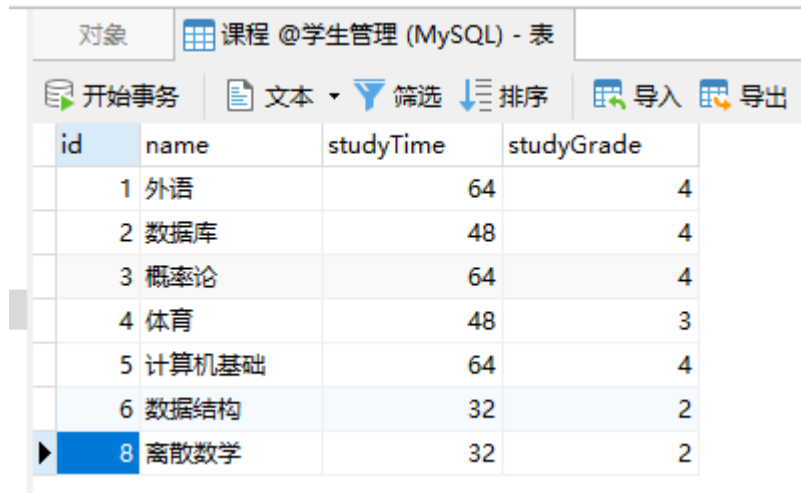


# Java 实现图形化课程管理系统

## 一、描述

1. 数据库“学生管理”中“课程”表如下：



| id | name  | studyTime | studyGrade |
|----|-------|-----------|------------|
| 1  | 外语    | 64        | 4          |
| 2  | 数据库   | 48        | 4          |
| 3  | 概率论   | 64        | 4          |
| 4  | 体育    | 48        | 3          |
| 5  | 计算机基础 | 64        | 4          |
| 6  | 数据结构  | 32        | 2          |
| 8  | 离散数学  | 32        | 2          |

2. 用 java 设计一个应用程序，实现对“课程”表的增、删、改、查。

3. 说明文档中有 java 连接数据库的关键代码说明。

要求：

可以运行，实现对学生课程数据库的操作；

## 二、实现过程及运行结果

1. 确定实验操作工具，实验采用 java 语言，使用 IntelliJ Idea 作为编译器，MYSQL 来存储数据库，c3p0 作为连接池

2. 配置 c3p0-config.xml 文件，设置要访问的数据库名和登录用户以及密码

```
<?xml version="1.0" encoding="UTF-8" ?>
<c3p0-config>
  <!-- 默认配置，如果没有指定则使用这个配置 -->
  <default-config>
    <property name="driverClass">com.mysql.cj.jdbc.Driver</property>
    <property name="jdbcUrl">jdbc:mysql://localhost:3306/学生管理?useSSL=false&serverTimezone=UTC</property>
    <property name="user">root</property>
    <property name="password">123456</property>
    <!-- 初始化池大小 -->
    <property name="initialPoolSize">10</property>
    <!-- 最大空闲时间 -->
    <property name="maxIdleTime">30</property>
    <!-- 最多有多少个连接 -->
    <property name="maxPoolSize">15</property>
    <!-- 最少几个连接 -->
    <property name="minPoolSize">5</property>
    <!-- 每次最多可以执行多少个批处理语句 -->
    <property name="maxStatements">50</property>
  </default-config>
  <!-- 命名的配置 -->
  <named-config name="51zixue"><!-- 这里是设置配置文件的名字 -->
    <property name="driverClass">com.mysql.cj.jdbc.Driver</property>
    <property name="jdbcUrl">jdbc:mysql://localhost:3306/学生管理?useSSL=false&serverTimezone=UTC</property>
    <property name="user">root</property><!-- mysql的登陆名 -->
    <property name="password">123456</property><!-- 如果没密码就可以设置成 <property name="password"></property> -->
    <property name="acquireIncrement">5</property><!-- 如果池中数据连接不够时一次增长多少个 -->
    <property name="initialPoolSize">10</property>
    <property name="minPoolSize">5</property>
    <property name="maxPoolSize">15</property>
    <property name="maxStatements">0</property>
    <property name="maxStatementsPerConnection">5</property> <!-- he's important, but there's only one of him -->
  </named-config>
</c3p0-config>
```

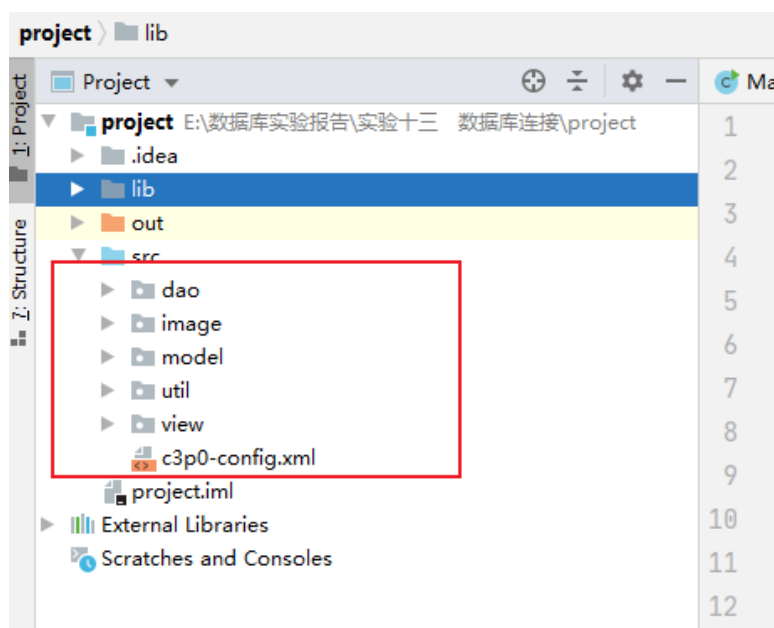
mysql 8.0及以上 加载驱动

数据库名

登录数据库用户及密码

### 3. 构造大致框架

- (1) dao 层：访问数据库的业务逻辑层
- (2) util 层：连接数据库
- (3) image 文件夹：存放窗口背景图片
- (3) model 层：存放对象类
- (4) view 层：使用面板，展现结果



#### 4. 下载 jar 包，并导入到项目的 lib 文件夹中



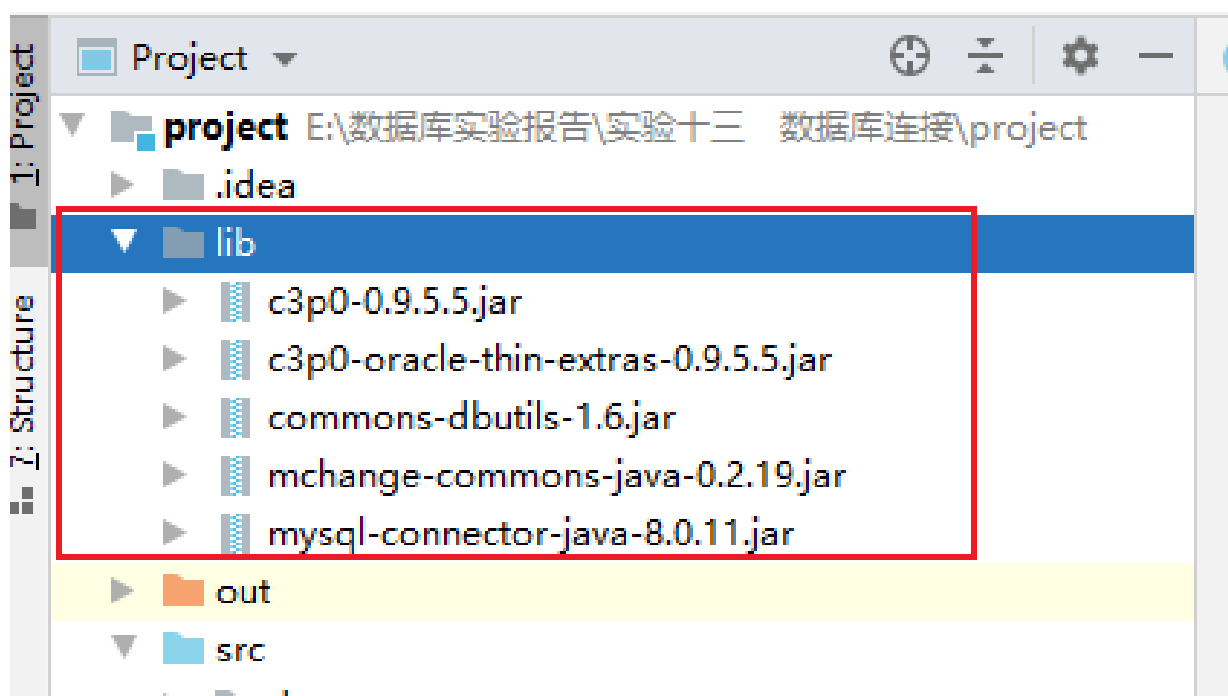
The screenshot shows the MySQL Connector/J 8.0.11 download page. The page title is "MySQL连接器/ J » 8.0.11" and it is described as "适用于MySQL的JDBC Type 4驱动程序". The page includes a table with the following information:

