

## 实践教学

计通学院

2019 年春季学期

# Java 课程设计

题 目: 学生成绩管理系统

专业班级: 计算机科学与技术 1 班

姓 名: \_\_\_\_\_

学 号: \_\_\_\_\_

指导教师: \_\_\_\_\_

成 绩: \_\_\_\_\_

## 目录

摘要.....	1
序言.....	2
1、 需求分析.....	3
2、 设计思路.....	4
3、 详细设计.....	5
3.1 登录界面 Login 类.....	5
3.2 返回数据库连接的 Connection 的主类 Main.....	8
3.3 返回查询结果 ReturnQueryResult 类.....	9
3.4 教师客户端 TeacherMenu 类.....	10
3.5 学生客户端类.....	29
4、 运行调试与分析讨论.....	35
(1) 登录.....	35
(2) 教师客户端主页面.....	36
(3) 学生客户端主页面.....	36
(4) 按学号查询.....	37
(5) 按姓名查找.....	37
(6) 修改成绩.....	38
(7) 彻底删除学生信息.....	39
(8) 修改学生信息.....	40
(9) 添加学生.....	41
(10) 成绩统计.....	41
(11) 个人信息.....	42
(12) 学生客户端主页面.....	43
(13) 修改信息.....	43
(14) 修改密码.....	44
5、 数据结构基本算法实现.....	45
5.1 利用 SQL 查询语句 select from 返回查询结果.....	45
5.2 用 row_number() over(order by )实现按照分数排名, sum()实现求学生成绩总分, case when then end 统计学生各科成绩。 .....	45
6、 课程设计总结.....	49

参考文献.....	50
致谢.....	51

## 摘要

在现今信息时代，生活速度的加快，使得人们越来越向信息化、数字化发展。随着学校的规模不断扩大，学生数量急剧增加，有关学生的各种信息量也成倍增长，尤其是学生的考试成绩数据。面对庞大的学生的成绩，需要有学生成绩管理系统来提高学生管理工作的效率。鉴于此提出了学生成绩管理系统，学生管理系统是计算机对学生档案信息进行管理，具有手工管理无可比拟的优点，如索检迅速、查找方便、可靠性高、存储量大等有点。现在我国的大中专院校的学生成绩管理水平正在不断提高，停留在纸介质基础上的管理手段早已被时代淘汰，因为它浪费了许多的人力和物力。本系统使用 Swing + MYSQL + IntelliJ IDEA 开发。为管理人员提供充足的信息和快捷的查询手段，提高学校学籍管理的效率。

**关键词：** Swing，MYSQL，学生成绩管理系统

## 序言

管理信息系统正在向着网络化、智能化和集成化等趋势发展。学生成绩管理系统是为了更好的管理学生考试成绩而开发的数据管理软件。它对于一个学校是不可缺少的重要部分，它的内容对于学校的决策者和管理者来说都至关重要。学生成绩管理管理系统为用户提供充足的信息和快捷的查询手段，实现学生基本信息、成绩的录入，删除，查询，维护以及成绩的统计分析等几方面的功能，是现实问题的迫切要求。

本系统开发的总体任务是实现学生成绩管理的系统化、规范化、自动化。达到提高学生成绩管理效率的目的。与传统管理方法相比有明显的优点：查找方便，可靠性高，保密性好，成本低。彻底改变了以前繁杂的管理模式，实现全面的、相对集中的、职能化的信息综合管理。

计算机被用到信息管理系统的的环境正是适应了当今时代飞速发展的信息时代。人们深刻的认识到了计算机功能的强大，对于复杂的信息管理，计算机充分发挥着它的优越性。检索迅速、查找方便、可靠性高、存储量大、保密性好、寿命长、成本低，这些优点极大地减轻了学院教学人员的工作量，缩小开支，提高了学生档案管理的效率和准确性，能够合理的安排时间，学生能够尽快的知道自己的考试成绩。同时，学生管理系统的应用也为今天的教育在未来市场的竞争力有所提高。

# 1、需求分析

## 1.1 需求分析概述

需求分析是开发软件系统的重要环节，是系统开发的第一步和基础环节。通过需求分析充分认识系统的目标、系统的各个组成部分、各部分的任务职责、工作流程、工作中使用的各种数据及数据结构、各部门的业务关系和数据流程等，为系统设计打下基础。

而在一些学校机构，随着学生数量的不断增加，学生的信息不断增多，人工管理信息的难度也越来越大。而且效率也是很低的。所以如何自动高效地管理信息是这些年来许多人所研究的。

