > 0;
    }

    public List<Criterion> getAllCriteria()
    {
        return criteria;
    }

    public List<Criterion> getCriteria()
    {
        return criteria;
    }

    protected void addCriterion(String condition)
    {
        if (condition == null)
        {
            throw new RuntimeException("Value for condition cannot be null");
        }

        criteria.add(new Criterion(condition));
    }

    protected void addCriterion(String condition, Object value, String property)
    {
        if (value == null)
        {
            throw new RuntimeException("Value for " + property + " cannot be null");
        }

        criteria.add(new Criterion(condition, value));
    }

    protected void addCriterion(String condition, Object value1, Object value2, String propert
    {
        if (value1 == null || value2 == null)
        {
            throw new RuntimeException("Between values for " + property + " cannot be null");
        }
```

```
            criteria.add(new Criterion(condition, value1, value2));
        }

        public Criteria andCourseIdIsNull()
        {
            addCriterion("course_id is null");

            return (Criteria) this;
        }

        public Criteria andCourseIdIsNotNull()
        {
            addCriterion("course_id is not null");

            return (Criteria) this;
        }

        public Criteria andCourseIdEqualTo(Integer value)
        {
            addCriterion("course_id =", value, "courseId");

            return (Criteria) this;
        }

        public Criteria andCourseIdNotEqualTo(Integer value)
        {
            addCriterion("course_id <>", value, "courseId");

            return (Criteria) this;
        }

        public Criteria andCourseIdGreaterThan(Integer value)
        {
            addCriterion("course_id >", value, "courseId");

            return (Criteria) this;
        }

        public Criteria andCourseIdGreaterThanOrEqualTo(Integer value)
        {
            addCriterion("course_id >=", value, "courseId");

            return (Criteria) this;
```

```
        }

        public Criteria andCourseIdLessThan(Integer value)
        {
            addCriterion("course_id <", value, "courseId");

            return (Criteria) this;
        }

        public Criteria andCourseIdLessThanOrEqualTo(Integer value)
        {
            addCriterion("course_id <=", value, "courseId");

            return (Criteria) this;
        }

        public Criteria andCourseIdIn(List<Integer> values)
        {
            addCriterion("course_id in", values, "courseId");

            return (Criteria) this;
        }

        public Criteria andCourseIdNotIn(List<Integer> values)
        {
            addCriterion("course_id not in", values, "courseId");

            return (Criteria) this;
        }

        public Criteria andCourseIdBetween(Integer value1, Integer value2)
        {
            addCriterion("course_id between", value1, value2, "courseId");

            return (Criteria) this;
        }

        public Criteria andCourseIdNotBetween(Integer value1, Integer value2)
        {
            addCriterion("course_id not between", value1, value2, "courseId");

            return (Criteria) this;
        }
```

```
        public Criteria andCourseNameIsNull()
        {
            addCriterion("course_name is null");

            return (Criteria) this;
        }

        public Criteria andCourseNameIsNotNull()
        {
            addCriterion("course_name is not null");

            return (Criteria) this;
        }

        public Criteria andCourseNameEqualTo(String value)
        {
            addCriterion("course_name =", value, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNameNotEqualTo(String value)
        {
            addCriterion("course_name <>", value, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNameGreaterThan(String value)
        {
            addCriterion("course_name >", value, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNameGreaterThanOrEqualTo(String value)
        {
            addCriterion("course_name >=", value, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNameLessThan(String value)
        {
```

```
            addCriterion("course_name <", value, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNameLessThanOrEqualTo(String value)
        {
            addCriterion("course_name <=", value, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNameLike(String value)
        {
            addCriterion("course_name like", value, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNameNotLike(String value)
        {
            addCriterion("course_name not like", value, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNameIn(List<String> values)
        {
            addCriterion("course_name in", values, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNameNotIn(List<String> values)
        {
            addCriterion("course_name not in", values, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNameBetween(String value1, String value2)
        {
            addCriterion("course_name between", value1, value2, "courseName");
```

```
            return (Criteria) this;
        }

        public Criteria andCourseNameNotBetween(String value1, String value2)
        {
            addCriterion("course_name not between", value1, value2, "courseName");

            return (Criteria) this;
        }

        public Criteria andCourseNatureIsNull()
        {
            addCriterion("course_nature is null");

            return (Criteria) this;
        }

        public Criteria andCourseNatureIsNotNull()
        {
            addCriterion("course_nature is not null");

            return (Criteria) this;
        }

        public Criteria andCourseNatureEqualTo(String value)
        {
            addCriterion("course_nature =", value, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureNotEqualTo(String value)
        {
            addCriterion("course_nature <>", value, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureGreaterThan(String value)
        {
            addCriterion("course_nature >", value, "courseNature");

            return (Criteria) this;
        }
```

```
        public Criteria andCourseNatureGreaterThanOrEqualTo(String value)
        {
            addCriterion("course_nature >=", value, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureLessThan(String value)
        {
            addCriterion("course_nature <", value, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureLessThanOrEqualTo(String value)
        {
            addCriterion("course_nature <=", value, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureLike(String value)
        {
            addCriterion("course_nature like", value, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureNotLike(String value)
        {
            addCriterion("course_nature not like", value, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureIn(List<String> values)
        {
            addCriterion("course_nature in", values, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureNotIn(List<String> values)
```

```
        {
            addCriterion("course_nature not in", values, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureBetween(String value1, String value2)
        {
            addCriterion("course_nature between", value1, value2, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseNatureNotBetween(String value1, String value2)
        {
            addCriterion("course_nature not between", value1, value2, "courseNature");

            return (Criteria) this;
        }

        public Criteria andCourseMajorIsNull()
        {
            addCriterion("course_major is null");

            return (Criteria) this;
        }

        public Criteria andCourseMajorIsNotNull()
        {
            addCriterion("course_major is not null");

            return (Criteria) this;
        }

        public Criteria andCourseMajorEqualTo(String value)
        {
            addCriterion("course_major =", value, "courseMajor");

            return (Criteria) this;
        }

        public Criteria andCourseMajorNotEqualTo(String value)
        {
            addCriterion("course_major <>", value, "courseMajor");
```

```
            return (Criteria) this;
        }

        public Criteria andCourseMajorGreaterThan(String value)
        {
            addCriterion("course_major >", value, "courseMajor");

            return (Criteria) this;
        }

        public Criteria andCourseMajorGreaterThanOrEqualTo(String value)
        {
            addCriterion("course_major >=", value, "courseMajor");

            return (Criteria) this;
        }

        public Criteria andCourseMajorLessThan(String value)
        {
            addCriterion("course_major <", value, "courseMajor");

           return (Criteria) this;
        }

        public Criteria andCourseMajorLessThanOrEqualTo(String value)
        {
            addCriterion("course_major <=", value, "courseMajor");

            return (Criteria) this;
        }

        public Criteria andCourseMajorLike(String value)
        {
            addCriterion("course_major like", value, "courseMajor");

            return (Criteria) this;
        }

        public Criteria andCourseMajorNotLike(String value)
        {
            addCriterion("course_major not like", value, "courseMajor");

            return (Criteria) this;
```

```
        }

        public Criteria andCourseMajorIn(List<String> values)
        {
            addCriterion("course_major in", values, "courseMajor");

            return (Criteria) this;
        }

        public Criteria andCourseMajorNotIn(List<String> values)
        {
            addCriterion("course_major not in", values, "courseMajor");

            return (Criteria) this;
        }

        public Criteria andCourseMajorBetween(String value1, String value2)
        {
            addCriterion("course_major between", value1, value2, "courseMajor");

            return (Criteria) this;
        }

        public Criteria andCourseMajorNotBetween(String value1, String value2)
        {
            addCriterion("course_major not between", value1, value2, "courseMajor");

            return (Criteria) this;
        }

        public Criteria andCourseTeacherIsNull()
        {
            addCriterion("course_teacher is null");

            return (Criteria) this;
        }

        public Criteria andCourseTeacherIsNotNull()
        {
            addCriterion("course_teacher is not null");

            return (Criteria) this;
        }
```

```
        public Criteria andCourseTeacherEqualTo(String value)
        {
            addCriterion("course_teacher =", value, "courseTeacher");

            return (Criteria) this;
        }

        public Criteria andCourseTeacherNotEqualTo(String value)
        {
            addCriterion("course_teacher <>", value, "courseTeacher");

            return (Criteria) this;
        }

        public Criteria andCourseTeacherGreaterThan(String value)
        {
            addCriterion("course_teacher >", value, "courseTeacher");

            return (Criteria) this;
        }

        public Criteria andCourseTeacherGreaterThanOrEqualTo(String value)
        {
            addCriterion("course_teacher >=", value, "courseTeacher");

            return (Criteria) this;
        }

        public Criteria andCourseTeacherLessThan(String value)
        {
            addCriterion("course_teacher <", value, "courseTeacher");

            return (Criteria) this;
        }

        public Criteria andCourseTeacherLessThanOrEqualTo(String value)
        {
            addCriterion("course_teacher <=", value, "courseTeacher");

            return (Criteria) this;
        }

        public Criteria andCourseTeacherLike(String value)
        {
```

```
            addCriterion("course_teacher like", value, "courseTeacher");

            return (Criteria) this;
        }

        public Criteria andCourseTeacherNotLike(String value)
        {
            addCriterion("course_teacher not like", value, "courseTeacher");

            return (Criteria) this;
        }

        public Criteria andCourseTeacherIn(List<String> values)
        {
            addCriterion("course_teacher in", values, "courseTeacher");

            return (Criteria) this;
        }

        public Criteria andCourseTeacherNotIn(List<String> values)
        {
            addCriterion("course_teacher not in", values, "courseTeacher");

            return (Criteria) this;
        }

        public Criteria andCourseTeacherBetween(String value1, String value2)
        {
            addCriterion("course_teacher between", value1, value2, "courseTeacher");

            return (Criteria) this;
        }

        public Criteria andCourseTeacherNotBetween(String value1, String value2)
        {
            addCriterion("course_teacher not between", value1, value2, "courseTeacher");

            return (Criteria) this;
        }

        public Criteria andCourseAttributesIsNull()
        {
            addCriterion("course_attributes is null");
```