|      |   |
|------|---|
| 执照   | GPL 2.0   |
| 分类目录 | MySQL驱动   |
| 组织   | 甲骨文公司   |
| 主页   | <a href="http://dev.mysql.com/doc/connector-j/en/">http://dev.mysql.com/doc/connector-j/en/</a> |
| 日期   | (2018年4月18日)  |
| 档案   | <a href="#">pom (1 KB)</a> <a href="#">源 (1.9 MB)</a> <a href="#">查看全部</a>                      |
| 储存库  | 中央  |
| 使用者  | 4,434件文物  |

Below the table, there is a warning box: "注意: 此工件有新版本" (Warning: This artifact has a new version). The new version is listed as 8.0.20.

At the bottom, there is a code block for the Maven dependency:

```
<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
    <groupId>mysql</groupId>
```



#### 5. 在 model 中创建 Course 类，有“课程号”、“课程名”、“学分”、“学时”四个成员变量，并生成 getter 和 setter 方法

```

3  /**
4   * @author Administrator
5   */
6  public class Course {
7      /**课程号*/
8      private int id;
9      /**课程名*/
10     private String name;
11     /**课程学时*/
12     private int studyTime;
13     /**课程学分*/
14     private int studyGrade;
15
16     /**空构造器*/
17     public Course(){
18
19     }

```

6. 编写 JdbcUtils 类以及关闭方式等，方便在 Dao 层获取数据源

```

/**
 * @author Administrator
 */
public class JdbcUtils {
    public static ComboPooledDataSource dataSource = null;

    static {
        dataSource = new ComboPooledDataSource();
    }

    /**
     * 获取连接对象
     *
     * @return
     * @throws SQLException
     */
    public static Connection getConn() throws SQLException {
        return dataSource.getConnection();
    }

    /**

```

7. 编写 dao 层代码，实现从数据库抓取、更改记录等操作

根据课程名查询、根据课程号查询、根据课程名删除、根据课程号删除、修改课程信息、

---

## 添加课程、查询所有课程

源代码及注解：

```
package dao;

import com.mchange.v2.c3p0.ComboPooledDataSource;
import model.Course;
import org.apache.commons.dbutils.handlers.BeanHandler;
import org.apache.commons.dbutils.handlers.BeanListHandler;
import util.JdbcUtils;
import org.apache.commons.dbutils.QueryRunner;

import java.sql.SQLException;
import java.util.List;

/**
 * @author Administrator
 */
public class CourseDao {
    /** 获取连接池 dataSource*/
    ComboPooledDataSource comboPooledDataSource = JdbcUtils.dataSource;
    /** 创建 queryRunner*/
    QueryRunner queryRunner = new QueryRunner(comboPooledDataSource);
    /**
     * 根据课程名搜索
     */
    public Course search(String name) throws SQLException {
        Course course = null;
        String sql = "select * from 课程 where name = ?";
        //SQL 查询 返回一个 Bean 对象
        course = queryRunner.query(sql, new BeanHandler<Course>(Course.class), name);
        return course;
    }

    /**
     * 根据课程号搜索
     */
    public Course search(int id) throws SQLException {
        Course course = null;
        String sql = "select * from 课程 where id = ?";
        //SQL 查询 返回一个 Bean 对象
        course = queryRunner.query(sql, new BeanHandler<Course>(Course.class), id);
        return course;
    }
}
```

---

```
}

/**
 *查询所有的课程记录
 */
public List<Course> searchAll() throws SQLException {
    List<Course> courses;
    String sql = "select * from 课程";
    //SQL 查询 返回 List 容器
    courses = queryRunner.query(sql, new BeanListHandler<Course>(Course.class));
    return courses;
}

/**
 *增加一条课程记录
 */
public boolean addCourse(Course course) throws SQLException {
    String sql = "insert into 课程(id,name,studyTime, studyGrade) values(?,?,?,?)";
    //SQL 查询 返回受影响的行数
    int row =
queryRunner.update(sql,course.getId(),course.getName(),course.getStudyTime(),course.getStudyGrade(
));
    //如果受影响的行数为正整数，那么添加记录成功
    if(row>0){
        return true;
    }else {
        return false;
    }
}

/**
 *根据 id 删除一条课程记录
 */
public boolean deleteCourse(int id) throws SQLException {
    String sql = "delete from 课程 where id = ?";
    //SQL 查询 返回受影响的行数
    int row = queryRunner.update(sql,id);
    //如果受影响的行数为正整数，那么删除记录成功
    if(row>0) {
        return true;
    }else {
        return false;
    }
}

/**
```

---

```

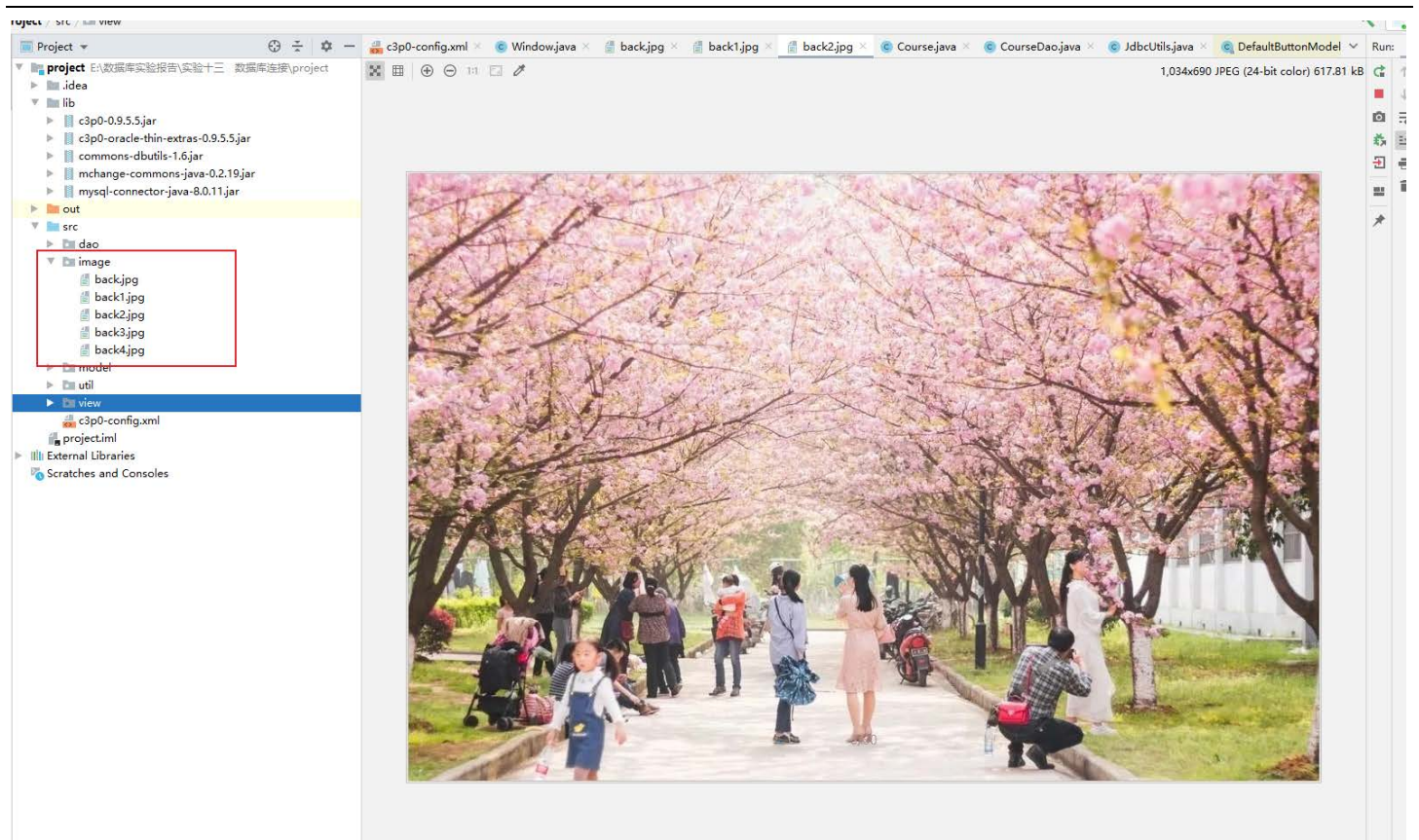
    *根据课程名删除一条记录
    */
    public boolean deleteCourse(String string) throws SQLException {
        String sql = "delete from 课程 where name = ?";
        //SQL 查询 返回受影响的行数
        int row = queryRunner.update(sql,string);
        //如果受影响的行数为正整数，那么删除记录成功
        if(row>0) {
            return true;
        }else {
            return false;
        }
    }

    /**
    *更改一条记录
    */
    public boolean updateCourse(int id, Course course) throws SQLException {
        String sql = "update 课程 set id = ?,name = ?,studyTime = ?, studyGrade = ? where id = ?";
        //SQL 查询 返回受影响的行数
        int row =
        queryRunner.update(sql,course.getId(),course.getName(),course.getStudyTime(),course.getStudyGrade(
        ),id);
        //如果受影响的行数为正整数，那么更改记录成功
        if(row>0){
            return true;
        }else {
            return false;
        }
    }
}