随着这些年电脑计算机的速度质的提高，成本的下降，IT 互联网大众趋势的发展。我们使用电脑的高效率才处理数据信息成为可能。学生学籍管理系统的出现，正是管理人员与信息数据，计算机的进入互动时代的体现。友好的人机交互模式，清晰简明的图形界面，高效安全的操作使得我们对成千上万的信息的管理得心应手。

## 1.2 系统需求分析

- (1) 实现数据的录入（添加）、删除、修改
- (2) 按姓名或学号查询
- (3) 按照成绩排序，按照成绩段统计汇总

## 2、设计思路

- (1) 系统通过窗口显示并以菜单方式工作,;
- (2) 能查询、修改和删除文件中考生信息;
- (3) 能够显示全部的考生信息;

根据上述系统功能分析,按照结构化程序设计的要求,得到系统的功能结构图,如图所示。

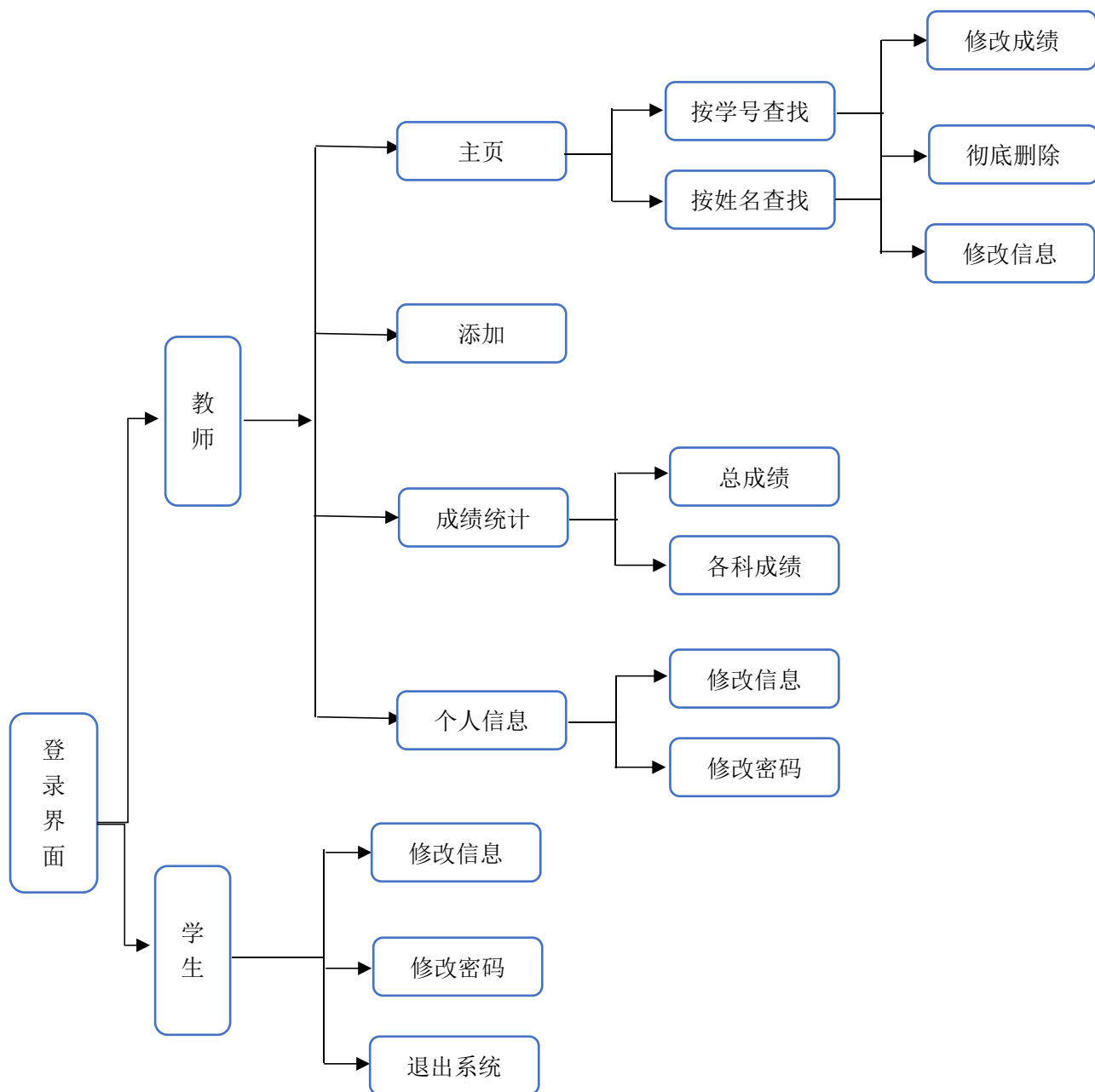


图 2.1 简易流程图

## 3、详细设计

### 3.1 登录界面 Login 类

通过选择教师或学生登录，从两张表查找各自的密码检查是否正确，从而进入不同的客户端。实现代码如下：

```
public class Login extends JFrame {

    Login father = this;
    JTextField user = null;
    JPasswordField password = null;
    JButton button1 = null;
    String flag = "teacher";

    Login() {
        this.setSize(500, 420);
        this.setLocationRelativeTo(null);
        this.setTitle("学生成绩管理系统");
        this.setResizable(false);
        init();
        this.setVisible(true);
        password.requestFocus(); //让密码输入框获得焦点
    }

    public void login() { //进入主菜单事件
        String password_true = null;
        String where = " where T_ID = ";
        if (flag.equals("teacher")) {
            where = " where T_ID = ";
        } else {
            where = " where S_ID = ";
        }
        try {
            String password_sql = "select passcode from ssms." + flag +
where + Integer.parseInt(user.getText()) + ";";
            Connection conn = Main.sendConnection();
            PreparedStatement stmt =
conn.prepareStatement(password_sql);
            ResultSet result = stmt.executeQuery();
            result.next();
            password_true = result.getString("passcode");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```



```

    }
    char[] pswd = password.getPassword();
    if (String.valueOf(pswd).equals(password_true)) { //密码
        JOptionPane.showMessageDialog(father, "登陆成功!");
        if (flag.equals("teacher")) {
            TeacherMenu teacherMenu = new TeacherMenu(); //创建主菜单界面

            teacherMenu.sendObject(father); //传递主窗口引用
            teacherMenu.sendID(user.getText()); //传递输入的账号ID 用于数据库的操作等
            teacherMenu.init();
        } else {
            StudentMenu studentMenu = new StudentMenu();
            studentMenu.sendObject(father);
            studentMenu.sendID(user.getText());
            studentMenu.init();
        }