```
            return (Criteria) this;
        }

        public Criteria andCourseAttributesIsNotNull()
        {
            addCriterion("course_attributes is not null");

            return (Criteria) this;
        }

        public Criteria andCourseAttributesEqualTo(String value)
        {
            addCriterion("course_attributes =", value, "courseAttributes");

            return (Criteria) this;
        }

        public Criteria andCourseAttributesNotEqualTo(String value)
        {
            addCriterion("course_attributes <>", value, "courseAttributes");

            return (Criteria) this;
        }

        public Criteria andCourseAttributesGreaterThan(String value)
        {
            addCriterion("course_attributes >", value, "courseAttributes");

            return (Criteria) this;
        }

        public Criteria andCourseAttributesGreaterThanOrEqualTo(String value)
        {
            addCriterion("course_attributes >=", value, "courseAttributes");

            return (Criteria) this;
        }

        public Criteria andCourseAttributesLessThan(String value)
        {
            addCriterion("course_attributes <", value, "courseAttributes");

            return (Criteria) this;
        }
```

```
        public Criteria andCourseAttributesLessThanOrEqualTo(String value)
        {
            addCriterion("course_attributes <=", value, "courseAttributes");

            return (Criteria) this;
        }

        public Criteria andCourseAttributesLike(String value)
        {
            addCriterion("course_attributes like", value, "courseAttributes");

            return (Criteria) this;
        }

        public Criteria andCourseAttributesNotLike(String value)
        {
            addCriterion("course_attributes not like", value, "courseAttributes");

            return (Criteria) this;
        }

        public Criteria andCourseAttributesIn(List<String> values)
        {
            addCriterion("course_attributes in", values, "courseAttributes");

            return (Criteria) this;
        }

        public Criteria andCourseAttributesNotIn(List<String> values)
        {
            addCriterion("course_attributes not in", values, "courseAttributes");

            return (Criteria) this;
        }

        public Criteria andCourseAttributesBetween(String value1, String value2)
        {
            addCriterion("course_attributes between", value1, value2, "courseAttributes");

            return (Criteria) this;
        }

        public Criteria andCourseAttributesNotBetween(String value1, String value2)
```

```
        {
            addCriterion("course_attributes not between", value1, value2, "courseAttributes");
            return (Criteria) this;
        }

        public Criteria andPreliminaryCourseIsNull()
        {
            addCriterion("preliminary_course is null");
            return (Criteria) this;
        }

        public Criteria andPreliminaryCourseIsNotNull()
        {
            addCriterion("preliminary_course is not null");
            return (Criteria) this;
        }

        public Criteria andPreliminaryCourseEqualTo(String value)
        {
            addCriterion("preliminary_course =", value, "preliminaryCourse");
            return (Criteria) this;
        }

        public Criteria andPreliminaryCourseNotEqualTo(String value)
        {
            addCriterion("preliminary_course <>", value, "preliminaryCourse");
            return (Criteria) this;
        }

        public Criteria andPreliminaryCourseGreaterThan(String value)
        {
            addCriterion("preliminary_course >", value, "preliminaryCourse");
            return (Criteria) this;
        }

        public Criteria andPreliminaryCourseGreaterThanOrEqualTo(String value)
        {
            addCriterion("preliminary_course >=", value, "preliminaryCourse");
```

```
            return (Criteria) this;
        }

        public Criteria andPreliminaryCourseLessThan(String value)
        {
            addCriterion("preliminary_course <", value, "preliminaryCourse");

            return (Criteria) this;
        }

        public Criteria andPreliminaryCourseLessThanOrEqualTo(String value)
        {
            addCriterion("preliminary_course <=", value, "preliminaryCourse");

            return (Criteria) this;
        }

        public Criteria andPreliminaryCourseLike(String value)
        {
            addCriterion("preliminary_course like", value, "preliminaryCourse");

            return (Criteria) this;
        }

        public Criteria andPreliminaryCourseNotLike(String value)
        {
            addCriterion("preliminary_course not like", value, "preliminaryCourse");

            return (Criteria) this;
        }

        public Criteria andPreliminaryCourseIn(List<String> values)
        {
            addCriterion("preliminary_course in", values, "preliminaryCourse");

            return (Criteria) this;
        }

        public Criteria andPreliminaryCourseNotIn(List<String> values)
        {
            addCriterion("preliminary_course not in", values, "preliminaryCourse");

            return (Criteria) this;
```

```
        }

        public Criteria andPreliminaryCourseBetween(String value1, String value2)
        {
            addCriterion("preliminary_course between", value1, value2, "preliminaryCourse");

            return (Criteria) this;
        }

        public Criteria andPreliminaryCourseNotBetween(String value1, String value2)
        {
            addCriterion("preliminary_course not between", value1, value2, "preliminaryCourse");

            return (Criteria) this;
        }

        public Criteria andFollowupCourseIsNull()
        {
            addCriterion("followup_course is null");

            return (Criteria) this;
        }

        public Criteria andFollowupCourseIsNotNull()
        {
            addCriterion("followup_course is not null");

            return (Criteria) this;
        }

        public Criteria andFollowupCourseEqualTo(String value)
        {
            addCriterion("followup_course =", value, "followupCourse");

            return (Criteria) this;
        }

        public Criteria andFollowupCourseNotEqualTo(String value)
        {
            addCriterion("followup_course <>", value, "followupCourse");

            return (Criteria) this;
        }
```

```
        public Criteria andFollowupCourseGreaterThan(String value)
        {
            addCriterion("followup_course >", value, "followupCourse");

            return (Criteria) this;
        }

        public Criteria andFollowupCourseGreaterThanOrEqualTo(String value)
        {
            addCriterion("followup_course >=", value, "followupCourse");

            return (Criteria) this;
        }

        public Criteria andFollowupCourseLessThan(String value)
        {
            addCriterion("followup_course <", value, "followupCourse");

            return (Criteria) this;
        }

        public Criteria andFollowupCourseLessThanOrEqualTo(String value)
        {
            addCriterion("followup_course <=", value, "followupCourse");

            return (Criteria) this;
        }

        public Criteria andFollowupCourseLike(String value)
        {
            addCriterion("followup_course like", value, "followupCourse");

            return (Criteria) this;
        }

        public Criteria andFollowupCourseNotLike(String value)
        {
            addCriterion("followup_course not like", value, "followupCourse");

            return (Criteria) this;
        }

        public Criteria andFollowupCourseIn(List<String> values)
        {
```

```
            addCriterion("followup_course in", values, "followupCourse");

            return (Criteria) this;
        }

        public Criteria andFollowupCourseNotIn(List<String> values)
        {
            addCriterion("followup_course not in", values, "followupCourse");

            return (Criteria) this;
        }

        public Criteria andFollowupCourseBetween(String value1, String value2)
        {
            addCriterion("followup_course between", value1, value2, "followupCourse");

            return (Criteria) this;
        }

        public Criteria andFollowupCourseNotBetween(String value1, String value2)
        {
            addCriterion("followup_course not between", value1, value2, "followupCourse");

            return (Criteria) this;
        }
    }

    public static class Criteria extends GeneratedCriteria
    {
        protected Criteria()
        {
            super();
        }
    }

    public static class Criterion
    {
        private final String condition;
        private Object value;
        private Object secondValue;
        private boolean noValue;
        private boolean singleValue;
        private boolean betweenValue;
        private boolean listValue;
```

```
        private final String typeHandler;

        public String getCondition()
        {
            return condition;
        }

        public Object getValue()
        {
            return value;
        }

        public Object getSecondValue()
        {
            return secondValue;
        }

        public boolean isNoValue()
        {
            return noValue;
        }

        public boolean isSingleValue()
        {
            return singleValue;
        }

        public boolean isBetweenValue()
        {
            return betweenValue;
        }

        public boolean isListValue()
        {
            return listValue;
        }

        public String getTypeHandler()
        {
            return typeHandler;
        }

        protected Criterion(String condition)
        {
```

```
            super();
            this.condition = condition;
            this.typeHandler = null;
            this.noValue = true;
        }

        protected Criterion(String condition, Object value, String typeHandler)
        {
            super();
            this.condition = condition;
            this.value = value;
            this.typeHandler = typeHandler;

            if (value instanceof List<?>)
            {
                this.listValue = true;
            }
            else
            {
                this.singleValue = true;
            }
        }

        protected Criterion(String condition, Object value)
        {
            this(condition, value, null);
        }

        protected Criterion(String condition, Object value, Object secondValue, String typeHandler
        {
            super();
            this.condition = condition;
            this.value = value;
            this.secondValue = secondValue;
            this.typeHandler = typeHandler;
            this.betweenValue = true;
        }

        protected Criterion(String condition, Object value, Object secondValue)
        {
            this(condition, value, secondValue, null);
        }
    }
}
```

2.1.10 UStudent.java

```java
package com.myxh.studentmanagementsystem.bean;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public class UStudent
{
    private String id;
    private String password;

    public UStudent()
    {

    }

    public UStudent(String id, String password)
    {
        this.id = id;
        this.password = password;
    }

    public String getId()
    {
        return id;
    }

    public void setId(String id)
    {
        this.id = id == null ? null : id.trim();
    }

    public String getPassword()
    {
        return password;
    }

    public void setPassword(String password)
    {
        this.password = password == null ? null : password.trim();
    }

    @Override
    public String toString()
```

```
    {
        return "UStudent{" +
                "id='" + id + '\'' +
                ", password='" + password + '\'' +
                '}';
    }
}
```

2.1.11 UStudentExample.java

```java
package com.myxh.studentmanagementsystem.bean;

import java.util.ArrayList;
import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public class UStudentExample
{
    protected String orderByClause;
    protected boolean distinct;
    protected List<Criteria> oredCriteria;

    public UStudentExample()
    {
        oredCriteria = new ArrayList<>();
    }

    public void setOrderByClause(String orderByClause)
    {
        this.orderByClause = orderByClause;
    }

    public String getOrderByClause()
    {
        return orderByClause;
    }

    public void setDistinct(boolean distinct)
    {
        this.distinct = distinct;
    }

    public boolean isDistinct()
    {
        return distinct;
    }

    public List<Criteria> getOredCriteria()
    {
        return oredCriteria;
    }
```

```
    public void or(Criteria criteria)
    {
        oredCriteria.add(criteria);
    }

    public Criteria or()
    {
        Criteria criteria = createCriteriaInternal();
        oredCriteria.add(criteria);

        return criteria;
    }

    public Criteria createCriteria()
    {
        Criteria criteria = createCriteriaInternal();

        if (oredCriteria.size() == 0)
        {
            oredCriteria.add(criteria);
        }

        return criteria;
    }

    protected Criteria createCriteriaInternal()
    {
        return new Criteria();
    }

    public void clear()
    {
        oredCriteria.clear();
        orderByClause = null;
        distinct = false;
    }

    protected abstract static class GeneratedCriteria
    {
        protected List<Criterion> criteria;

        protected GeneratedCriteria()
        {
```

```
        super();
        criteria = new ArrayList<>();
    }

    public boolean isValid()
    {
        return criteria.size() > 0;
    }

    public List<Criterion> getAllCriteria()
    {
        return criteria;
    }

    public List<Criterion> getCriteria()
    {
        return criteria;
    }

    protected void addCriterion(String condition)
    {
        if (condition == null)
        {
            throw new RuntimeException("Value for condition cannot be null");
        }

        criteria.add(new Criterion(condition));
    }

    protected void addCriterion(String condition, Object value, String property)
    {
        if (value == null)
        {
            throw new RuntimeException("Value for " + property + " cannot be null");
        }

        criteria.add(new Criterion(condition, value));
    }

    protected void addCriterion(String condition, Object value1, Object value2, String propert
    {
        if (value1 == null || value2 == null)
        {
            throw new RuntimeException("Between values for " + property + " cannot be null");
```

```
            }

            criteria.add(new Criterion(condition, value1, value2));
        }

        public Criteria andIdIsNull()
        {
            addCriterion("ID is null");

            return (Criteria) this;
        }

        public Criteria andIdIsNotNull()
        {
            addCriterion("ID is not null");

            return (Criteria) this;
        }

        public Criteria andIdEqualTo(String value)
        {
            addCriterion("ID =", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdNotEqualTo(String value)
        {
            addCriterion("ID <>", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdGreaterThan(String value)
        {
            addCriterion("ID >", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdGreaterThanOrEqualTo(String value)
        {
            addCriterion("ID >=", value, "id");
```