```

8. 导入图片到 image 文件夹，方便后续代码获取主界面背景





## 9. 设计主界面 Window 类，继承 JFrame 类



主界面

源代码:



---

```
package view;

import dao.CourseDao;
import model.Course;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.SQLException;

/**
 * @author Administrator
 * <p>窗口：程序主界面
 */
public class Window extends JFrame {
    CourseDao courseDao = new CourseDao();

    public Window() throws HeadlessException {
        // 设置主界面标题
        this.setTitle("课程管理");
        // 设置窗口弹出居中
        this.setLocationRelativeTo(null);
        // 设置窗口大小
        this.setSize(800, 400);
        // 设置关闭方式为程序退出
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // 窗口布局
        this.setLayout(new BorderLayout());
        // 新建面板，并设置网格布局
        JPanel jPanel = new JPanel();
        jPanel.setLayout(new GridLayout(0, 1));
        // 在窗口左侧添加图片
        JLabel label = new JLabel();
        ImageIcon imageIcon = new ImageIcon("src/image/back2.jpg");
        label.setIcon(imageIcon);
        // 新建操作按钮
        JButton searchAllButton = new JButton("查询所有课程");
        JButton searchIdButton = new JButton("课程号查询");
        JButton searchNameButton = new JButton("课程名查询");
        JButton addButton = new JButton("添加课程");
        JButton deleteIdButton = new JButton("课程号删除");
        JButton deleteNameButton = new JButton("课程名删除");
        JButton updateButton = new JButton("更改课程信息");
        // 设置按钮的大小
        searchAllButton.setSize(50, 30);
```

---

```
searchIdButton.setSize(50, 30);
searchNameButton.setSize(50, 30);
addButton.setSize(500, 30);
deleteIdButton.setSize(50, 30);
deleteNameButton.setSize(50, 30);
updateButton.setSize(50, 30);
// 查询所有课程 按钮 添加动作监听
searchAllButton.addActionListener(
    new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            try {
                // 弹出一个新的查询结果界面
                new SearchAll();
            } catch (SQLException throwables) {
                throwables.printStackTrace();
            }
        }
    }
);

// 为根据 Id 搜索课程 按钮添加动作监听
searchIdButton.addActionListener(
    new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            // 直到点击取消按钮或者叉，关闭窗口
            while (true) {
                // 弹出输入框，输入 id，类型为 String
                String id = JOptionPane.showInputDialog(null, "请输入课程号");
                // 如果没有输入，直接点击取消或者叉
                if (id == null) {
                    // 弹出正在取消按钮
                    JOptionPane.showMessageDialog(null, "正在取消.....", "提示",
JOptionPane.WARNING_MESSAGE);
                    // 跳出循环，不再弹出窗口
                    break;
                }
                // 如果没有输入而点击确认按钮
                else if (id.equals("")) {
                    // 弹出提示框：请输入课程号
                    JOptionPane.showMessageDialog(null, "未输入课程号！ ", "提示",
JOptionPane.WARNING_MESSAGE);
                    // 继续循环
                    continue;
                }
            }
            // 有信息输入
```

---

```
else {
    // 如果输入的是数字串
    if (isDigital(id)) {
        Course result = null;
        try {
            // 将字符串转换成整形数字, 查询数据库
            result = courseDao.search(Integer.parseInt(id));
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        }
        // 成功查询
        if (result != null) {
            // 弹出提示框, 展示查询结果
            JOptionPane.showMessageDialog(
                null, result.getInfo(), "查询结果", JOptionPane.WARNING_MESSAGE);
        } else {
            // 没有查询到相关课程
            JOptionPane.showMessageDialog(
                null, "未查询到该课程", "查询结果", JOptionPane.WARNING_MESSAGE);
        }
    }
    // 如果字符串不是数字串
    else {
        // 弹出提示: 输入的不是整型课程号
        JOptionPane.showMessageDialog(
            null, "输入的不是整型课程号, 请重新输入!", "提示",
JOptionPane.WARNING_MESSAGE);
        continue;
    }
}
}
}
});

// 为根据课程名查询 按钮添加动作监听
searchNameButton.addActionListener(
    new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            while (true) {
                // 弹出输入框, 输入课程名
                String name = JOptionPane.showInputDialog(null, "请输入课程名");
                // 如果点击取消或者叉, 弹出正在取消窗口, 并退出
                if (name == null) {
                    JOptionPane.showMessageDialog(null, "正在取消.....", "提示",
JOptionPane.WARNING_MESSAGE);
```

---

```
        break;
    }
    // 如果没有输入，点击了确定
    else if (name.equals("")) {
        // 弹出提示框，没有输入课程名
        JOptionPane.showMessageDialog(null, "未输入课程名！", "提示",
JOptionPane.WARNING_MESSAGE);
        continue;
    }
    // 如果有输入信息
    else {
        Course result = null;
        try {
            // 根据课程名查询
            result = courseDao.search(name);
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        }
        // 查询成功
        if (result != null) {
            // 弹出窗口，展示查询结果
            JOptionPane.showMessageDialog(
                null, result.getInfo(), "查询结果", JOptionPane.WARNING_MESSAGE);
        } else {
            // 查询失败，弹出提示框
            JOptionPane.showMessageDialog(
                null, "未查询到该课程", "查询结果", JOptionPane.WARNING_MESSAGE);
        }
    }
}
}
});
```

```
// 为新增按钮添加动作监听
addButton.addActionListener(
    new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            new AddCourse();
        }
    }
);
```

```
// 为根据课程号删除 按钮 添加动作监听
deleteIdButton.addActionListener(
    new ActionListener() {
        @Override
```

---

```
public void actionPerformed(ActionEvent e) {
    while (true) {
        // 弹出窗口，输入课程号
        String id = JOptionPane.showInputDialog(null, "请输入要删除的课程号");
        // 如果点击取消按钮或者叉，弹出正在取消界面
        if (id == null) {
            JOptionPane.showMessageDialog(null, "正在取消.....", "提示",
JOptionPane.WARNING_MESSAGE);
            break;
        }
        // 如果没有输入而点击确定，弹出提示框
        else if (id.equals("")) {
            JOptionPane.showMessageDialog(null, "未输入课程号！", "提示",
JOptionPane.WARNING_MESSAGE);
            continue;
        }
        // 如果有信息输入
        else {
            // 输入的字符串是数字串
            if (isDigital(id)) {
                boolean result = false;
                try {
                    // 根据课程号删除课程记录
                    result = courseDao.deleteCourse(Integer.parseInt(id));
                } catch (SQLException throwables) {
                    throwables.printStackTrace();
                }
                // 删除成功
                if (result) {
                    // 弹出提示框
                    JOptionPane.showMessageDialog(
                        null, "已查询到该课程，删除成功！", "查询结果",
JOptionPane.WARNING_MESSAGE);
                } else {
                    // 没有找到相应的课程 删除失败
                    JOptionPane.showMessageDialog(
                        null, "未查询到该课程，删除失败！", "查询结果",
JOptionPane.WARNING_MESSAGE);
                }
            } else {
                // 如果输入的不是整形数字串 弹出提示窗口
                JOptionPane.showMessageDialog(
                    null, "输入的不是整形课程号，请重新输入！", "提示",
JOptionPane.WARNING_MESSAGE);
                continue;
            }
        }
    }
}
```