        father.setVisible(false); //进入主菜单界面时隐藏登录界面
    } else {
        JOptionPane.showMessageDialog(father, "账号或密码错误!", "错误", JOptionPane.WARNING_MESSAGE);
    }
}

public void init() {

    setLayout(new GridLayout(2, 1, 0, 0));
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    JPanel panell = new JPanel() { //登录界面上半部分
        private static final long serialVersionUID = 1L;
        ImageIcon image = new ImageIcon("Image\\up.png"); //登录界面上部图片

        public void paintComponent(Graphics g) {
            super.paintComponent(g);
            g.drawImage(image.getImage(), 0, 0, this.getWidth(), this.getHeight(), this);
        }
    };

    panell.setLayout(null);
    JPanel head = new JPanel() { //头像
        private static final long serialVersionUID = 1L;
        ImageIcon icon = new ImageIcon("Image\\flower.jpg");

```

```

        public void paintComponent(Graphics g) {
            super.paintComponent(g);
            g.drawImage(icon.getImage(), 0, 0, this.getWidth(),
this.getHeight(), this);
        }
    };
    head.setBounds(200, 70, 100, 100);
    panel1.add(head);
    JPanel panel2 = new JPanel(new GridLayout(2, 1, 0, 0)); //登录
界面下半部分
    add(panel1);
    add(panel2);
    JLabel label1 = new JLabel("账    号: ");
    JLabel label2 = new JLabel("密    码: ");
    user = new JTextField("10001", 15); //账号输入框
    password = new JPasswordField(15); //密码输入框
    password.addActionListener(e -> login()); //密码框添加回车登录
事件, 进入主菜单界面
    JPanel panel3 = new JPanel(new GridLayout(2, 1, 0, 0)); //放置
用户名和密码及其输入框
    JPanel panel4 = new JPanel(new FlowLayout(FlowLayout.CENTER, 0,
16)); //放置用户名及其输入框
    JPanel panel5 = new JPanel(new FlowLayout(FlowLayout.CENTER, 0,
16)); //放置密码及其输入框
    panel3.add(panel4);
    panel3.add(panel5);
    panel4.add(label1);
    panel4.add(user); //添加用户名输入框
    panel5.add(label2);
    panel5.add(password); //添加密码输入框
    panel2.add(panel3);
    button1 = new JButton("登    录");
    JButton button2 = new JButton("退    出");
    button1.setFocusPainted(false); //不绘制焦点
    button2.setFocusPainted(false);

    try { //改变成 win10 风格
        UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
    } catch (ClassNotFoundException | InstantiationException |
IllegalAccessException
        | UnsupportedLookAndFeelException e) {
        e.printStackTrace();
    }
}

```