```
            return (Criteria) this;
        }

        public Criteria andIdLessThan(String value)
        {
            addCriterion("ID <", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdLessThanOrEqualTo(String value)
        {
            addCriterion("ID <=", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdLike(String value)
        {
            addCriterion("ID like", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdNotLike(String value)
        {
            addCriterion("ID not like", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdIn(List<String> values)
        {
            addCriterion("ID in", values, "id");

            return (Criteria) this;
        }

        public Criteria andIdNotIn(List<String> values)
        {
            addCriterion("ID not in", values, "id");

            return (Criteria) this;
        }
```

```
        public Criteria andIdBetween(String value1, String value2)
        {
            addCriterion("ID between", value1, value2, "id");

            return (Criteria) this;
        }

        public Criteria andIdNotBetween(String value1, String value2)
        {
            addCriterion("ID not between", value1, value2, "id");

            return (Criteria) this;
        }

        public Criteria andPasswordIsNull()
        {
            addCriterion("password is null");

            return (Criteria) this;
        }

        public Criteria andPasswordIsNotNull()
        {
            addCriterion("password is not null");

            return (Criteria) this;
        }

        public Criteria andPasswordEqualTo(String value)
        {
            addCriterion("password =", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordNotEqualTo(String value)
        {
            addCriterion("password <>", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordGreaterThan(String value)
```

```
        {
            addCriterion("password >", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordGreaterThanOrEqualTo(String value)
        {
            addCriterion("password >=", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordLessThan(String value)
        {
            addCriterion("password <", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordLessThanOrEqualTo(String value)
        {
            addCriterion("password <=", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordLike(String value)
        {
            addCriterion("password like", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordNotLike(String value)
        {
            addCriterion("password not like", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordIn(List<String> values)
        {
            addCriterion("password in", values, "password");
```

```
            return (Criteria) this;
        }

        public Criteria andPasswordNotIn(List<String> values)
        {
            addCriterion("password not in", values, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordBetween(String value1, String value2)
        {
            addCriterion("password between", value1, value2, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordNotBetween(String value1, String value2)
        {
            addCriterion("password not between", value1, value2, "password");

            return (Criteria) this;
        }
    }

    public static class Criteria extends GeneratedCriteria
    {
        protected Criteria()
        {
            super();
        }
    }

    public static class Criterion
    {
        private final String condition;
        private Object value;
        private Object secondValue;
        private boolean noValue;
        private boolean singleValue;
        private boolean betweenValue;
        private boolean listValue;
        private final String typeHandler;
```

```
        public String getCondition()
        {
            return condition;
        }

        public Object getValue()
        {
            return value;
        }

        public Object getSecondValue()
        {
            return secondValue;
        }

        public boolean isNoValue()
        {
            return noValue;
        }

        public boolean isSingleValue()
        {
            return singleValue;
        }

        public boolean isBetweenValue()
        {
            return betweenValue;
        }

        public boolean isListValue()
        {
            return listValue;
        }

        public String getTypeHandler()
        {
            return typeHandler;
        }

        protected Criterion(String condition)
        {
            super();
```

```
            this.condition = condition;
            this.typeHandler = null;
            this.noValue = true;
        }

        protected Criterion(String condition, Object value, String typeHandler)
        {
            super();
            this.condition = condition;
            this.value = value;
            this.typeHandler = typeHandler;

            if (value instanceof List<?>)
            {
                this.listValue = true;
            }
            else
            {
                this.singleValue = true;
            }
        }

        protected Criterion(String condition, Object value)
        {
            this(condition, value, null);
        }

        protected Criterion(String condition, Object value, Object secondValue, String typeHandler
        {
            super();
            this.condition = condition;
            this.value = value;
            this.secondValue = secondValue;
            this.typeHandler = typeHandler;
            this.betweenValue = true;
        }

        protected Criterion(String condition, Object value, Object secondValue)
        {
            this(condition, value, secondValue, null);
        }
    }
}
```

2.1.12 UTeacher.java

```java
package com.myxh.studentmanagementsystem.bean;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public class UTeacher
{
    private String id;
    private String username;
    private String password;

    public String getId()
    {
        return id;
    }

    public void setId(String id)
    {
        this.id = id == null ? null : id.trim();
    }

    public String getUsername()
    {
        return username;
    }

    public void setUsername(String username)
    {
        this.username = username == null ? null : username.trim();
    }

    public String getPassword()
    {
        return password;
    }

    public void setPassword(String password)
    {
        this.password = password == null ? null : password.trim();
    }

    @Override
    public String toString()
```

```
    {
        return "UTeacher{" +
                "id='" + id + '\'' +
                ", username='" + username + '\'' +
                ", password='" + password + '\'' +
                '}';
    }
}
```

2.1.13 UTeacherExample.java

```java
package com.myxh.studentmanagementsystem.bean;

import java.util.ArrayList;
import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public class UTeacherExample
{
    protected String orderByClause;
    protected boolean distinct;
    protected List<Criteria> oredCriteria;

    public UTeacherExample()
    {
        oredCriteria = new ArrayList<>();
    }

    public void setOrderByClause(String orderByClause)
    {
        this.orderByClause = orderByClause;
    }

    public String getOrderByClause()
    {
        return orderByClause;
    }

    public void setDistinct(boolean distinct)
    {
        this.distinct = distinct;
    }

    public boolean isDistinct()
    {
        return distinct;
    }

    public List<Criteria> getOredCriteria()
    {
        return oredCriteria;
    }
```

```
    public void or(Criteria criteria)
    {
        oredCriteria.add(criteria);
    }

    public Criteria or()
    {
        Criteria criteria = createCriteriaInternal();
        oredCriteria.add(criteria);

        return criteria;
    }

    public Criteria createCriteria()
    {
        Criteria criteria = createCriteriaInternal();

        if (oredCriteria.size() == 0)
        {
            oredCriteria.add(criteria);
        }

        return criteria;
    }

    protected Criteria createCriteriaInternal()
    {
        return new Criteria();
    }

    public void clear()
    {
        oredCriteria.clear();
        orderByClause = null;
        distinct = false;
    }

    protected abstract static class GeneratedCriteria
    {
        protected List<Criterion> criteria;

        protected GeneratedCriteria()
        {
```

```
        super();
        criteria = new ArrayList<>();
    }

    public boolean isValid()
    {
        return criteria.size() > 0;
    }

    public List<Criterion> getAllCriteria()
    {
        return criteria;
    }

    public List<Criterion> getCriteria()
    {
        return criteria;
    }

    protected void addCriterion(String condition)
    {
        if (condition == null)
        {
            throw new RuntimeException("Value for condition cannot be null");
        }

        criteria.add(new Criterion(condition));
    }

    protected void addCriterion(String condition, Object value, String property)
    {
        if (value == null)
        {
            throw new RuntimeException("Value for " + property + " cannot be null");
        }

        criteria.add(new Criterion(condition, value));
    }

    protected void addCriterion(String condition, Object value1, Object value2, String propert
    {
        if (value1 == null || value2 == null)
        {
            throw new RuntimeException("Between values for " + property + " cannot be null");
```

```
            }

            criteria.add(new Criterion(condition, value1, value2));
        }

        public Criteria andIdIsNull()
        {
            addCriterion("ID is null");

            return (Criteria) this;
        }

        public Criteria andIdIsNotNull()
        {
            addCriterion("ID is not null");

            return (Criteria) this;
        }

        public Criteria andIdEqualTo(String value)
        {
            addCriterion("ID =", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdNotEqualTo(String value)
        {
            addCriterion("ID <>", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdGreaterThan(String value)
        {
            addCriterion("ID >", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdGreaterThanOrEqualTo(String value)
        {
            addCriterion("ID >=", value, "id");
```

```
            return (Criteria) this;
        }

        public Criteria andIdLessThan(String value)
        {
            addCriterion("ID <", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdLessThanOrEqualTo(String value)
        {
            addCriterion("ID <=", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdLike(String value)
        {
            addCriterion("ID like", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdNotLike(String value)
        {
            addCriterion("ID not like", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdIn(List<String> values)
        {
            addCriterion("ID in", values, "id");

            return (Criteria) this;
        }

        public Criteria andIdNotIn(List<String> values)
        {
            addCriterion("ID not in", values, "id");

            return (Criteria) this;
        }
```

```
        public Criteria andIdBetween(String value1, String value2)
        {
            addCriterion("ID between", value1, value2, "id");

            return (Criteria) this;
        }

        public Criteria andIdNotBetween(String value1, String value2)
        {
            addCriterion("ID not between", value1, value2, "id");

            return (Criteria) this;
        }

        public Criteria andUsernameIsNull()
        {
            addCriterion("username is null");

            return (Criteria) this;
        }

        public Criteria andUsernameIsNotNull()
        {
            addCriterion("username is not null");

            return (Criteria) this;
        }

        public Criteria andUsernameEqualTo(String value)
        {
            addCriterion("username =", value, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameNotEqualTo(String value)
        {
            addCriterion("username <>", value, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameGreaterThan(String value)
```

```
        {
            addCriterion("username >", value, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameGreaterThanOrEqualTo(String value)
        {
            addCriterion("username >=", value, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameLessThan(String value)
        {
            addCriterion("username <", value, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameLessThanOrEqualTo(String value)
        {
            addCriterion("username <=", value, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameLike(String value)
        {
            addCriterion("username like", value, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameNotLike(String value)
        {
            addCriterion("username not like", value, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameIn(List<String> values)
        {
            addCriterion("username in", values, "username");
```

```
            return (Criteria) this;
        }

        public Criteria andUsernameNotIn(List<String> values)
        {
            addCriterion("username not in", values, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameBetween(String value1, String value2)
        {
            addCriterion("username between", value1, value2, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameNotBetween(String value1, String value2)
        {
            addCriterion("username not between", value1, value2, "username");

            return (Criteria) this;
        }

        public Criteria andPasswordIsNull()
        {
            addCriterion("password is null");

            return (Criteria) this;
        }

        public Criteria andPasswordIsNotNull()
        {
            addCriterion("password is not null");

            return (Criteria) this;
        }

        public Criteria andPasswordEqualTo(String value)
        {
            addCriterion("password =", value, "password");

            return (Criteria) this;
```

```
        }

        public Criteria andPasswordNotEqualTo(String value)
        {
            addCriterion("password <>", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordGreaterThan(String value)
        {
            addCriterion("password >", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordGreaterThanOrEqualTo(String value)
        {
            addCriterion("password >=", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordLessThan(String value)
        {
            addCriterion("password <", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordLessThanOrEqualTo(String value)
        {
            addCriterion("password <=", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordLike(String value)
        {
            addCriterion("password like", value, "password");

            return (Criteria) this;
        }
```

```
        public Criteria andPasswordNotLike(String value)
        {
            addCriterion("password not like", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordIn(List<String> values)
        {
            addCriterion("password in", values, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordNotIn(List<String> values)
        {
            addCriterion("password not in", values, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordBetween(String value1, String value2)
        {
            addCriterion("password between", value1, value2, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordNotBetween(String value1, String value2)
        {
            addCriterion("password not between", value1, value2, "password");

            return (Criteria) this;
        }
    }

    public static class Criteria extends GeneratedCriteria
    {
        protected Criteria()
        {
            super();
        }
    }
```