---

```
JOptionPane.WARNING_MESSAGE);
        }
    }
}
});

// 为 更改课程信息 添加动作监听
updateButton.addActionListener(
    new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            // 弹出一个新的窗口
            new UpdateCourse();
        }
    });

// 将各个按钮添加到面板中
jPanel.add(searchNameButton);
jPanel.add(searchIdButton);
jPanel.add(addButton);
jPanel.add(deleteNameButton);
jPanel.add(deleteIdButton);
jPanel.add(updateButton);
jPanel.add(searchAllButton);

// 将图片标签添加到窗口
this.add(label);
// 将面板添加到窗口的右侧
this.add(jPanel, BorderLayout.EAST);
// 设置窗口可见
this.setVisible(true);
}

/** 判断字符串是否为整形数字串 */
public static boolean isDigital(String string) {
    for (int i = 0; i < string.length(); i++) {
        // 如果不是'0'-'9', 则不是数字串, 返回 false
        if (!(string.charAt(i) >= 48 && string.charAt(i) <= 57)) {
            return false;
        }
    }
    return true;
}
```

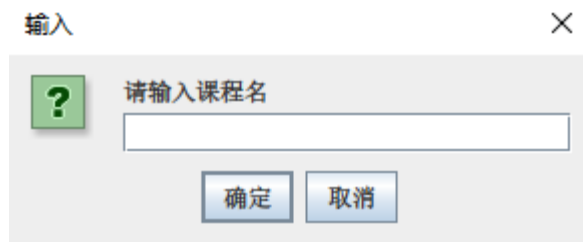
---

## 10. 实现课程名查询

```
// 为根据课程名查询 按钮添加动作监听
searchNameButton.addActionListener(
    new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            while (true) {
                // 弹出输入框，输入课程名
                String name = JOptionPane.showInputDialog(null, "请输入课程名");
                // 如果点击取消或者叉，弹出正在取消窗口，并退出
                if (name == null) {
                    JOptionPane.showMessageDialog(null, "正在取消.....", "提示",
JOptionPane.WARNING_MESSAGE);
                    break;
                }
                // 如果没有输入，点击了确定
                else if (name.equals("")) {
                    // 弹出提示框，没有输入课程名
                    JOptionPane.showMessageDialog(null, "未输入课程名！ ", "提示",
JOptionPane.WARNING_MESSAGE);
                    continue;
                }
                // 如果有输入信息
                else {
                    Course result = null;
                    try {
                        // 根据课程名查询
                        result = courseDao.search(name);
                    } catch (SQLException throwables) {
                        throwables.printStackTrace();
                    }
                    // 查询成功
                    if (result != null) {
                        // 弹出窗口，展示查询结果
                        JOptionPane.showMessageDialog(
                            null, result.getInfo(), "查询结果", JOptionPane.WARNING_MESSAGE);
                    } else {
                        // 查询失败，弹出提示框
                        JOptionPane.showMessageDialog(
                            null, "未查询到该课程", "查询结果", JOptionPane.WARNING_MESSAGE);
                    }
                }
            }
        }
    }
}
```

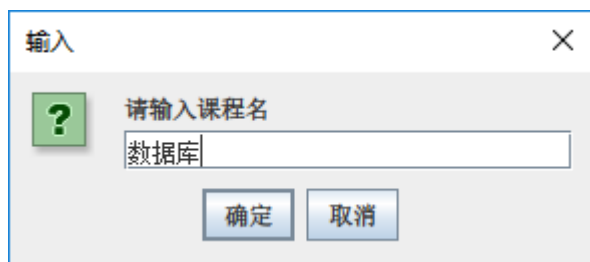
```
}  
});
```

课程名查询：

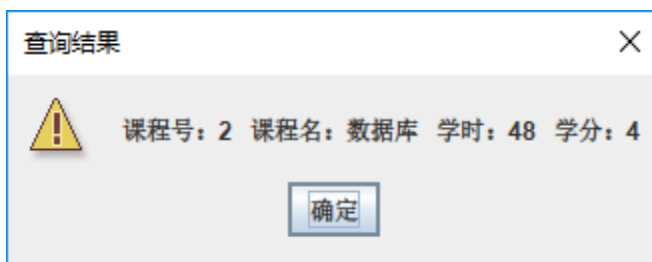


A dialog box titled "输入" (Input) with a close button (X) in the top right corner. It contains a green question mark icon, the text "请输入课程名" (Please enter course name), an empty text input field, and two buttons at the bottom: "确定" (OK) and "取消" (Cancel).

(1) 输入“数据库”，点击确定，正确查询



The "输入" dialog box with "数据库" entered in the text field. The "确定" button is highlighted.



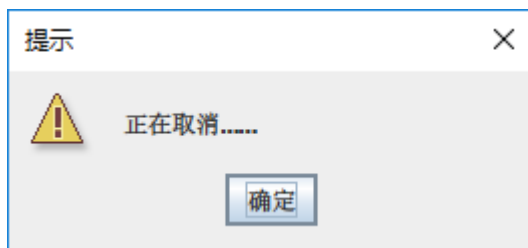
A dialog box titled "查询结果" (Query Result) with a close button (X) in the top right corner. It contains a yellow warning icon, the text "课程号：2 课程名：数据库 学时：48 学分：4", and a "确定" button at the bottom.

(2) 不输入，直接点击确定



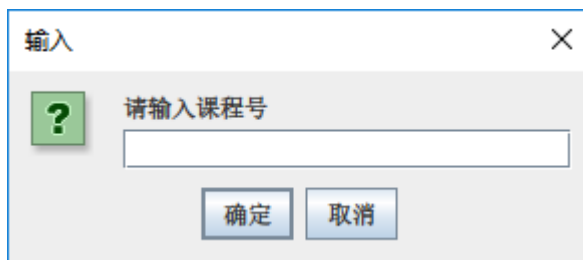
A dialog box titled "提示" (Prompt) with a close button (X) in the top right corner. It contains a yellow warning icon, the text "未输入课程名!", and a "确定" button at the bottom.

(3) 点击取消或者右上方叉号



A dialog box titled "提示" (Prompt) with a close button (X) in the top right corner. It contains a yellow warning icon, the text "正在取消.....", and a "确定" button at the bottom.

11. 实现根据课程号查询



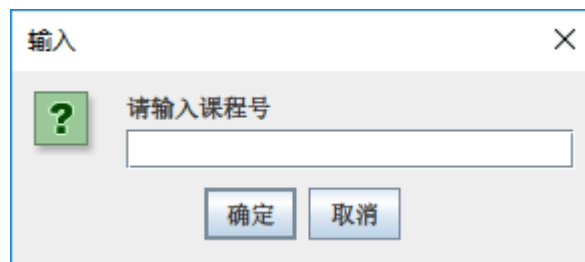
A dialog box titled "输入" (Input) with a close button (X) in the top right corner. It contains a green question mark icon, the text "请输入课程号" (Please enter course number), an empty text input field, and two buttons at the bottom: "确定" (OK) and "取消" (Cancel).