```

    }
    button1.addActionListener(e -> login());
    button2.addActionListener(e -> System.exit(0)); //lambda 表达式
    JPanel panel_choice = new JPanel();
    JRadioButton radioButton_1 = new JRadioButton("教师", true); //
默认选择教师登录
    JRadioButton radioButton_2 = new JRadioButton("学生");
    radioButton_1.setFocusPainted(false);
    radioButton_2.setFocusPainted(false);
    radioButton_1.addActionListener(e -> flag = "teacher");
    radioButton_2.addActionListener(e -> flag = "student");
    ButtonGroup group = new ButtonGroup();
    group.add(radioButton_1);
    group.add(radioButton_2);
    panel_choice.add(radioButton_1);
    panel_choice.add(radioButton_2);
    JPanel panel6 = new JPanel(new GridLayout(3, 1, 0, 0));
    panel2.add(panel6);
    panel6.add(panel_choice);
    JPanel panel_button = new JPanel(new
FlowLayout(FlowLayout.CENTER, 30, 5));
    panel_button.add(button1);
    panel_button.add(button2);
    panel6.add(panel_button);
    panel6.add(new JPanel());
}
}

```

### 3.2 返回数据库连接的 Connection 的主类 Main 类

将连接数据库的操作封装在主类中，定义一个静态方法，该方法返回一个连接数据库的 **Connection** 类型的引用，以便随时方便连接数据库。

```

public class Main {
    static Connection sendConnection() {
        String driver = "com.mysql.cj.jdbc.Driver";
        //URL 指向要访问的数据库名
        String url =
"jdbc:mysql://localhost:3306/world?serverTimezone=UTC";
        //MySQL 配置时的用户名
        String user = "root";
        //MySQL 配置时的密码
        String password = "n3483226";
        Connection conn = null;
        try {

```

```

        Class.forName(driver);
        conn = DriverManager.getConnection(url, user, password);

    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return conn;
}

public static void main(String[] args) {

    new Login();
}
}

```

### 3.3 返回查询结果 ReturnQueryResult 类

将数据库的查询操作封装成一个类，定义一个静态方法，通过传入不同的 SQL 语句，以字符串二维数组的方式返回查询结果。

```

class ReturnQueryResult {
    public static String[][] send(String sql) {
        String[][] result = null;
        Connection conn = null;
        Statement statement = null;
        try {
            conn = Main.sendConnection();
            statement = conn.createStatement();

            ResultSet rs = statement.executeQuery(sql);
            ResultSetMetaData rsmd = rs.getMetaData();
            int column = rsmd.getColumnCount(); // 获取列数
            rs.last();
            int row = rs.getRow(); // 获取行数
            rs.beforeFirst();
            result = new String[row][column];
            int count = 0;
            while (rs.next()) {
                for (int i = 1; i <= column; i++) {
                    result[count][i - 1] = rs.getString(i);
                }
                count++;
            }
        }
    }
}

```

```

        } catch (SQLException e) {
            e.printStackTrace();
        }
        return result;
    }
}

```

### 3.4 教师客户端 TeacherMenu 类

将教师的所有功能封装在一个类中。详细代码如下

```

public class TeacherMenu extends JFrame {

    TeacherMenu father = this;
    Login send = null;
    JPanelRight flag = null;
    String input_ID = null;
    JPanel panel_top = null;
    JPanelLeft panel_left = null;
    JPanelRight panel_right = null;
    JPanelLeft showmessage = null;
    static final CardLayout cl = new CardLayout();

    public TeacherMenu() {
        this.setLayout(new GridBagLayout()); //主菜单界面为网格包布局
        this.setSize(1200, 800);
        this.setLocationRelativeTo(null);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    //获取登录界面对象的引用
    void sendObject(Login o) {
        send = o;
    }

    //获取登录时用户输入的账号
    void sendID(String ID) {
        input_ID = ID;
    }

    public void init() { //在主菜单界面添加 panel
        panel_top = new JPanel();
        JLabel label_title = new JLabel("欢迎使用学生成绩管理系统");
    }
}

```