```
    public static class Criterion
    {
        private final String condition;
        private Object value;
        private Object secondValue;
        private boolean noValue;
        private boolean singleValue;
        private boolean betweenValue;
        private boolean listValue;
        private final String typeHandler;

        public String getCondition()
        {
            return condition;
        }

        public Object getValue()
        {
            return value;
        }

        public Object getSecondValue()
        {
            return secondValue;
        }

        public boolean isNoValue()
        {
            return noValue;
        }

        public boolean isSingleValue()
        {
            return singleValue;
        }

        public boolean isBetweenValue()
        {
            return betweenValue;
        }

        public boolean isListValue()
        {
            return listValue;
```

```
    }

    public String getTypeHandler()
    {
        return typeHandler;
    }

    protected Criterion(String condition)
    {
        super();
        this.condition = condition;
        this.typeHandler = null;
        this.noValue = true;
    }

    protected Criterion(String condition, Object value, String typeHandler)
    {
        super();
        this.condition = condition;
        this.value = value;
        this.typeHandler = typeHandler;

        if (value instanceof List<?>)
        {
            this.listValue = true;
        }
        else
        {
            this.singleValue = true;
        }
    }

    protected Criterion(String condition, Object value)
    {
        this(condition, value, null);
    }

    protected Criterion(String condition, Object value, Object secondValue, String typeHandler
    {
        super();
        this.condition = condition;
        this.value = value;
        this.secondValue = secondValue;
        this.typeHandler = typeHandler;
```

```
            this.betweenValue = true;
        }

        protected Criterion(String condition, Object value, Object secondValue)
        {
            this(condition, value, secondValue, null);
        }
    }
}
```

2.1.14 UtSecretary.java

```java
package com.myxh.studentmanagementsystem.bean;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public class UtSecretary
{
    private String id;
    private String username;
    private String password;

    public String getId()
    {
        return id;
    }

    public void setId(String id)
    {
        this.id = id == null ? null : id.trim();
    }

    public String getUsername()
    {
        return username;
    }

    public void setUsername(String username)
    {
        this.username = username == null ? null : username.trim();
    }

    public String getPassword()
    {
        return password;
    }

    public void setPassword(String password)
    {
        this.password = password == null ? null : password.trim();
    }
}
```

2.1.15 UtSecretaryExample.java

```java
package com.myxh.studentmanagementsystem.bean;

import java.util.ArrayList;
import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public class UtSecretaryExample
{
    protected String orderByClause;
    protected boolean distinct;
    protected List<Criteria> oredCriteria;

    public UtSecretaryExample()
    {
        oredCriteria = new ArrayList<>();
    }

    public void setOrderByClause(String orderByClause)
    {
        this.orderByClause = orderByClause;
    }

    public String getOrderByClause()
    {
        return orderByClause;
    }

    public void setDistinct(boolean distinct)
    {
        this.distinct = distinct;
    }

    public boolean isDistinct()
    {
        return distinct;
    }

    public List<Criteria> getOredCriteria()
    {
        return oredCriteria;
    }
```

```
    public void or(Criteria criteria)
    {
        oredCriteria.add(criteria);
    }

    public Criteria or()
    {
        Criteria criteria = createCriteriaInternal();
        oredCriteria.add(criteria);

        return criteria;
    }

    public Criteria createCriteria()
    {
        Criteria criteria = createCriteriaInternal();

        if (oredCriteria.size() == 0)
        {
            oredCriteria.add(criteria);
        }

        return criteria;
    }

    protected Criteria createCriteriaInternal()
    {
        return new Criteria();
    }

    public void clear()
    {
        oredCriteria.clear();
        orderByClause = null;
        distinct = false;
    }

    protected abstract static class GeneratedCriteria
    {
        protected List<Criterion> criteria;

        protected GeneratedCriteria()
        {
```

```
        super();
        criteria = new ArrayList<>();
    }

    public boolean isValid()
    {
        return criteria.size() > 0;
    }

    public List<Criterion> getAllCriteria()
    {
        return criteria;
    }

    public List<Criterion> getCriteria()
    {
        return criteria;
    }

    protected void addCriterion(String condition)
    {
        if (condition == null)
        {
            throw new RuntimeException("Value for condition cannot be null");
        }

        criteria.add(new Criterion(condition));
    }

    protected void addCriterion(String condition, Object value, String property)
    {
        if (value == null)
        {
            throw new RuntimeException("Value for " + property + " cannot be null");
        }

        criteria.add(new Criterion(condition, value));
    }

    protected void addCriterion(String condition, Object value1, Object value2, String propert
    {
        if (value1 == null || value2 == null)
        {
            throw new RuntimeException("Between values for " + property + " cannot be null");
```

```
            }

            criteria.add(new Criterion(condition, value1, value2));
        }

        public Criteria andIdIsNull()
        {
            addCriterion("ID is null");

            return (Criteria) this;
        }

        public Criteria andIdIsNotNull()
        {
            addCriterion("ID is not null");

            return (Criteria) this;
        }

        public Criteria andIdEqualTo(String value)
        {
            addCriterion("ID =", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdNotEqualTo(String value)
        {
            addCriterion("ID <>", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdGreaterThan(String value)
        {
            addCriterion("ID >", value, "id");
            return (Criteria) this;
        }

        public Criteria andIdGreaterThanOrEqualTo(String value)
        {
            addCriterion("ID >=", value, "id");

            return (Criteria) this;
```

```
        }

        public Criteria andIdLessThan(String value)
        {
            addCriterion("ID <", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdLessThanOrEqualTo(String value)
        {
            addCriterion("ID <=", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdLike(String value)
        {
            addCriterion("ID like", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdNotLike(String value)
        {
            addCriterion("ID not like", value, "id");

            return (Criteria) this;
        }

        public Criteria andIdIn(List<String> values)
        {
            addCriterion("ID in", values, "id");

            return (Criteria) this;
        }

        public Criteria andIdNotIn(List<String> values)
        {
            addCriterion("ID not in", values, "id");

            return (Criteria) this;
        }
```

```
        public Criteria andIdBetween(String value1, String value2)
        {
            addCriterion("ID between", value1, value2, "id");

            return (Criteria) this;
        }

        public Criteria andIdNotBetween(String value1, String value2)
        {
            addCriterion("ID not between", value1, value2, "id");

            return (Criteria) this;
        }

        public Criteria andUsernameIsNull()
        {
            addCriterion("username is null");

            return (Criteria) this;
        }

        public Criteria andUsernameIsNotNull()
        {
            addCriterion("username is not null");

            return (Criteria) this;
        }

        public Criteria andUsernameEqualTo(String value)
        {
            addCriterion("username =", value, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameNotEqualTo(String value)
        {
            addCriterion("username <>", value, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameGreaterThan(String value)
        {
```

```
            addCriterion("username >", value, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameGreaterThanOrEqualTo(String value)
        {
            addCriterion("username >=", value, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameLessThan(String value)
        {
            addCriterion("username <", value, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameLessThanOrEqualTo(String value)
        {
            addCriterion("username <=", value, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameLike(String value)
        {
            addCriterion("username like", value, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameNotLike(String value)
        {
            addCriterion("username not like", value, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameIn(List<String> values)
        {
            addCriterion("username in", values, "username");
```

```
            return (Criteria) this;
        }

        public Criteria andUsernameNotIn(List<String> values)
        {
            addCriterion("username not in", values, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameBetween(String value1, String value2)
        {
            addCriterion("username between", value1, value2, "username");

            return (Criteria) this;
        }

        public Criteria andUsernameNotBetween(String value1, String value2)
        {
            addCriterion("username not between", value1, value2, "username");

            return (Criteria) this;
        }

        public Criteria andPasswordIsNull()
        {
            addCriterion("password is null");

            return (Criteria) this;
        }

        public Criteria andPasswordIsNotNull()
        {
            addCriterion("password is not null");

            return (Criteria) this;
        }

        public Criteria andPasswordEqualTo(String value)
        {
            addCriterion("password =", value, "password");

            return (Criteria) this;
        }
```

```
        public Criteria andPasswordNotEqualTo(String value)
        {
            addCriterion("password <>", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordGreaterThan(String value)
        {
            addCriterion("password >", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordGreaterThanOrEqualTo(String value)
        {
            addCriterion("password >=", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordLessThan(String value)
        {
            addCriterion("password <", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordLessThanOrEqualTo(String value)
        {
            addCriterion("password <=", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordLike(String value)
        {
            addCriterion("password like", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordNotLike(String value)
```

```
        {
            addCriterion("password not like", value, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordIn(List<String> values)
        {
            addCriterion("password in", values, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordNotIn(List<String> values)
        {
            addCriterion("password not in", values, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordBetween(String value1, String value2)
        {
            addCriterion("password between", value1, value2, "password");

            return (Criteria) this;
        }

        public Criteria andPasswordNotBetween(String value1, String value2)
        {
            addCriterion("password not between", value1, value2, "password");

            return (Criteria) this;
        }
    }

    public static class Criteria extends GeneratedCriteria
    {
        protected Criteria()
        {
            super();
        }
    }

    public static class Criterion
```

```
{
    private final String condition;
    private Object value;
    private Object secondValue;
    private boolean noValue;
    private boolean singleValue;
    private boolean betweenValue;
    private boolean listValue;
    private final String typeHandler;

    public String getCondition()
    {
        return condition;
    }

    public Object getValue()
    {
        return value;
    }

    public Object getSecondValue()
    {
        return secondValue;
    }

    public boolean isNoValue()
    {
        return noValue;
    }

    public boolean isSingleValue()
    {
        return singleValue;
    }

    public boolean isBetweenValue()
    {
        return betweenValue;
    }

    public boolean isListValue()
    {
        return listValue;
    }
```

```
    public String getTypeHandler()
    {
        return typeHandler;
    }

    protected Criterion(String condition)
    {
        super();
        this.condition = condition;
        this.typeHandler = null;
        this.noValue = true;
    }

    protected Criterion(String condition, Object value, String typeHandler)
    {
        super();
        this.condition = condition;
        this.value = value;
        this.typeHandler = typeHandler;

        if (value instanceof List<?>)
        {
            this.listValue = true;
        }
        else
        {
            this.singleValue = true;
        }
    }

    protected Criterion(String condition, Object value)
    {
        this(condition, value, null);
    }

    protected Criterion(String condition, Object value, Object secondValue, String typeHandler
    {
        super();
        this.condition = condition;
        this.value = value;
        this.secondValue = secondValue;
        this.typeHandler = typeHandler;
        this.betweenValue = true;
```

```
        }

        protected Criterion(String condition, Object value, Object secondValue)
        {
            this(condition, value, secondValue, null);
        }
    }
}
```

2.2 dao 包

2.2.1 ClassInfoMapper.java

```java
package com.myxh.studentmanagementsystem.dao;

import com.myxh.studentmanagementsystem.bean.ClassInfo;
import com.myxh.studentmanagementsystem.bean.ClassInfoExample;
import org.apache.ibatis.annotations.Param;

import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public interface ClassInfoMapper
{
    long countByExample(ClassInfoExample example);

    int deleteByExample(ClassInfoExample example);

    int deleteByPrimaryKey(String className);

    int insert(ClassInfo record);

    int insertSelective(ClassInfo record);

    List<ClassInfo> selectByExample(ClassInfoExample example);