界面

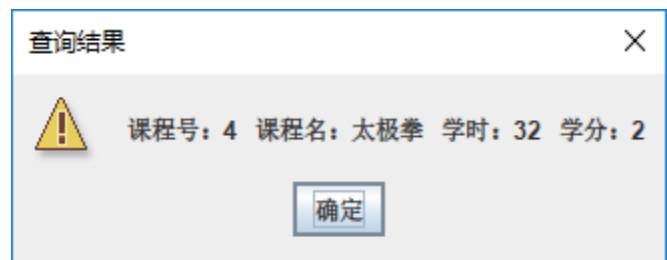
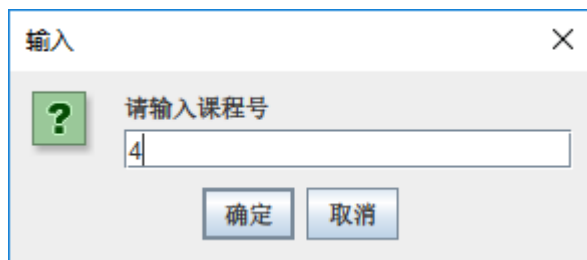
```
// 为根据Id 搜索课程 按钮添加动作监听
searchIdButton.addActionListener(
    new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            // 直到点击取消按钮或者叉，关闭窗口
            while (true) {
                // 弹出输入框，输入id，类型为String
                String id = JOptionPane.showInputDialog(null, "请输入课程号");
                // 如果没有输入，直接点击取消或者叉
                if (id == null) {
                    // 弹出正在取消按钮
                    JOptionPane.showMessageDialog(null, "正在取消.....", "提示",
                        JOptionPane.WARNING_MESSAGE);
                    // 跳出循环，不再弹出窗口
                    break;
                }
                // 如果没有输入而点击确认按钮
                else if (id.equals("")) {
                    // 弹出提示框：请输入课程号
                    JOptionPane.showMessageDialog(null, "未输入课程号！", "提示",
                        JOptionPane.WARNING_MESSAGE);
                    // 继续循环
                    continue;
                }
                // 有信息输入
                else {
                    // 如果输入的是数字串
                    if (isDigital(id)) {
                        Course result = null;
                        try {
                            // 将字符串转换成整形数字，查询数据库
                            result = courseDao.search(Integer.parseInt(id));
                        } catch (SQLException throwables) {
                            throwables.printStackTrace();
                        }
                    }
                    // 成功查询
                    if (result != null) {
                        // 弹出提示框，展示查询结果
                        JOptionPane.showMessageDialog(
                            null, result.getInfo(), "查询结果", JOptionPane.WARNING_MESSAGE);
                    } else {
                        // 没有查询到相关课程
                        JOptionPane.showMessageDialog(
```

```
        null, "未查询到该课程", "查询结果", JOptionPane.WARNING_MESSAGE);
    }
}
// 如果字符串不是数字串
else {
    // 弹出提示: 输入的不是整型课程号
    JOptionPane.showMessageDialog(
        null, "输入的不是整形课程号, 请重新输入!", "提示",
        JOptionPane.WARNING_MESSAGE);
    continue;
}
}
}
}
});
```

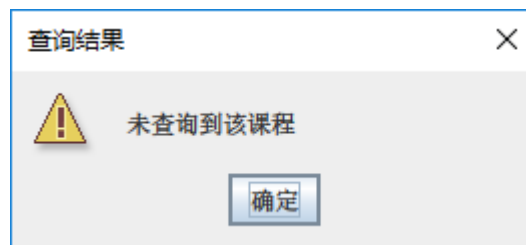
课程号查询:



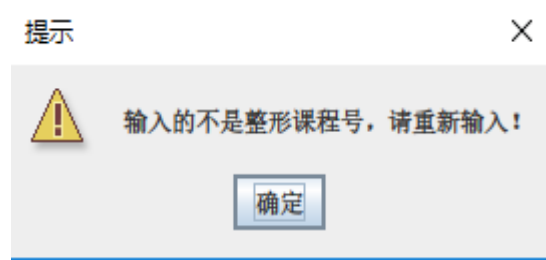
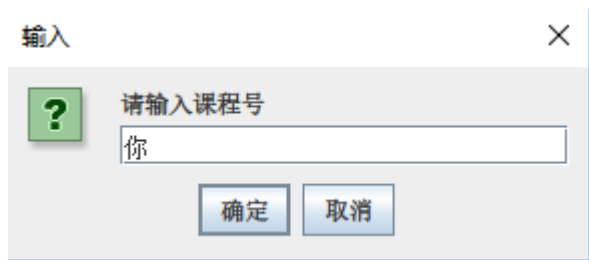
(1) 输入 4，点击确定，正确查询



(2) 输入 100，点击确定，查找失败（数据库中没有课程号为 100 的课程）



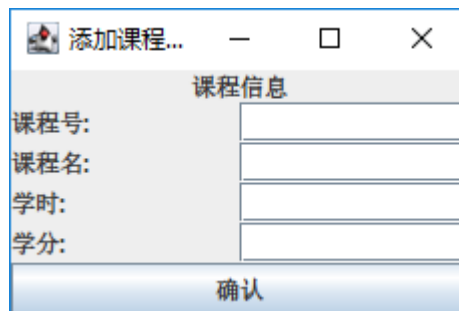
(3) 输入非数字串，比如“你”，点击确定



(4) 点击取消按钮或者右上角叉号



12. 实现添加课程，创建 AddCourse 类，继承 JFrame 类



界面

```
package view;
```

```
import dao.CourseDao;
import model.Course;
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.SQLException;
```

```
/**
 * @author Administrator
 * 窗口：添加课程
 *
 * */
public class AddCourse extends JFrame {
    public AddCourse() {
        // 设置窗口弹出位置居中
```



```
this.setLocationRelativeTo(null);
// 设置窗口名称
this.setTitle("添加课程信息");
JTextField id, name, studyTime, studyGrade;
// 提示标签
JLabel jl1 = new JLabel("课程号:");
// 输入框
id = new JTextField(10);
JLabel jl2 = new JLabel("课程名:");
name = new JTextField(10);
JLabel jl3 = new JLabel("学时:");
studyTime = new JTextField(10);
JLabel jl4 = new JLabel("学分:");
studyGrade = new JTextField(10);
// 设置主面板为 4*2 的网格布局
JPanel jp = new JPanel(new GridLayout(4, 2));
// 将各个组件加入到面板中
jp.add(jl1);
jp.add(id);
jp.add(jl2);
jp.add(name);
jp.add(jl3);
jp.add(studyTime);
jp.add(jl4);
jp.add(studyGrade);
JButton jb = new JButton("确认");
JLabel jlinfo;
jlinfo = new JLabel("课程信息", JLabel.CENTER);
// 为“确认”按钮添加动作监听
jb.addActionListener(
    new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            // 如果没有输入完整信息，比如课程名没有填写等
            if (id.getText().equals(""))
                || name.getText().equals("")
                || studyGrade.getText().equals("")
                || studyTime.getText().equals("")) {
                // 弹出窗口，提示输入完整信息
                JOptionPane.showMessageDialog(
                    null, "信息不完整，请填写好信息", "提示", JOptionPane.WARNING_MESSAGE);
            } else {
                if ((!Window.isDigital(id.getText())) || (!Window.isDigital(studyTime.getText())) ||
                    (!Window.isDigital(studyGrade.getText())) {
                    JOptionPane.showMessageDialog(
                        null, "课程号 or 学时 or 学分输入的不是整形课程号，请重新输入！", "提示
```

```

", JOptionPane.WARNING_MESSAGE);
    }else{
        // 将输入的信息封装成对象
        Course newCourse = new Course();
        newCourse.setId(Integer.parseInt(id.getText()));
        newCourse.setName(name.getText());
        newCourse.setStudyTime(Integer.parseInt(studyTime.getText()));
        newCourse.setStudyGrade(Integer.parseInt(studyGrade.getText()));
        boolean result = false;
        try {
            // 创建 dao 层实例化对象
            CourseDao courseDao = new CourseDao();
            // 执行添加课程方法，返回布尔值
            result = courseDao.addCourse(newCourse);
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        }
        // 如果添加成功，将文本域清空，以便继续添加
        if (result) {
            // 弹出添加成功提示框
            JOptionPane.showMessageDialog(null, "添加成功！ ", "结果",
JOptionPane.WARNING_MESSAGE);
            id.setText("");
            name.setText("");
            studyTime.setText("");
            studyGrade.setText("");
        } else {
            // 课程表的课程号为主键，因此不能有重复的课程号，这里弹出添加失败
            JOptionPane.showMessageDialog(
                null, "已有课程号为" + id.getText() + "的课程，添加失败！ ", "结果",
JOptionPane.WARNING_MESSAGE);
        }
    }
}
}
}
});

// 添加面板
this.add(jp);
this.add(jlinfo, BorderLayout.NORTH);
// 在底部添加确认按钮
this.add(jb, BorderLayout.SOUTH);
// 窗口自动调整大小
this.pack();
// 设置窗口可见
this.setVisible(true);

```

}  
}

(1) 输入

课程号：7                      课程名：初中数学

学时：32                      学分：2

点击确定

添加课程...

课程信息

课程号:

7

课程名:

初中数学

学时:

32

学分:

2

确认

结果

添加成功!