```

        label_title.setFont(new Font("宋体", Font.ROMAN_BASELINE,
40));
        panel_top.add(label_title);

        panel_top.setBackground(new Color(0xAFEEEE));
        this.add(panel_top, new MyGridBagConstraints(0, 0, 2, 1).
            setFill(GridBagConstraints.BOTH).setIpad(200,
50).setWeight(100, 0));

        panel_right = new JPanelRight();
        this.add(panel_right, new MyGridBagConstraints(1, 1, 1,
1).setFill(GridBagConstraints.BOTH));

        panel_left = new JPanelLeft();
        this.add(panel_left, new MyGridBagConstraints(0, 1, 1, 1).
            setFill(GridBagConstraints.BOTH).setWeight(0,
100).setIpad(100, 300));
        this.setVisible(true);
    }

    class JPanelLeft extends JPanel { //放置左侧按钮面板

        JPanelLeft dad = this;

        public JPanelLeft() {
            showmessage = this;
            this.setLayout(new GridBagLayout());
            init();
        }

        JPanel panel_head = new JPanel();
        JPanel panel_tail = new JPanel();
        JButton button_search = new JButton("主页");
        JButton button_add = new JButton("添加");
        JButton button_score = new JButton("成绩统计");
        JButton button_message = new JButton("个人信息");
        JButton button_quit = new JButton("退出系统");

        void init() {
            button_search.addActionListener(e -> cl.show(panel_right,
"home"));
            button_add.addActionListener(e -> cl.show(panel_right,
"insert"));
            button_score.addActionListener(e -> {

```

```

        flag.init();
        cl.show(panel_right, "score");});
        button_message.addActionListener(e -> cl.show(panel_right,
"message"));
        add(panel_head, new MyGridBagConstraints(0, 0, 1,
2).setInset(new Insets(5, 0, 5, 0)).
            setFill(GridBagConstraints.BOTH).setWeight(10,
10).setIpad(1, 1));
        add(button_search, new MyGridBagConstraints(0, 2, 1,
1).setInset(new Insets(5, 0, 5, 0)).
            setFill(GridBagConstraints.BOTH).setIpad(20,
10));
        add(button_add, new MyGridBagConstraints(0, 3, 1,
1).setInset(new Insets(5, 0, 5, 0)).
            setFill(GridBagConstraints.BOTH).setIpad(20,
10));
        add(button_score, new MyGridBagConstraints(0, 4, 1,
1).setInset(new Insets(5, 0, 5, 0)).
            setFill(GridBagConstraints.BOTH).setIpad(20,
10));
        add(button_message, new MyGridBagConstraints(0, 6, 1,
1).setInset(new Insets(5, 0, 5, 0)).
            setFill(GridBagConstraints.BOTH).setIpad(20,
10));
        add(panel_tail, new MyGridBagConstraints(0, 7, 1,
2).setInset(new Insets(5, 0, 5, 0)).
            setFill(GridBagConstraints.BOTH).setWeight(10,
30).setIpad(1, 1));
    }
}

```

class JPanelRight extends JPanel { //放置右侧 主页面板, 个人信息面板等

```

    JPanelRight parent = this;

    public JPanelRight() {
        flag = this;
        this.setBackground(Color.MAGENTA);
        this.setLayout(cl); //CardLayout
        init();
    }

    void init() {

```

```

JPanelHome panel_home = new JPanelHome();
JPanelInsert panel_insert = new JPanelInsert();
JPanelScore panel_score = new JPanelScore();
PanelMessage panel_message = new PanelMessage();
add(panel_home, "home");
add(panel_insert, "insert");
add(panel_score, "score");
add(panel_message, "message");
}

//添加学生信息面板类
class JPanelInsert extends JPanelHome {
    JPanelInsert() {
        this.setLayout(new GridLayout(7, 2));
    }

    void init() {
        JPanel[] panels = new JPanel[7];
        for (int i = 0; i < panels.length; i++) {
            panels[i] = new JPanel();
        }
        JLabel title = new JLabel("添加学生信息: ");
        panels[0].setLayout(new FlowLayout(FlowLayout.LEFT));
        panels[0].add(title);
        JLabel[] jLabels = new JLabel[5];
        JTextField[] jTextFields = new JTextField[5];
        String[] label_text = {"学号", "姓名", "性别", "年龄",
"电话"};

        for (int i = 0; i < 5; i++) {
            jLabels[i] = new JLabel(label_text[i]);
            jTextFields[i] = new JTextField(20);
            panels[i + 1].add(jLabels[i]);
            panels[i + 1].add(jTextFields[i]);
        }
        JButton button_confirm = new JButton("添加");
        button_confirm.setPreferredSize(new Dimension(100,
30));

        button_confirm.addActionListener(e -> {
            Connection conn = Main.sendConnection();
            try {
                String sql_student = "insert into ssms.student
values (" + jTextFields[0].getText() + ", ' "
+ jTextFields[1].getText() + "', ' " +
jTextFields[2].getText() + "', "

```