    ClassInfo selectByPrimaryKey(String className);

    int updateByExampleSelective(@Param("record") ClassInfo record, @Param("example") ClassInfoExa

    int updateByExample(@Param("record") ClassInfo record, @Param("example") ClassInfoExample exam

    int updateByPrimaryKeySelective(ClassInfo record);

    int updateByPrimaryKey(ClassInfo record);
}
```

2.2.2 StuCurriculumMapper.java

```java
package com.myxh.studentmanagementsystem.dao;

import com.myxh.studentmanagementsystem.bean.StuCurriculum;
import com.myxh.studentmanagementsystem.bean.StuCurriculumExample;
import org.apache.ibatis.annotations.Param;

import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public interface StuCurriculumMapper
{
    long countByExample(StuCurriculumExample example);

    int deleteByExample(StuCurriculumExample example);

    int deleteByPrimaryKey(Integer id);

    int insert(StuCurriculum record);

    int insertSelective(StuCurriculum record);

    List<StuCurriculum> selectByExample(StuCurriculumExample example);

    StuCurriculum selectByPrimaryKey(Integer id);

    int updateByExampleSelective(@Param("record") StuCurriculum record, @Param("example") StuCurri

    int updateByExample(@Param("record") StuCurriculum record, @Param("example") StuCurriculumExam

    int updateByPrimaryKeySelective(StuCurriculum record);

    int updateByPrimaryKey(StuCurriculum record);

    void updateCourseGradeIsNull(String courseName, String stuId);
}
```

2.2.3 StudentMapper.java

```java
package com.myxh.studentmanagementsystem.dao;

import com.myxh.studentmanagementsystem.bean.Student;
import com.myxh.studentmanagementsystem.bean.StudentExample;
import org.apache.ibatis.annotations.Param;

import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public interface StudentMapper
{
    long countByExample(StudentExample example);

    int deleteByExample(StudentExample example);

    int deleteByPrimaryKey(String id);

    int insert(Student record);

    int insertSelective(Student record);

    List<Student> selectByExample(StudentExample example);

    Student selectByPrimaryKey(String id);

    int updateByExampleSelective(@Param("record") Student record, @Param("example") StudentExample

    int updateByExample(@Param("record") Student record, @Param("example") StudentExample example)

    int updateByPrimaryKeySelective(Student record);

    int updateByPrimaryKey(Student record);

    // 查询所有专业名
    List<String> selectAllMajors();

    List<String> getClassesByMajor(String major);
}
```

2.2.4 TrainingProgramMapper.java

```java
package com.myxh.studentmanagementsystem.dao;

import com.myxh.studentmanagementsystem.bean.TrainingProgram;
import com.myxh.studentmanagementsystem.bean.TrainingProgramExample;
import org.apache.ibatis.annotations.Param;

import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public interface TrainingProgramMapper
{
    long countByExample(TrainingProgramExample example);

    int deleteByExample(TrainingProgramExample example);

    int deleteByPrimaryKey(Integer courseId);

    int insert(TrainingProgram record);

    int insertSelective(TrainingProgram record);

    List<TrainingProgram> selectByExample(TrainingProgramExample example);

    TrainingProgram selectByPrimaryKey(Integer courseId);

    int updateByExampleSelective(@Param("record") TrainingProgram record, @Param("example") Traini

    int updateByExample(@Param("record") TrainingProgram record, @Param("example") TrainingProgram

    int updateByPrimaryKeySelective(TrainingProgram record);

    int updateByPrimaryKey(TrainingProgram record);

    List<String> getCourseNamesByMajor(String major);
}
```

2.2.5 UStudentMapper.java

```java
package com.myxh.studentmanagementsystem.dao;

import com.myxh.studentmanagementsystem.bean.UStudent;
import com.myxh.studentmanagementsystem.bean.UStudentExample;
import org.apache.ibatis.annotations.Param;

import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public interface UStudentMapper
{
    long countByExample(UStudentExample example);

    int deleteByExample(UStudentExample example);

    int deleteByPrimaryKey(String id);

    int insert(UStudent record);

    int insertSelective(UStudent record);

    List<UStudent> selectByExample(UStudentExample example);

    UStudent selectByPrimaryKey(String id);

    int updateByExampleSelective(@Param("record") UStudent record, @Param("example") UStudentExamp

    int updateByExample(@Param("record") UStudent record, @Param("example") UStudentExample exampl

    int updateByPrimaryKeySelective(UStudent record);

    int updateByPrimaryKey(UStudent record);
}
```

2.2.6 UTeacherMapper.java

```java
package com.myxh.studentmanagementsystem.dao;

import com.myxh.studentmanagementsystem.bean.UTeacher;
import com.myxh.studentmanagementsystem.bean.UTeacherExample;
import org.apache.ibatis.annotations.Param;

import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public interface UTeacherMapper
{
    long countByExample(UTeacherExample example);

    int deleteByExample(UTeacherExample example);

    int deleteByPrimaryKey(String id);

    int insert(UTeacher record);

    int insertSelective(UTeacher record);

    List<UTeacher> selectByExample(UTeacherExample example);

    UTeacher selectByPrimaryKey(String id);

    int updateByExampleSelective(@Param("record") UTeacher record, @Param("example") UTeacherExamp

    int updateByExample(@Param("record") UTeacher record, @Param("example") UTeacherExample exampl

    int updateByPrimaryKeySelective(UTeacher record);

    int updateByPrimaryKey(UTeacher record);
}
```

2.2.7 UtSecretaryMapper.java

```java
package com.myxh.studentmanagementsystem.dao;

import com.myxh.studentmanagementsystem.bean.UtSecretary;
import com.myxh.studentmanagementsystem.bean.UtSecretaryExample;
import org.apache.ibatis.annotations.Param;

import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public interface UtSecretaryMapper
{
    long countByExample(UtSecretaryExample example);

    int deleteByExample(UtSecretaryExample example);

    int deleteByPrimaryKey(String id);

    int insert(UtSecretary record);

    int insertSelective(UtSecretary record);

    List<UtSecretary> selectByExample(UtSecretaryExample example);

    UtSecretary selectByPrimaryKey(String id);

    int updateByExampleSelective(@Param("record") UtSecretary record, @Param("example") UtSecretar

    int updateByExample(@Param("record") UtSecretary record, @Param("example") UtSecretaryExample

    int updateByPrimaryKeySelective(UtSecretary record);

    int updateByPrimaryKey(UtSecretary record);
}
```

2.3 controller 包

2.3.1 CurriculumController.java

```java
package com.myxh.studentmanagementsystem.controller;

import com.myxh.studentmanagementsystem.bean.Msg;
import com.myxh.studentmanagementsystem.bean.StuCurriculum;
import com.myxh.studentmanagementsystem.service.CurriculumService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.*;

import java.util.ArrayList;
import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
@Controller
public class CurriculumController
{
    @Autowired
    private CurriculumService curriculumService;

    // 获取所有已选课程(学生)
    @ResponseBody
    @RequestMapping(value = "/courses/{ID}", method = RequestMethod.GET)
    public Msg getAllSelectedCourseById(@PathVariable("ID") String ID)
    {
        List<StuCurriculum> allSelectedCourse = curriculumService.getAllSelectedCourseById(ID);

        return Msg.success().add("courses", allSelectedCourse);
    }

    // 获取必修课程(学生)
    @ResponseBody
    @RequestMapping(value = "/compulsory/{ID}", method = RequestMethod.GET)
    public Msg getCompulsoryCourseById(@PathVariable("ID") String ID)
    {
        List<StuCurriculum> compulsoryCourse = curriculumService.getCompulsoryCourseById(ID);

        return Msg.success().add("courses", compulsoryCourse);
    }

    // 获取选修课程(学生)
    @ResponseBody
```

```
    @RequestMapping(value = "/elective/{ID}", method = RequestMethod.GET)
    public Msg getElectiveCourseById(@PathVariable("ID") String ID)
    {
        List<StuCurriculum> electiveCourse = curriculumService.getElectiveCourseById(ID);

        return Msg.success().add("courses", electiveCourse);
    }

    // 获取已修课程(学生)
    @ResponseBody
    @RequestMapping(value = "/learned/{ID}", method = RequestMethod.GET)
    public Msg getLearnedCourseById(@PathVariable("ID") String ID)
    {
        List<StuCurriculum> learnedCourse = curriculumService.getLearnedCourseById(ID);

        return Msg.success().add("courses", learnedCourse);
    }

    // 获取未修课程(学生)
    @ResponseBody
    @RequestMapping(value = "/notlearned/{ID}", method = RequestMethod.GET)
    public Msg getNotLearnedCourseById(@PathVariable("ID") String ID)
    {
        List<StuCurriculum> notLearnedCourse = curriculumService.getNotLearnedCourseById(ID);

        return Msg.success().add("courses", notLearnedCourse);
    }

    // 获取有成绩的课程(学生)
    @ResponseBody
    @RequestMapping(value = "/grade/{ID}", method = RequestMethod.GET)
    public Msg getGradeById(@PathVariable("ID") String ID)
    {
        List<StuCurriculum> grade = curriculumService.getGradeById(ID);

        return Msg.success().add("grade", grade);
    }

    // 删除课程(学生)
    @ResponseBody
    @RequestMapping(value = "/deleteCourse/{id}", method = RequestMethod.DELETE)
    public Msg deleteSelectedCourseById(@PathVariable("id") Integer id)
    {
        curriculumService.deleteSelectedCourseById(id);
```

```
        return Msg.success();
    }

    // 获取未授课程名和已授课程名(未打成绩)(教师)
    @ResponseBody
    @RequestMapping(value = "/unTeachedCourses/{courseNames}", method = RequestMethod.GET)
    public Msg getCourseNames(@PathVariable("courseNames") String allCourses)
    {
        List<String> unTeachedCourseNames = new ArrayList<>();
        List<String> teachedCourseNames = new ArrayList<>();
        String[] allCourseNames = allCourses.split("-");

        for (String allCourseName : allCourseNames)
        {
            boolean b = curriculumService.isUnTeachedCourse(allCourseName);
            boolean b1 = curriculumService.isTeachedCourse(allCourseName);

            if (b)
            {
                unTeachedCourseNames.add(allCourseName);
            }
            else if (b1)
            {
                teachedCourseNames.add(allCourseName);
            }
        }

        return Msg.success().add("unTeachedCourseNames", unTeachedCourseNames).add("teachedCourseN
    }

    // 获取指定课程列表(教师、教秘)
    @ResponseBody
    @RequestMapping(value = "/course/{courseName}", method = RequestMethod.GET)
    public Msg getCourseByCourseName(@PathVariable("courseName") String courseName)
    {
        List<StuCurriculum> courseList = curriculumService.getCourseByCourseName(courseName);

        return Msg.success().add("courseList", courseList);
    }

    // 更新课程成绩或课程状态(教师、教秘)
    @ResponseBody
    @RequestMapping(value = "/updateGrade/{updateGradeData}", method = RequestMethod.PUT)
```