确定

|       | 表    | 视图                   | 函数     | 事件        | 用户         | 查询 | 报表 |
|-------|------|----------------------|--------|-----------|------------|----|----|
|       | 对象   | 课程 @学生管理 (MySQL) - 表 |        |           |            |    |    |
| hema  | 开始事务 | 文本                   | 筛选     | 排序        | 导入         | 导出 |    |
| chema |      | id                   | name   | studyTime | studyGrade |    |    |
|       |      | 1                    | 外语     | 64        | 4          |    |    |
|       |      | 2                    | 数据库    | 48        | 4          |    |    |
|       |      | 3                    | 概率论    | 64        | 4          |    |    |
|       |      | 4                    | 太极拳    | 32        | 2          |    |    |
|       |      | 5                    | 计算机基础  | 64        | 4          |    |    |
|       |      | 9                    | 高中数学   | 64        | 4          |    |    |
|       |      | 10                   | 数据结构实训 | 32        | 2          |    |    |
|       |      | 12                   | 操作系统   | 64        | 4          |    |    |

原课程表

| 对象 课程 @学生管理 (MySQL) - 表 |        |           |            |  |
|-------------------------|--------|-----------|------------|--|
| 开始事务 文本 筛选 排序 导入 导出     |        |           |            |  |
| id                      | name   | studyTime | studyGrade |  |
| 1                       | 外语     | 64        | 4          |  |
| 2                       | 数据库    | 48        | 4          |  |
| 3                       | 概率论    | 64        | 4          |  |
| 4                       | 太极拳    | 32        | 2          |  |
| 5                       | 计算机基础  | 64        | 4          |  |
| 7                       | 初中数学   | 32        | 2          |  |
| 9                       | 高中数学   | 64        | 4          |  |
| 10                      | 数据结构实训 | 32        | 2          |  |
| 12                      | 操作系统   | 64        | 4          |  |

刷新后的课程表（添加成功）

（2）输入

课程号：9

课程名：语文

学时：64

学分：4

添加课程...

课程信息

课程号:

9

课程名:

语文

学时:

64

学分:

4

确认

结果

!

已有课程号为9的课程，添加失败！

确定

对象 课程 @学生管理 (MySQL) - 表

开始事务 文本 筛选 排序 导入 导出

a

| id | name   | studyTime | studyGrade |
|----|--------|-----------|------------|
| 1  | 外语     | 64        | 4          |
| 2  | 数据库    | 48        | 4          |
| 3  | 概率论    | 64        | 4          |
| 4  | 太极拳    | 32        | 2          |
| 5  | 计算机基础  | 64        | 4          |
| 7  | 初中数学   | 32        | 2          |
| 9  | 高中数学   | 64        | 4          |
| 10 | 数据结构实训 | 32        | 2          |
| 12 | 操作系统   | 64        | 4          |

原课程表（已有课程号为 9 的课程）

（3）输入

课程号：13

课程名：语文

学时：你

学分：4

点击确认

添加课程...

课程信息

课程号: 13

课程名: 语文

学时: 你

学分: 4

确认

提示

课程号or学时or学分输入的不是整形课程号，请重新输入！

确定

（4）输入

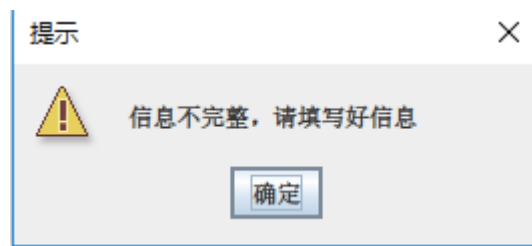
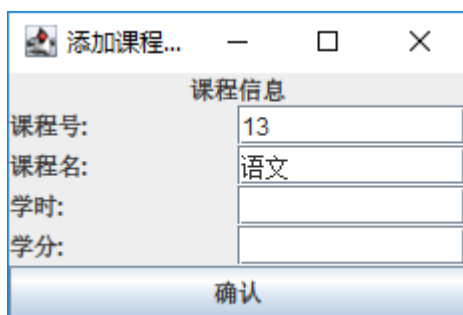
课程号：13

课程名：语文

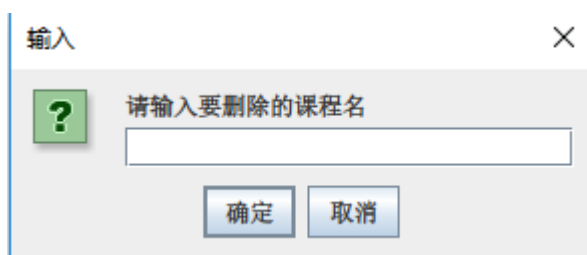
学时：不输入

学分：不输入

点击确认



### 13. 实现根据课程名删除



界面

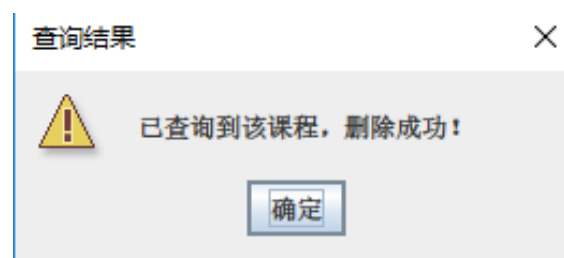
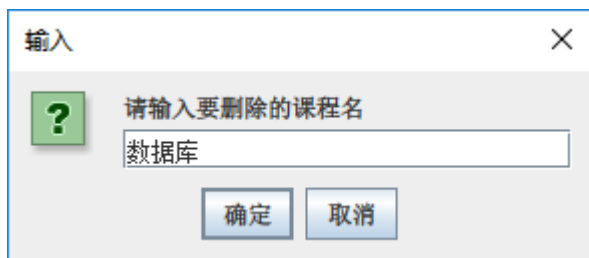
源代码：

```
// 为根据课程名删除添加动作监听
deleteNameButton.addActionListener(
    new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            while (true) {
                // 弹出窗口 输入课程名
                String name = JOptionPane.showInputDialog(null, "请输入要删除的课程名");
                // 如果点击取消或者叉
                if (name == null) {
                    // 弹出正在取消窗口
                    JOptionPane.showMessageDialog(null, "正在取消.....", "提示",
                        JOptionPane.WARNING_MESSAGE);
                    break;
                }
                // 如果没有输入 而点击了确认
                else if (name.equals("")) {
                    // 弹出 输入课程名提示框
                    JOptionPane.showMessageDialog(null, "未输入课程名!", "提示",
                        JOptionPane.WARNING_MESSAGE);
                    continue;
                }
                // 有信息输入
                else {
```



```
boolean result = false;
try {
    // 根据课程名删除课程记录
    result = courseDao.deleteCourse(name);
} catch (SQLException throwables) {
    throwables.printStackTrace();
}
// 删除成功
if (result) {
    // 成功提示框
    JOptionPane.showMessageDialog(
        null, "已查询到该课程，删除成功！", "查询结果",
        JOptionPane.WARNING_MESSAGE);
} else {
    // 没有找到相应课程    删除课程失败
    JOptionPane.showMessageDialog(
        null, "未查询到该课程，删除失败！", "查询结果",
        JOptionPane.WARNING_MESSAGE);
}
}
}
}
});
```

(1) 输入“数据库”，点击确定



| id | name   | studyTime | studyGrade |
|----|--------|-----------|------------|
| 1  | 外语     | 64        | 4          |
| 2  | 数据库    | 48        | 4          |
| 3  | 概率论    | 64        | 4          |
| 4  | 太极拳    | 32        | 2          |
| 5  | 计算机基础  | 64        | 4          |
| 7  | 初中数学   | 32        | 2          |
| 9  | 高中数学   | 64        | 4          |
| 10 | 数据结构实训 | 32        | 2          |
| 12 | 操作系统   | 64        | 4          |

原课程表

| id | name   | studyTime | studyGrade |
|----|--------|-----------|------------|
| 1  | 外语     | 64        | 4          |
| 3  | 概率论    | 64        | 4          |
| 4  | 太极拳    | 32        | 2          |
| 5  | 计算机基础  | 64        | 4          |
| 7  | 初中数学   | 32        | 2          |
| 9  | 高中数学   | 64        | 4          |
| 10 | 数据结构实训 | 32        | 2          |
| 12 | 操作系统   | 64        | 4          |

刷新后的课程表（成功删除）

(2) 无输入，直接点击确定

输入

×

?