```

        + jTextField3.getText() + ", ' " +
jTextField4.getText() + "', '123456')";
        PreparedStatement pstmt =
conn.prepareStatement(sql_student);
        pstmt.executeUpdate();
        for (int i = 1; i <= 4; i++) { //在 score 表添加
默认成绩 0 分
            String sql_score = "insert into ssms.score
values(" + jTextField0.getText() + ", 1000" + i + ", 0)";
            pstmt = conn.prepareStatement(sql_score);
            pstmt.executeUpdate();
        }
        JOptionPane.showMessageDialog(this, "添加成功!",
",
        "Succeed",
JOptionPane.INFORMATION_MESSAGE);
    } catch (Exception ex) {
        ex.printStackTrace();
        JOptionPane.showMessageDialog(this, "格式有误,
请重新输入!",
        "Error",
JOptionPane.WARNING_MESSAGE);
    }
});
panels[6].add(button_confirm);
for (int i = 0; i < panels.length; i++) {
    this.add(panels[i]);
}
}
}

```

//统计学生信息面板类

```
class JPanelScore extends JPanelHome {
```

```
    JPanelShow panel_show = new JPanelShow();
```

```
    CardLayout layout = (CardLayout)panel_show.getLayout();
```

```
    JPanelScore() {
```

```
        this.setLayout(new BorderLayout());
```

```
//        this.removeAll();
```

```
        JLabel title = new JLabel("成绩统计:");
```

```
        title.setFont(new Font("楷体", Font.BOLD, 30));
```

```
        this.add(title, BorderLayout.NORTH);
```

```

        this.add(panel_show, BorderLayout.CENTER);
        JPanelButton panel_button = new JPanelButton();
        panel_button.setPreferredSize(new Dimension(0, 100));
        this.add(panel_button, BorderLayout.SOUTH);
    }

    class JPanelButton extends JPanel {
        JPanelButton() {
            this.setLayout(new FlowLayout(FlowLayout.CENTER,
30, 40));

            String[] button = {"总成绩", "Java", "数据库原理",
"算法与数据结构", "C++"};
            JButton[] button_all = new JButton[5];
            for(int i=0; i<button.length; i++) {
                button_all[i] = new JButton(button[i]);
                this.add(button_all[i]);
            }

            button_all[0].addActionListener(e->layout.show(panel_show, "0"));

            button_all[1].addActionListener(e->layout.show(panel_show, "1"));

            button_all[2].addActionListener(e->layout.show(panel_show, "2"));

            button_all[3].addActionListener(e->layout.show(panel_show, "3"));

            button_all[4].addActionListener(e->layout.show(panel_show, "4"));
        }
    }

    class JPanelShow extends JPanel {
        JPanel[] panels = new JPanel[5];
        JPanelShow() {
            this.setLayout(new CardLayout());
            for(int i=0; i<panels.length; i++) {
                panels[i] = new JPanel(new BorderLayout());
                this.add(panels[i], i + "");
            }
            show_all();
            show_single(1, "10001");
            show_single(2, "10002");
            show_single(3, "10003");
            show_single(4, "10004");
        }
        void show_single(int no, String t_no) {

```