```
public Msg updateGrade(@PathVariable("updateGradeData") String updateGradeData)
{
    String[] strings = updateGradeData.split("-");
    StuCurriculum curriculum;

    if (strings.length == 3)
    {
        if (strings[2].equals("未修") || strings[2].equals("已修"))
        {
            curriculum = new StuCurriculum(null, null, strings[0], null, strings[1], null, str
        }
        else
        {
            Integer grade = Integer.parseInt(strings[2]);
            curriculum = new StuCurriculum(null, null, strings[0], null, strings[1], null, nul
        }

        curriculumService.updateGradeByCourseNameAndStuId(curriculum);
    }
    else if (strings.length == 4)
    {
        if (strings[3].equals("00"))
        {
            curriculumService.updateCourseGradeIsNull(strings[0], strings[1]);
        }
        else
        {
            Integer grade = Integer.parseInt(strings[3]);
            curriculum = new StuCurriculum(null, null, strings[0], null, strings[1], null, str
            curriculumService.updateGradeByCourseNameAndStuId(curriculum);
        }
    }

    return Msg.success();
}

// 更新课程为已修(教师)
@ResponseBody
@RequestMapping(value = "/updateCourseStatus/{courseName}", method = RequestMethod.PUT)
public Msg updateCourseStatusByCourseName(@PathVariable("courseName") String courseName)
{
    curriculumService.updateCourseStatusByCourseName(courseName);

    return Msg.success();
```

```
    }

    // 获取按成绩排序后指定课程(教师、教秘)
    @ResponseBody
    @RequestMapping(value = "/sortCourse/{courseName}", method = RequestMethod.GET)
    public Msg getSortCourseByCourseName(@PathVariable("courseName") String courseName)
    {
        List<StuCurriculum> courseList = curriculumService.getSortCourseByCourseName(courseName);

        return Msg.success().add("courseList", courseList);
    }
}
```

2.3.2 LoginController.java

```java
package com.myxh.studentmanagementsystem.controller;

import com.myxh.studentmanagementsystem.bean.*;
import com.myxh.studentmanagementsystem.service.LoginService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

/**
 * @author MYXH
 * @date 2023/6/4
 */
@Controller
public class LoginController
{
    @Autowired
    private LoginService loginService;

    // 学生登录
    @ResponseBody
    @PostMapping("/stu")
    public Msg studentLogin(UStudent uStu, HttpSession httpSession)
    {
        UStudent uStudent = loginService.getUserInfo(uStu.getId());

        if (uStudent != null)
        {
            if (uStudent.getPassword().equals(uStu.getPassword()))
            {
                Student student = loginService.getUsernameById(uStu.getId());
                httpSession.setAttribute("user", student);

                return Msg.success();
            }
            else
            {
                return Msg.fail().add("msg", "密码错误");
            }
        }
```

```
        return Msg.fail().add("msg", "用户不存在");
    }

    // 教师登录
    @ResponseBody
    @PostMapping("/tea")
    public Msg teacherLogin(UTeacher uTea, HttpSession httpSession)
    {
        UTeacher uTeacher = loginService.getTeacherInfo(uTea.getId());

        if (uTeacher != null)
        {
            if (uTeacher.getPassword().equals(uTea.getPassword()))
            {
                httpSession.setAttribute("user", uTeacher);
                return Msg.success();
            }
            else
            {
                return Msg.fail().add("msg", "密码错误");
            }
        }

        return Msg.fail().add("msg", "用户不存在");
    }

    // 教秘登录
    @ResponseBody
    @PostMapping("/sec")
    public Msg secretaryLogin(UtSecretary uSec, HttpSession httpSession)
    {
        UtSecretary uSecretary = loginService.getSecretaryInfo(uSec.getId());

        if (uSecretary != null)
        {
            if (uSecretary.getPassword().equals(uSec.getPassword()))
            {
                httpSession.setAttribute("user", uSecretary);
                return Msg.success();
            }
            else
            {
                return Msg.fail().add("msg", "密码错误");
```

```
            }
        }

        return Msg.fail().add("msg", "用户不存在");
    }

    // 跳转到学生教务系统主页
    @RequestMapping("/stuIndex")
    public String toStuIndex()
    {
        return "stu_index";
    }

    // 跳转到教师教务系统主页
    @RequestMapping("/teaIndex")
    public String toTeaIndex()
    {
        return "tea_index";
    }

    // 跳转到教秘教务系统主页
    @RequestMapping("/secIndex")
    public String toSecIndex()
    {
        return "sec_index";
    }

    // 注销
    @RequestMapping("/logout")
    public String logout(HttpServletRequest request)
    {
        request.getSession().invalidate();

        return "login";
    }
}
```

2.3.3 OtherController.java

```java
package com.myxh.studentmanagementsystem.controller;

import com.myxh.studentmanagementsystem.bean.Msg;
import com.myxh.studentmanagementsystem.service.OtherService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
@Controller
public class OtherController
{
    @Autowired
    private OtherService otherService;

    // 获取所有专业名
    @ResponseBody
    @RequestMapping(value = "/majors", method = RequestMethod.GET)
    public Msg getAllMajors()
    {
        List<String> majors = otherService.getAllMajors();

        return Msg.success().add("majors", majors);
    }

    // 通过专业获取所有班级
    @ResponseBody
    @RequestMapping(value = "/classes/{major}", method = RequestMethod.GET)
    public Msg getClassesByMajor(@PathVariable("major") String major)
    {
        List<String> classes = otherService.getClassesByMajor(major);

        return Msg.success().add("classes", classes);
    }
```

```java
    // 获取指定班级的学号
    @ResponseBody
    @RequestMapping(value = "/stus/{student_ids}", method = RequestMethod.GET)
    public Msg getStudentIdsByClass(@PathVariable("student_ids") String student_ids)
    {
        String[] strings = student_ids.split("-");

        ArrayList<String> list = new ArrayList<>(Arrays.asList(strings).subList(1, strings.length));

        List<String> ids = otherService.getStudentIdsByClass(strings[0], list);

        return Msg.success().add("ids", ids);
    }
}
```

2.3.4 TrainProgramController.java

```java
package com.myxh.studentmanagementsystem.controller;

import com.myxh.studentmanagementsystem.bean.Msg;
import com.myxh.studentmanagementsystem.bean.StuCurriculum;
import com.myxh.studentmanagementsystem.bean.Student;
import com.myxh.studentmanagementsystem.bean.TrainingProgram;
import com.myxh.studentmanagementsystem.service.CurriculumService;
import com.myxh.studentmanagementsystem.service.TrainingProgramService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;

import javax.servlet.http.HttpSession;
import java.util.ArrayList;
import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
@Controller
public class TrainProgramController
{
    @Autowired
    private TrainingProgramService trainingProgramService;
    @Autowired
    private CurriculumService curriculumService;

    // 根据专业获取培养方案
    @ResponseBody
    @RequestMapping(value = "/train/{major}", method = RequestMethod.GET)
    public Msg getTrainProgramByMajor(@PathVariable("major") String major, HttpSession session)
    {
        List<TrainingProgram> trainingPrograms = trainingProgramService.getTrainProgramByMajor(maj
        Student student = (Student) session.getAttribute("user");
        List<StuCurriculum> allSelectedCourse = curriculumService.getAllSelectedCourseById(student
        ArrayList<Integer> list = new ArrayList<>();

        for (StuCurriculum selectedCourse : allSelectedCourse)
        {
            list.add(selectedCourse.getCourseId());
```

```
        }

        return Msg.success().add("trainingPrograms", trainingPrograms).add("selectedCourseId", lis
    }

    // 从培养方案中添加课程
    @ResponseBody
    @RequestMapping(value = "/addCourse/{ids}", method = RequestMethod.POST)
    public Msg addCourseByCourseIds(@PathVariable("ids") String ids, HttpSession session)
    {
        String[] idS = ids.split("-");
        List<Integer> list = new ArrayList<>();

        for (String id : idS)
        {
            list.add(Integer.parseInt(id));
        }

        Student student = (Student) session.getAttribute("user");
        List<TrainingProgram> courses = trainingProgramService.getTrainProgramByCourseIds(list);

        for (TrainingProgram course : courses)
        {
            StuCurriculum curriculum = new StuCurriculum(null, course.getCourseId(), course.getCou
                    course.getCourseNature(), student.getId(), student.getStudentName(), "未修", n
            curriculumService.saveCourse(curriculum);
        }

        return Msg.success();
    }

    // 获取教授的课程(教师)
    @ResponseBody
    @RequestMapping(value = "/teachingCourses/{name}", method = RequestMethod.GET)
    public Msg getTeachingCoursesByName(@PathVariable("name") String teacherName)
    {
        List<TrainingProgram> teachingCourses = trainingProgramService.getTeachingCoursesByName(te

        return Msg.success().add("teachingCourses", teachingCourses);
    }

    // 修改课程属性(教师)
    @ResponseBody
    @RequestMapping(value = "/updateCourse/{update_data}", method = RequestMethod.PUT)
```

```
    public Msg updateTeachingCoursesById(@PathVariable("update_data") String data)
    {
        String[] infos = data.split("-");
        Integer course_id = Integer.parseInt(infos[0]);
        TrainingProgram course = new TrainingProgram(course_id, null, null, infos[1], null, infos[
        trainingProgramService.updateTeachingCourse(course);

        return Msg.success();
    }

    // 获取所有培养方案(教秘)
    @ResponseBody
    @RequestMapping(value = "/trains", method = RequestMethod.GET)
    public Msg getAllTrainProgram()
    {
        List<TrainingProgram> allTrainProgram = trainingProgramService.getAllTrainProgram();

        return Msg.success().add("allTrainProgram", allTrainProgram);
    }

    // 添加培养方案(教秘)
    @ResponseBody
    @RequestMapping(value = "/train", method = RequestMethod.POST)
    public Msg addTrainProgram(TrainingProgram trainingProgram)
    {
        trainingProgramService.addTrainProgram(trainingProgram);

        return Msg.success();
    }

    // 修改培养方案(教秘)
    @ResponseBody
    @RequestMapping(value = "/train/{courseId}", method = RequestMethod.PUT)
    public Msg UpdateTrainProgram(TrainingProgram trainingProgram, @PathVariable("courseId") Integ
    {
        trainingProgram.setCourseId(courseId);
        trainingProgramService.UpdateTrainProgram(trainingProgram);

        return Msg.success();
    }

    // 删除培养方案(教秘)
    @ResponseBody
    @RequestMapping(value = "/train/{courseId}", method = RequestMethod.DELETE)
```

```
    public Msg deleteTrainProgramByCourseId(@PathVariable("courseId") Integer courseId)
    {
        trainingProgramService.deleteTrainProgramByCourseId(courseId);

        return Msg.success();
    }

    // 根据专业获取所有课程名
    @ResponseBody
    @RequestMapping(value = "/courseNames/{major}", method = RequestMethod.GET)
    public Msg getCourseNamesByMajor(@PathVariable("major") String major)
    {
        List<String> courseNames = trainingProgramService.getCourseNamesByMajor(major);

        return Msg.success().add("courseNames", courseNames);
    }
}
```

2.4 filter 包

2.4.1 LoginFilter.java

```java
package com.myxh.studentmanagementsystem.filter;

import javax.servlet.*;
import javax.servlet.http.HttpServletRequest;
import java.io.IOException;

/**
 * @author MYXH
 * @date 2023/6/4
 */
public class LoginFilter implements Filter
{
    @Override
    public void doFilter(ServletRequest servletRequest, ServletResponse servletResponse, FilterCha
    {
        HttpServletRequest req = (HttpServletRequest) servletRequest;
        Object user = req.getSession().getAttribute("user");
        String spath = req.getServletPath();