请输入要删除的课程名

确定

取消

提示

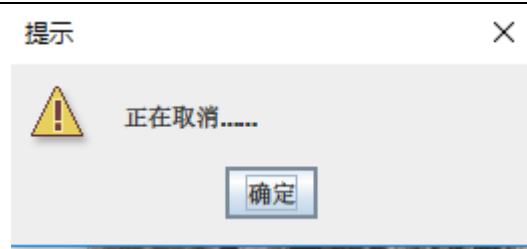
×

!

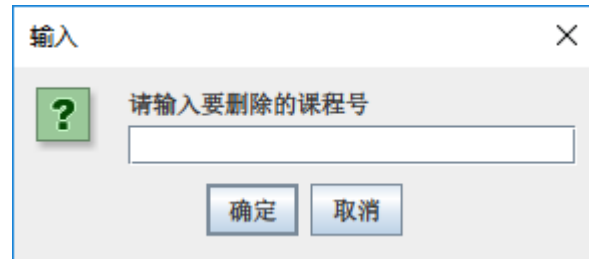
未输入课程名！

确定

(3) 点击取消按钮或者右上角的叉号



#### 14. 实现根据课程号删除



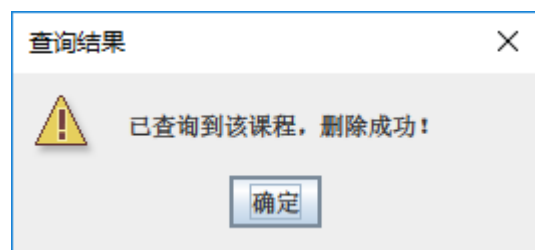
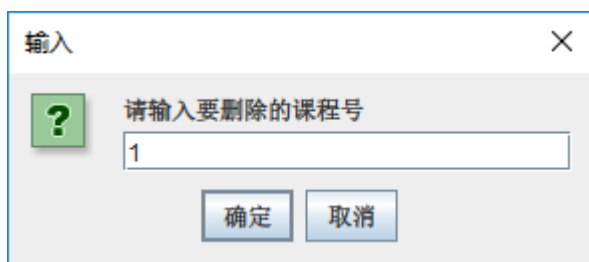
界面

源代码:

```
// 为根据课程号删除 按钮 添加动作监听
deleteIdButton.addActionListener(
    new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            while (true) {
                // 弹出窗口，输入课程号
                String id = JOptionPane.showInputDialog(null, "请输入要删除的课程号");
                // 如果点击取消按钮或者叉，弹出正在取消界面
                if (id == null) {
                    JOptionPane.showMessageDialog(null, "正在取消.....", "提示",
JOptionPane.WARNING_MESSAGE);
                    break;
                }
                // 如果没有输入而点击确定，弹出提示框
                else if (id.equals("")) {
                    JOptionPane.showMessageDialog(null, "未输入课程号!", "提示",
JOptionPane.WARNING_MESSAGE);
                    continue;
                }
                // 如果有信息输入
                else {
                    // 输入的字符串是数字串
                    if (isDigital(id)) {
                        boolean result = false;
                        try {
                            // 根据课程号删除课程记录
```

```
        result = courseDao.deleteCourse(Integer.parseInt(id));
    } catch (SQLException throwables) {
        throwables.printStackTrace();
    }
    // 删除成功
    if (result) {
        // 弹出提示框
        JOptionPane.showMessageDialog(
            null, "已查询到该课程，删除成功！", "查询结果",
JOptionPane.WARNING_MESSAGE);
    } else {
        // 没有找到相应的课程 删除失败
        JOptionPane.showMessageDialog(
            null, "未查询到该课程，删除失败！", "查询结果",
JOptionPane.WARNING_MESSAGE);
    }
    } else {
        // 如果输入的不是整形数字串 弹出提示窗口
        JOptionPane.showMessageDialog(
            null, "输入的不是整形课程号，请重新输入！", "提示",
JOptionPane.WARNING_MESSAGE);
        continue;
    }
}
}
}
});
```

(1) 输入“1”，点击确定



| id | name   | studyTime | studyGrade |
|----|--------|-----------|------------|
| 1  | 外语     | 64        | 4          |
| 3  | 概率论    | 64        | 4          |
| 4  | 太极拳    | 32        | 2          |
| 5  | 计算机基础  | 64        | 4          |
| 7  | 初中数学   | 32        | 2          |
| 9  | 高中数学   | 64        | 4          |
| 10 | 数据结构实训 | 32        | 2          |
| 12 | 操作系统   | 64        | 4          |

原课程表

| id | name   | studyTime | studyGrade |
|----|--------|-----------|------------|
| 3  | 概率论    | 64        | 4          |
| 4  | 太极拳    | 32        | 2          |
| 5  | 计算机基础  | 64        | 4          |
| 7  | 初中数学   | 32        | 2          |
| 9  | 高中数学   | 64        | 4          |
| 10 | 数据结构实训 | 32        | 2          |
| 12 | 操作系统   | 64        | 4          |

刷新后的课程表（成功删除）

(2) 输入“我”，非整形数字串

输入

×

?

请输入要删除的课程号

我

确定

取消

提示

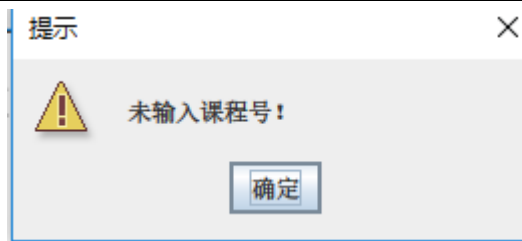
×

!

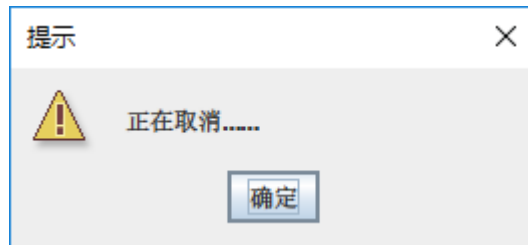
输入的不是整形课程号，请重新输入！

确定

(3) 不输入，直接点击确定



(4) 点击取消或者右上方的叉号



## 16. 实现更改课程信息

界面

源代码:

```
package view;
```

```
import dao.CourseDao;
```

```
import model.Course;
```

```
import javax.swing.*;
```

```
import java.awt.*;
```

```
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;
```

```
import java.sql.SQLException;
```

```
/** @author Administrator 窗口: 更新课程信息 */
```

```
public class UpdateCourse extends JFrame {
```

```
    public UpdateCourse() {
```

```
        // 设置窗口弹出位置居中
```

```
        this.setLocationRelativeTo(null);
```

```
        // 设置窗口名称
```

```
        this.setTitle("修改课程信息");
```



---

```
// 窗口布局
this.setLayout(new BorderLayout());
// 声明文本域对象
JTextField id, newId, newName, newStudyTime, newStudyGrade;
// 提示标签
JLabel jl0 = new JLabel("原课程号: ");
// 用来输入的文本域
id = new JTextField(10);
JLabel jl1 = new JLabel("课程号:");
newId = new JTextField(10);
JLabel jl2 = new JLabel("课程名:");
newName = new JTextField(10);
JLabel jl3 = new JLabel("学时:");
newStudyTime = new JTextField(10);
JLabel jl4 = new JLabel("学分:");
newStudyGrade = new JTextField(10);
JPanel jp = new JPanel(new GridLayout(5, 2));
// 将标签和文本域添加到面板中
jp.add(jl0);
jp.add(id);
jp.add(jl1);
jp.add(newId);
jp.add(jl2);
jp.add(newName);
jp.add(jl3);
jp.add(newStudyTime);
jp.add(jl4);
jp.add(newStudyGrade);
// 确认按钮
JButton jb = new JButton("确认");
// 为确认按钮添加动作监听
jb.addActionListener(
    new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            // 如果没有输入完整信息, 比如课程名没有填写等
            if (id.getText().equals(""))
                || newId.getText().equals(""))
                || newName.getText().equals(""))
                || newStudyTime.getText().equals(""))
                || newStudyGrade.getText().equals("")) {
                // 弹出窗口, 提示输入完整信息
                JOptionPane.showMessageDialog(
                    null, "信息不完整, 请填写好信息", "提示", JOptionPane.WARNING_MESSAGE);
            } else {
                // 原课程号和课程号输入不是整形数字串
```