```

        String[] head = {"学号", "姓名", "成绩", "排名"};
        String sql = "select r.*, row_number() over(order by
'subject') as row_rank\n" +
            "from(\n" +
            "select score.S_ID, student.name,\n" +
            "sum(score.grade) as 'subject'\n" +
            "from ssms.score, ssms.student\n" +
            "where score.S_ID = student.S_ID and
score.T_ID = " + t_no +
            " group by score.S_ID\n" +
            "order by score.grade desc) r";
        String[][] result = ReturnQueryResult.send(sql);
        MyTable table = new MyTable(result, head);
        JScrollPane jsp = new JScrollPane(table);
        panels[no].add(jsp, BorderLayout.CENTER);
        String[] head_1 = {"分数段", "优秀(90~100)", "良好
(80~90)", "中等(70~80)", "及格(60~70)", "不及格(0~60)"};
        String[][] result_1 = new String[1][6];
        result_1[0][0] = "人数";
        String sql_1 = "select count(case when
score.grade >= 90 then 1 end) as '优秀',\n" +
            "count(case when score.grade < 90 and
score.grade >= 80 then 1 end) as '良好',\n" +
            "count(case when score.grade < 80 and
score.grade >= 70 then 1 end) as '中等',\n" +
            "count(case when score.grade < 70 and
score.grade >= 60 then 1 end) as '及格',\n" +
            "count(case when score.grade < 60 then 1 end)
as '不及格'\n" +
            "from ssms.score\n" +
            "where score.T_ID = " + t_no;
        String[][] resultSet =
ReturnQueryResult.send(sql_1);
        for(int i = 1; i<6; i++) {
            result_1[0][i] = resultSet[0][i-1];
        }
        MyTable table_1 = new MyTable(result_1, head_1);
        JScrollPane jsp_1 = new JScrollPane(table_1);
        jsp_1.setPreferredSize(new Dimension(0, 100));
        panels[no].add(jsp_1, BorderLayout.SOUTH);
    }

    void show_all() {

```

```

        String[] head = {"学号", "姓名", "Java", "数据库原理", "算法与数据结构", "C++", "总分", "平均分", "排名"};
        String sql = "select r.*, row_number() over (order by
' 平均分') as row_rank\n" +
                    "from(\n" +
                    "select score.S_ID, student.name,\n" +
                    "sum(case when T_ID = 10001 then score.grade
else 0 end) as 'java',\n" +
                    "sum(case when T_ID = 10002 then score.grade
else 0 end) as '数据库',\n" +
                    "sum(case when T_ID = 10003 then score.grade
else 0 end) as '算法',\n" +
                    "sum(case when T_ID = 10004 then score.grade
else 0 end) as 'C++',\n" +
                    "sum(score.grade) as '总分',\n" +
                    "format(sum(score.grade)/count(*), 2) as '
平均分'\n" +
                    "from ssms.score, ssms.student\n" +
                    "where score.S_ID = student.S_ID\n" +
                    "group by score.S_ID\n" +
                    "order by 平均分 desc) r";
        String[][] result = ReturnQueryResult.send(sql);
        MyTable table = new MyTable(result, head);
        JScrollPane jsp = new JScrollPane(table);
        panels[0].add(jsp, BorderLayout.CENTER);
        String[] head_1 = {"分数段", "优秀(90~100)", "良好
(80~90)", "中等(70~80)", "及格(60~70)", "不及格(0~60)"};
        String[][] result_1 = new String[1][6];
        result_1[0][0] = "人数";
        String sql_1 = "select count(case when ave >= 90 then
1 end) as '优秀',\n" +
                    "count(case when ave < 90 and ave >= 80 then
1 end) as '良好',\n" +
                    "count(case when ave < 80 and ave >= 70 then
1 end) as '中等',\n" +
                    "count(case when ave < 70 and ave >= 60 then
1 end) as '及格',\n" +
                    "count(case when ave < 60 then 1 end) as '
不及格'\n" +
                    "from(select score.S_ID, student.name,\n" +
                    "sum(case when T_ID = 10001 then score.grade
else 0 end) as 'java',\n" +

```

```

        "sum(case when T_ID = 10002 then score. grade
else 0 end) as '数据库',\n" +
        "sum(case when T_ID = 10003 then score. grade
else 0 end) as '算法',\n" +
        "sum(case when T_ID = 10004 then score. grade
else 0 end) as 'C++',\n" +
        "sum(score. grade) as '总分',\n" +
        "format(sum(score. grade)/count(*), 2) as
ave,\n" +
        "format(@r := @r + 1, 0) as '排名' \n" +
        "from ssms. score, ssms. student, (select
@r := 0) alias\n" +
        "where score. S_ID = student. S_ID\n" +
        "group by score. S_ID) t;";
String[][] resultSet =
ReturnQueryResult. send(sql_1);
for(int i = 1; i<6; i++) {
    result_1[0][i] = resultSet[0][i-1];
}
MyTable table_1 = new MyTable(result_1, head_1);
JScrollPane jsp_1 = new JScrollPane(table_1);
jsp_1.setPreferredSize(new Dimension(0, 100));
panels[0].add(jsp_1, BorderLayout. SOUTH);
    }
}
}

```

//个人信息面板类

class PanelMessage extends JPanel { //个人信息面板