        // 不需要过滤的 url
        String[] urls = {"/stu", "/tea", "/sec", "/logout", "/login", "/json", ".js", ".css", ".ic
        boolean flag = true;

        for (String str : urls)
        {
            if (spath.contains(str))
            {
                flag = false;
                break;
            }
        }

        if (flag)
        {
            if (user == null)
            {
                req.getRequestDispatcher("/WEB-INF/pages/login.jsp").forward(servletRequest, servl
            }
            else
            {
                filterChain.doFilter(servletRequest, servletResponse);
            }
        }
        else
```

```
        {
            filterChain.doFilter(servletRequest, servletResponse);
        }
    }
}
```

2.5 service 包

2.5.1 CurriculumService.java

```java
package com.myxh.studentmanagementsystem.service;

import com.myxh.studentmanagementsystem.bean.StuCurriculum;
import com.myxh.studentmanagementsystem.bean.StuCurriculumExample;
import com.myxh.studentmanagementsystem.dao.StuCurriculumMapper;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
@Service
public class CurriculumService
{
    @Autowired
    private StuCurriculumMapper stuCurriculumMapper;

    public List<StuCurriculum> getAllSelectedCourseById(String id)
    {
        StuCurriculumExample example = new StuCurriculumExample();
        example.createCriteria().andCourseSelectedIdEqualTo(id);

        return stuCurriculumMapper.selectByExample(example);
    }

    public List<StuCurriculum> getCompulsoryCourseById(String id)
    {
        StuCurriculumExample example = new StuCurriculumExample();
        example.createCriteria().andCourseSelectedIdEqualTo(id).andCourseNatureEqualTo("必修课");

        return stuCurriculumMapper.selectByExample(example);
    }

    public List<StuCurriculum> getElectiveCourseById(String id)
    {
        StuCurriculumExample example = new StuCurriculumExample();
        example.createCriteria().andCourseSelectedIdEqualTo(id).andCourseNatureEqualTo("选修课");

        return stuCurriculumMapper.selectByExample(example);
    }
```

```
    public List<StuCurriculum> getLearnedCourseById(String id)
    {
        StuCurriculumExample example = new StuCurriculumExample();
        example.createCriteria().andCourseSelectedIdEqualTo(id).andCourseStatusEqualTo("已修");

        return stuCurriculumMapper.selectByExample(example);
    }

    public List<StuCurriculum> getNotLearnedCourseById(String id)
    {
        StuCurriculumExample example = new StuCurriculumExample();
        example.createCriteria().andCourseSelectedIdEqualTo(id).andCourseStatusEqualTo("未修");

        return stuCurriculumMapper.selectByExample(example);
    }

    public List<StuCurriculum> getGradeById(String id)
    {
        StuCurriculumExample example = new StuCurriculumExample();
        example.createCriteria().andCourseSelectedIdEqualTo(id).andGradeIsNotNull();

        return stuCurriculumMapper.selectByExample(example);
    }

    public void saveCourse(StuCurriculum curriculum)
    {
        stuCurriculumMapper.insert(curriculum);
    }

    public void deleteSelectedCourseById(Integer id)
    {
        stuCurriculumMapper.deleteByPrimaryKey(id);
    }

    public boolean isUnTeachedCourse(String courseName)
    {
        StuCurriculumExample example = new StuCurriculumExample();
        example.createCriteria().andCourseNameEqualTo(courseName).andCourseStatusEqualTo("未修");
        List<StuCurriculum> courses = stuCurriculumMapper.selectByExample(example);

        return courses.size() != 0;
    }

    public boolean isTeachedCourse(String courseName)
```

```
    {
        StuCurriculumExample example = new StuCurriculumExample();
        example.createCriteria().andCourseNameEqualTo(courseName).andCourseStatusEqualTo("已修");
        List<StuCurriculum> courses = stuCurriculumMapper.selectByExample(example);

        return courses.size() != 0;
    }

    public List<StuCurriculum> getCourseByCourseName(String courseName)
    {
        StuCurriculumExample example = new StuCurriculumExample();
        example.createCriteria().andCourseNameEqualTo(courseName);

        return stuCurriculumMapper.selectByExample(example);
    }

    public void updateGradeByCourseNameAndStuId(StuCurriculum curriculum)
    {
        StuCurriculumExample example = new StuCurriculumExample();
        example.createCriteria().andCourseNameEqualTo(curriculum.getCourseName()).andCourseSelecte
        stuCurriculumMapper.updateByExampleSelective(curriculum, example);
    }

    public void updateCourseStatusByCourseName(String courseName)
    {
        StuCurriculumExample example = new StuCurriculumExample();
        example.createCriteria().andCourseNameEqualTo(courseName);
        StuCurriculum curriculum = new StuCurriculum(null, null, null, null, null, null, "已修", n
        stuCurriculumMapper.updateByExampleSelective(curriculum, example);
    }

    public void updateCourseGradeIsNull(String courseName, String stuId)
    {
        stuCurriculumMapper.updateCourseGradeIsNull(courseName, stuId);
    }

    public List<StuCurriculum> getSortCourseByCourseName(String courseName)
    {
        StuCurriculumExample example = new StuCurriculumExample();
        example.createCriteria().andCourseNameEqualTo(courseName);
        example.setOrderByClause("grade desc");

        return stuCurriculumMapper.selectByExample(example);
    }
```

```
}
```

2.5.2 LoginService.java

```java
package com.myxh.studentmanagementsystem.service;

import com.myxh.studentmanagementsystem.bean.Student;
import com.myxh.studentmanagementsystem.bean.UStudent;
import com.myxh.studentmanagementsystem.bean.UTeacher;
import com.myxh.studentmanagementsystem.bean.UtSecretary;
import com.myxh.studentmanagementsystem.dao.StudentMapper;
import com.myxh.studentmanagementsystem.dao.UStudentMapper;
import com.myxh.studentmanagementsystem.dao.UTeacherMapper;
import com.myxh.studentmanagementsystem.dao.UtSecretaryMapper;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

/**
 * @author MYXH
 * @date 2023/6/4
 */
@Service
public class LoginService
{
    @Autowired
    private UStudentMapper uStudentMapper;
    @Autowired
    private UTeacherMapper uTeacherMapper;
    @Autowired
    private UtSecretaryMapper utSecretaryMapper;
    @Autowired
    private StudentMapper studentMapper;

    public UStudent getUserInfo(String id)
    {
        return uStudentMapper.selectByPrimaryKey(id);
    }

    public Student getUsernameById(String id)
    {
        return studentMapper.selectByPrimaryKey(id);
    }

    public UTeacher getTeacherInfo(String id)
    {
        return uTeacherMapper.selectByPrimaryKey(id);
    }
```

```
    public UtSecretary getSecretaryInfo(String id)
    {
        return utSecretaryMapper.selectByPrimaryKey(id);
    }
}
```

2.5.3 OtherService.java

```java
package com.myxh.studentmanagementsystem.service;

import com.myxh.studentmanagementsystem.bean.Student;
import com.myxh.studentmanagementsystem.bean.StudentExample;
import com.myxh.studentmanagementsystem.dao.StudentMapper;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.ArrayList;
import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
@Service
public class OtherService
{
    @Autowired
    private StudentMapper studentMapper;

    public List<String> getAllMajors()
    {
        return studentMapper.selectAllMajors();
    }

    public List<String> getClassesByMajor(String major)
    {
        return studentMapper.getClassesByMajor(major);
    }

    public List<String> getStudentIdsByClass(String className, ArrayList<String> list)
    {
        StudentExample example = new StudentExample();
        example.createCriteria().andClassNameEqualTo(className).andIdIn(list);
        List<Student> students = studentMapper.selectByExample(example);
        ArrayList<String> ids = new ArrayList<>();

        for (Student student : students)
        {
            ids.add(student.getId());
        }

        return ids;
```

```
    }
}
```

2.5.4 TrainingProgramService.java

```java
package com.myxh.studentmanagementsystem.service;

import com.myxh.studentmanagementsystem.bean.TrainingProgram;
import com.myxh.studentmanagementsystem.bean.TrainingProgramExample;
import com.myxh.studentmanagementsystem.dao.TrainingProgramMapper;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.ArrayList;
import java.util.List;

/**
 * @author MYXH
 * @date 2023/6/4
 */
@Service
public class TrainingProgramService
{
    @Autowired
    private TrainingProgramMapper trainingProgramMapper;

    public List<TrainingProgram> getTrainProgramByMajor(String major)
    {
        List<String> list = new ArrayList<>();
        list.add("必修课");
        list.add(major);
        TrainingProgramExample example = new TrainingProgramExample();
        example.createCriteria().andCourseMajorIn(list);

        return trainingProgramMapper.selectByExample(example);
    }

    public List<TrainingProgram> getTrainProgramByCourseIds(List<Integer> ids)
    {
        TrainingProgramExample example = new TrainingProgramExample();
        example.createCriteria().andCourseIdIn(ids);

        return trainingProgramMapper.selectByExample(example);
    }

    public List<TrainingProgram> getTeachingCoursesByName(String teacherName)
    {
        TrainingProgramExample example = new TrainingProgramExample();
        example.createCriteria().andCourseTeacherEqualTo(teacherName);
```

```
        return trainingProgramMapper.selectByExample(example);
    }

    public void updateTeachingCourse(TrainingProgram course)
    {
        trainingProgramMapper.updateByPrimaryKeySelective(course);
    }

    public List<TrainingProgram> getAllTrainProgram()
    {
        return trainingProgramMapper.selectByExample(null);
    }

    public void addTrainProgram(TrainingProgram trainingProgram)
    {
        trainingProgramMapper.insertSelective(trainingProgram);
    }

    public void UpdateTrainProgram(TrainingProgram trainingProgram)
    {
        trainingProgramMapper.updateByPrimaryKeySelective(trainingProgram);
    }

    public void deleteTrainProgramByCourseId(Integer courseId)
    {
        trainingProgramMapper.deleteByPrimaryKey(courseId);
    }