---

```

        if ((!Window.isDigital(id.getText())) || (!Window.isDigital(newId.getText())) ||
(!Window.isDigital(newStudyTime.getText())) || (!Window.isDigital(newStudyGrade.getText())) {
            JOptionPane.showMessageDialog(
                null, "原课程号 or 课程号 or 学时 or 学分输入的不是整形课程号，请重新输入！
", "提示", JOptionPane.WARNING_MESSAGE);
        } else {
            // 信息输入完成，进行更改操作
            CourseDao courseDao = new CourseDao();
            // 将输入的信息封装成一个 Bean 对象
            Course newCourse = new Course();
            newCourse.setId(Integer.parseInt(newId.getText()));
            newCourse.setName(newName.getText());
            newCourse.setStudyTime(Integer.parseInt(newStudyTime.getText()));
            newCourse.setStudyGrade(Integer.parseInt(newStudyGrade.getText()));
            boolean result = false;
            try {
                // 执行更新语句，利用原课程号找到课程记录，再进行更改
                result = courseDao.updateCourse(Integer.parseInt(id.getText()), newCourse);
            } catch (SQLException throwables) {
                throwables.printStackTrace();
            }
            // 如果修改成功，将输入文本域清空，以便继续修改
            if (result) {
                // 弹出提示框，修改成功
                JOptionPane.showMessageDialog(null, "修改成功！ ", "结果",
JOptionPane.WARNING_MESSAGE);
                id.setText("");
                newId.setText("");
                newName.setText("");
                newStudyGrade.setText("");
                newStudyTime.setText("");
            } else {
                // 课程表的课程号为主键，因此不能有重复的课程号，这里弹出添加失败
                JOptionPane.showMessageDialog(
                    null,
                    "没有课程号为" + id.getText() + "的课程，修改失败！ ",
                    "结果",
                    JOptionPane.WARNING_MESSAGE);
            }
        }
    }
}
});
// 将面板添加进窗口
this.add(jp);
// 将按钮添加到窗口底部

```

```

this.add(jb, BorderLayout.SOUTH);
// 自动调整窗口大小
this.pack();
// 设置窗口可见
this.setVisible(true);
}
}

```

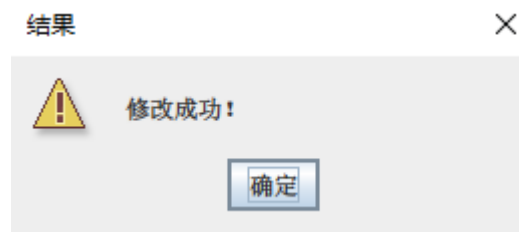
(1) 输入

原课程号：4

课程号：4                  课程名：体育-太极拳

学时：32                  学分：2

点击确认



| 对象 课程 @学生管理 (MySQL) - 表 |        |           |            |  |
|-------------------------|--------|-----------|------------|--|
| 开始事务 文本 筛选 排序 导入 导出     |        |           |            |  |
| id                      | name   | studyTime | studyGrade |  |
| 3                       | 概率论    | 64        | 4          |  |
| 4                       | 太极拳    | 32        | 2          |  |
| 5                       | 计算机基础  | 64        | 4          |  |
| 7                       | 初中数学   | 32        | 2          |  |
| 9                       | 高中数学   | 64        | 4          |  |
| 10                      | 数据结构实训 | 32        | 2          |  |
| 12                      | 操作系统   | 64        | 4          |  |

原课程表

| 对象 课程 @学生管理 (MySQL) - 表 |        |           |            |
|-------------------------|--------|-----------|------------|
| 开始事务 文本 筛选 排序 导入 导出     |        |           |            |
| id                      | name   | studyTime | studyGrade |
| 3                       | 概率论    | 64        | 4          |
| 4                       | 体育-太极拳 | 32        | 2          |
| 5                       | 计算机基础  | 64        | 4          |
| 7                       | 初中数学   | 32        | 2          |
| 9                       | 高中数学   | 64        | 4          |
| 10                      | 数据结构实训 | 32        | 2          |
| 12                      | 操作系统   | 64        | 4          |

刷新后的课程表（修改成功）

（2）输入


原课程号：2

课程号：15          课程名：政治

学时：32          学分：2

点击确认

| 修改课程... |    |
|---------|----|
| 原课程号：   | 2  |
| 课程号：    | 15 |
| 课程名：    | 政治 |
| 学时：     | 32 |
| 学分：     | 2  |
| 确认      |    |

| 结果  |                  |
|---|------------------|
|  | 没有课程号为2的课程，修改失败！ |
| 确定  |                  |

| 开始事务 文本 筛选 排序 导入 导出 |        |           |            |  |
|---------------------|--------|-----------|------------|--|
| id                  | name   | studyTime | studyGrade |  |
| 3                   | 概率论    | 64        | 4          |  |
| 4                   | 体育-太极拳 | 32        | 2          |  |
| 5                   | 计算机基础  | 64        | 4          |  |
| 7                   | 初中数学   | 32        | 2          |  |
| 9                   | 高中数学   | 64        | 4          |  |
| 10                  | 数据结构实训 | 32        | 2          |  |
| 12                  | 操作系统   | 64        | 4          |  |

课程表（没有课程号为 2 的课程）

### （3）输入

原课程号：4

课程号：15                  课程名：政治

学时：32                  学分：你好

点击确认

修改课程...

原课程号：

4

课程号：

15

课程名：

政治

学时：

32

学分：

你好

确认

提示

!

原课程号or课程号or学时or学分输入的不是整形课程号，请重新输入！

确定

### （4）输入

原课程号：4

课程号：15                  课程名：无输入

学时：32                  学分：2

点击确认

## 17. 查询所有课程信息

| 课程号 | 课程名    | 学时 | 学分 |
|-----|--------|----|----|
| 3   | 概率论    | 64 | 4  |
| 4   | 体育-太极拳 | 32 | 2  |
| 5   | 计算机基础  | 64 | 4  |
| 7   | 初中数学   | 32 | 2  |
| 9   | 高中数学   | 64 | 4  |
| 10  | 数据结构实训 | 32 | 2  |
| 12  | 操作系统   | 64 | 4  |

界面

源代码:

```
package view;

import dao.CourseDao;
import model.Course;

import javax.swing.*;
import javax.swing.table.DefaultTableCellRenderer;
import java.awt.*;
import java.sql.SQLException;
import java.util.List;

/**
 * @author Administrator
 * 窗口：查询所有的课程并显示
 */
public class SearchAll extends JFrame {
    CourseDao courseDao = new CourseDao();

    public SearchAll() throws SQLException {
        //设置窗口弹出位置居中
        this.setLocationRelativeTo(null);
    }
}
```

---

```
//设置窗口名称
this.setTitle("查询所有课程");
//窗口布局
this.setLayout(new BorderLayout());
JPanel jPanel = new JPanel();
//面板布局
jPanel.setLayout(new BorderLayout());
//利用表格来展示查询到的数据库结果
JTable jTable = getTable();
//表格分成表头和表尾来添加到面板
jPanel.add(jTable.getTableHeader(), BorderLayout.NORTH);
jPanel.add(jTable, BorderLayout.CENTER);
jPanel.add(jTable);
this.add(jPanel);
//窗口自动调整大小
this.pack();
//设置窗口可见
this.setVisible(true);
}

/**
 * 从数据库中读取数据，放在一个表格里，方法返回一个表格
 *
 */
public JTable getTable() throws SQLException {
    List<Course> lists;
    //从数据库查询所有课程，返回一个 List
    lists = courseDao.searchAll();
    //表格的列名
    Object[] columnNames = {"课程号", "课程名", "学时", "学分"};
    //表格的数据区域（表尾）
    Object[][] data = new Object[lists.size()][4];
    for (int i = 0; i < lists.size(); i++) {
        data[i][0] = lists.get(i).getId();
        data[i][1] = lists.get(i).getName();
        data[i][2] = lists.get(i).getStudyTime();
        data[i][3] = lists.get(i).getStudyGrade();
    }
    JTable jTable = new JTable(data, columnNames);
    //设置表格数据居中显示
    DefaultTableCellRenderer defaultTableCellRenderer = new DefaultTableCellRenderer();
    defaultTableCellRenderer.setHorizontalAlignment(SwingConstants.CENTER);
    jTable.setDefaultRenderer(Object.class, defaultTableCellRenderer);
    return jTable;
}
```

---

## 四、总结

通过本次实验，掌握了数据库的实际应用

通过本次实验，加强了数据库增删改查语句的运用

通过本次实验，巩固了 java 窗口 swing 技术