    public List<String> getCourseNamesByMajor(String major)
    {
        return trainingProgramMapper.getCourseNamesByMajor(major);
    }
}
```

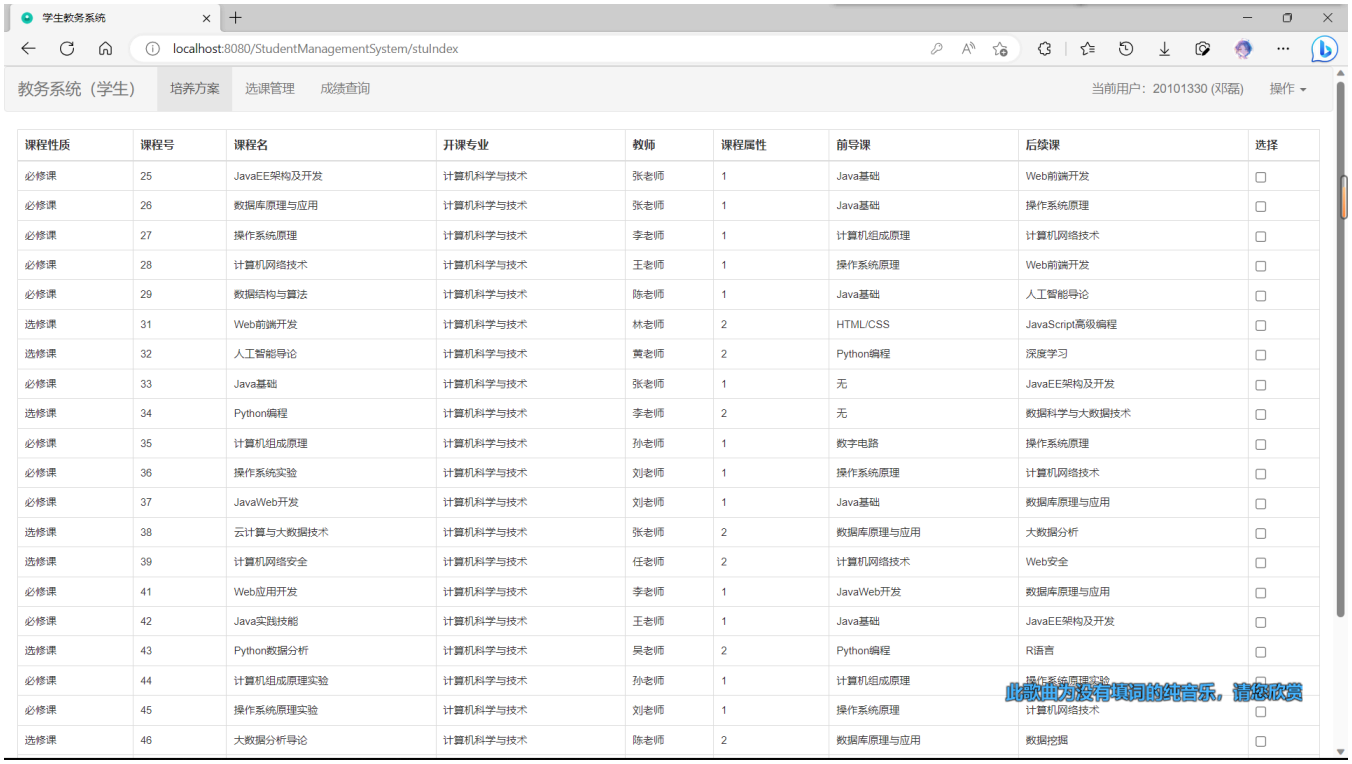
3. 系统界面展示

以下是本学生管理系统的几个主要界面截图：

3.1 登录界面

用户可以在此界面输入用户名和密码进行登录，只有在登录成功后才能进入主界面进行操作。

3.2 学生主界面



课程性质	课程号	课程名	开课专业	教师	课程属性	前导课	后续课	选择
必修课	25	JavaEE架构及开发	计算机科学与技术	张老师	1	Java基础	Web前端开发	☐
必修课	26	数据库原理与应用	计算机科学与技术	张老师	1	Java基础	操作系统原理	☐
必修课	27	操作系统原理	计算机科学与技术	李老师	1	计算机组成原理	计算机网络技术	☐
必修课	28	计算机网络技术	计算机科学与技术	王老师	1	操作系统原理	Web前端开发	☐
必修课	29	数据结构与算法	计算机科学与技术	陈老师	1	Java基础	人工智能导论	☐
选修课	31	Web前端开发	计算机科学与技术	林老师	2	HTML/CSS	JavaScript高级编程	☐
选修课	32	人工智能导论	计算机科学与技术	黄老师	2	Python编程	深度学习	☐
必修课	33	Java基础	计算机科学与技术	张老师	1	无	JavaEE架构及开发	☐
选修课	34	Python编程	计算机科学与技术	李老师	2	无	数据科学与大数据技术	☐
必修课	35	计算机组成原理	计算机科学与技术	孙老师	1	数字电路	操作系统原理	☐
必修课	36	操作系统实验	计算机科学与技术	刘老师	1	操作系统原理	计算机网络技术	☐
必修课	37	JavaWeb开发	计算机科学与技术	刘老师	1	Java基础	数据库原理与应用	☐
选修课	38	云计算与大数据技术	计算机科学与技术	张老师	2	数据库原理与应用	大数据分析	☐
选修课	39	计算机网络安全	计算机科学与技术	任老师	2	计算机网络技术	Web安全	☐
必修课	41	Web应用开发	计算机科学与技术	李老师	1	JavaWeb开发	数据库原理与应用	☐
必修课	42	Java实践技能	计算机科学与技术	王老师	1	Java基础	JavaEE架构及开发	☐
选修课	43	Python数据分析	计算机科学与技术	吴老师	2	Python编程	R语言	☐
必修课	44	计算机组成原理实验	计算机科学与技术	孙老师	1	计算机组成原理	操作系统原理实验	☐
必修课	45	操作系统原理实验	计算机科学与技术	刘老师	1	操作系统原理	计算机网络技术	☐
选修课	46	大数据分析导论	计算机科学与技术	陈老师	2	数据库原理与应用	数据挖掘	☐

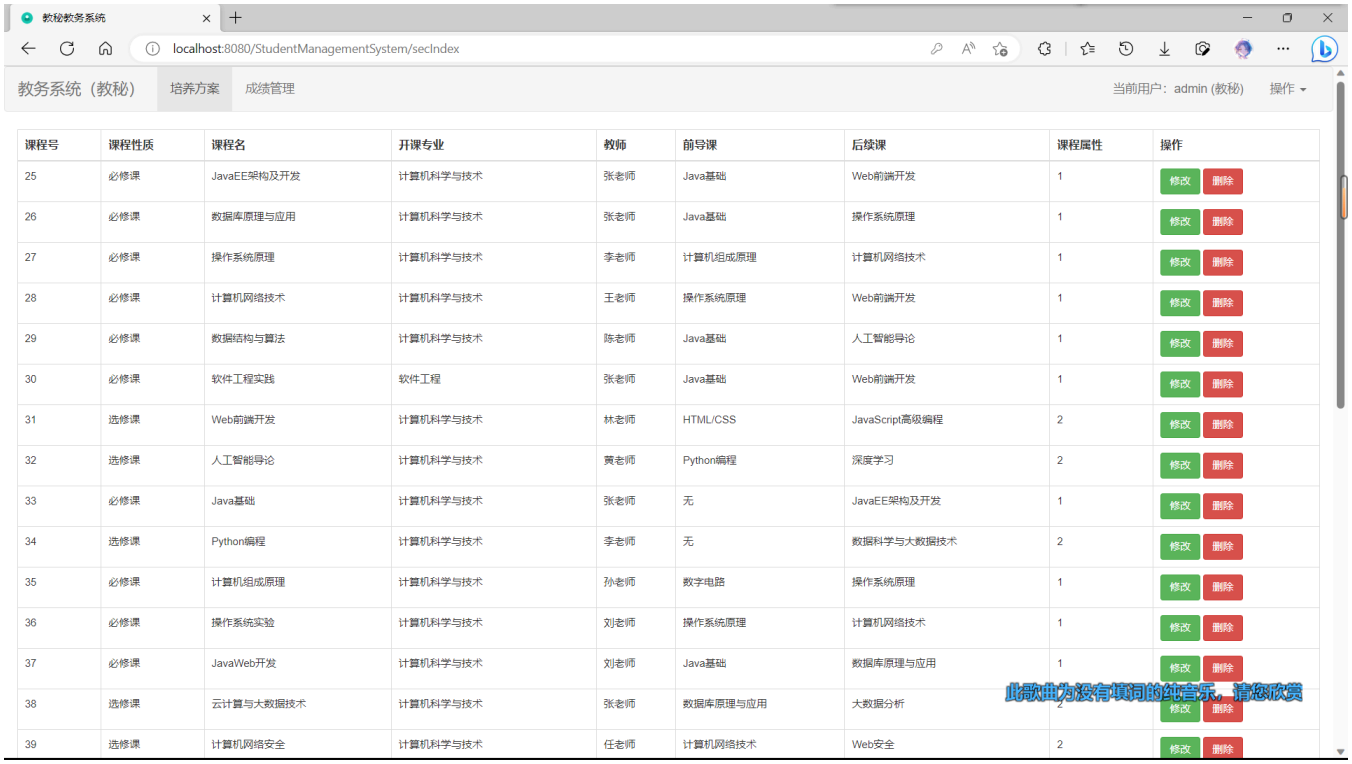
学生可以通过该界面进行课程添加、删除和成绩查询等操作。

3.3 教师主界面

课程号	课程性质	课程名	开课专业(可修改)	前导课(可修改)	后续课(可修改)	课程属性(可修改)	操作
25	JavaEE架构及开发	必修课	计算机科学与技术	Java基础	Web前端开发	1	修改
26	数据库原理与应用	必修课	计算机科学与技术	Java基础	操作系统原理	1	修改
30	软件工程实践	必修课	软件工程	Java基础	Web前端开发	1	修改
33	Java基础	必修课	计算机科学与技术	无	JavaEE架构及开发	1	修改
38	云计算与大数据技术	选修课	计算机科学与技术	数据库原理与应用	大数据分析	2	修改
52	移动开发	选修课	软件工程	JavaWeb开发	Android开发	2	修改

教师可以通过该界面进行课程属性修改、成绩管理和已授课程成绩查询等操作。

3.4 教务处主界面



课程号	课程性质	课程名	开课专业	教师	前导课	后续课	课程属性	操作
25	必修课	JavaEE架构及开发	计算机科学与技术	张老师	Java基础	Web前端开发	1	修改 删除
26	必修课	数据库原理与应用	计算机科学与技术	张老师	Java基础	操作系统原理	1	修改 删除
27	必修课	操作系统原理	计算机科学与技术	李老师	计算机组成原理	计算机网络技术	1	修改 删除
28	必修课	计算机网络技术	计算机科学与技术	王老师	操作系统原理	Web前端开发	1	修改 删除
29	必修课	数据结构与算法	计算机科学与技术	陈老师	Java基础	人工智能导论	1	修改 删除
30	必修课	软件工程实践	软件工程	张老师	Java基础	Web前端开发	1	修改 删除
31	选修课	Web前端开发	计算机科学与技术	林老师	HTML/CSS	JavaScript高级编程	2	修改 删除
32	选修课	人工智能导论	计算机科学与技术	黄老师	Python编程	深度学习	2	修改 删除
33	必修课	Java基础	计算机科学与技术	张老师	无	JavaEE架构及开发	1	修改 删除
34	选修课	Python编程	计算机科学与技术	李老师	无	数据科学与大数据技术	2	修改 删除
35	必修课	计算机组成原理	计算机科学与技术	孙老师	数字电路	操作系统原理	1	修改 删除
36	必修课	操作系统实验	计算机科学与技术	刘老师	操作系统原理	计算机网络技术	1	修改 删除
37	必修课	JavaWeb开发	计算机科学与技术	刘老师	Java基础	数据库原理与应用	1	修改 删除
38	选修课	云计算与大数据技术	计算机科学与技术	张老师	数据库原理与应用	大数据分析	2	修改 删除
39	选修课	计算机网络安全	计算机科学与技术	任老师	计算机网络技术	Web安全	2	修改 删除

教务处可以通过该界面进行培养方案管理、成绩管理、信息查询和成绩排名等操作。

4. 总结

本文档详细描述了一个学生管理系统的开发过程和功能实现，包括系统目标、系统设计、系统实现、系统部署和界面展示等内容。该系统采用了 SSM（Spring+SpringMVC+MyBatis）架构，使用了 Java 编程语言进行 Web 应用程序的开发，使用了 MySQL 数据库和 Maven 构建工具。该系统实现了登录、课程添加、课程删除、成绩查询、课程属性修改、成绩管理、已授课程成绩查询、培养方案管理、成绩管理、信息查询和成绩排名等功能，并提供了良好的用户界面。同时，本文档还提供了系统相关代码和界面截图以供参考。