```

    PanelMessage() {
        this.setLayout(new GridBagLayout());
        init();
    }

```

```

    void init() {
        JPanel panel_top = new JPanel(new GridLayout(10, 1));
        JLabel l_message = new JLabel("个人信息:");
        JLabel[] labels = new JLabel[8];
        String[] label_name = {"教工号:", "姓名:", "性别:", "
年龄:", "电话:", "课程号:", "密码:", "课程名:"};
        String sql_message = "SELECT teacher.*, course.Cname
FROM ssms. teacher, ssms. course where T_ID = " + input_ID
        + " and teacher. subject = course. Cno;";
    }

```

```

        String[][] result =
ReturnQueryResult.send(sql_message);
        for (int i = 0; i < labels.length; i++) {
            labels[i] = new JLabel("        " + label_name[i] +
"    " + result[0][i]);
        }
        panel_top.add(l_message);
        for (JLabel l : labels) {
            if (l == labels[6])//不显示密码
                continue;
            panel_top.add(l);
        }
        TPanelBottom panel_bottom = new TPanelBottom();
        this.add(panel_top, new MyGridBagConstraints(0, 0, 10,
5).setFill(GridBagConstraints.BOTH)
            .setWeight(10, 100));
        this.add(panel_bottom, new MyGridBagConstraints(0, 16,
10, 1).setFill(GridBagConstraints.BOTH)
            .setWeight(10, 1));
    }

    class TPanelBottom extends JPanel { //放置两个按钮

        TPanelBottom() {
            this.setLayout(new FlowLayout(FlowLayout.CENTER,
100, 30));

            init();
        }

        public void init() {
            JButton button_alter = new JButton("修改信息");
            JButton button_password = new JButton("修改密码");
            button_alter.setFocusPainted(false);
            button_password.setFocusPainted(false);
            button_alter.setPreferredSize(new Dimension(100,
30));

            button_password.setPreferredSize(new
Dimension(100, 30));
            button_alter.addActionListener(e -> new
AlterDialog(father)); //弹出修改个人信息对话框
            button_password.addActionListener(e -> new
PasswordDialog(father));
            this.add(button_alter);
            this.add(button_password);
        }
    }

```

```

    }
}

class AlterDialog extends JDialog { //修改个人信息对话框

    AlterDialog(JFrame parents) {
        super(parents);
        this.setSize(350, 460);
        this.setLocationRelativeTo(parents);
        init();
        this.setVisible(true);
    }

    void init() {
        Container c = this.getContentPane();
        c.setLayout(new GridLayout(10, 1, 0, 15));
        JLabel label_age = new JLabel("请输入年龄:");
        this.add(label_age);
        JTextField text_age = new JTextField();
        this.add(text_age);
        JLabel label_phone = new JLabel("请输入电话:");
        this.add(label_phone);
        JTextField text_phone = new JTextField();
        this.add(text_phone);
        JButton bt_save = new JButton("保    存");
        Connection conn = Main.sendConnection();
        String sql = "update ssms.teacher set teacher.age=?,
teacher.phone=? where teacher.T_ID=?";
        bt_save.addActionListener(e -> {
            PreparedStatement pstmt = null;
            try {
                pstmt = conn.prepareStatement(sql);
                pstmt.setInt(1,
Integer.parseInt(text_age.getText()));
                pstmt.setString(2, text_phone.getText());
                pstmt.setInt(3,
Integer.parseInt(input_ID));
                pstmt.executeUpdate();
                parent.init();
                cl.show(panel_right, "message");

                showmessage.button_message.requestFocus();
                JOptionPane.showMessageDialog(this, "修改
成功! ",

```

```

                                "Succeed",
JOptionPane.INFORMATION_MESSAGE);
                                } catch (SQLException | NumberFormatException
ex) {
                                ex.printStackTrace();
                                JOptionPane.showMessageDialog(this, "格式
有误，请重新输入！",
                                "Error",
JOptionPane.WARNING_MESSAGE);
                                }
                                });
                                this.add(bt_save);

this.setDefaultCloseOperation(JDialog.HIDE_ON_CLOSE);
    }
}

```

```

class PasswordDialog extends JDialog { //修改密码对话框

```

```

    PasswordDialog(JFrame parents) {
        super(parents);
        this.setSize(300, 260);
        this.setLocationRelativeTo(parents);
        init();
        this.setVisible(true);
    }

    void compare(String s1, String s2) throws Exception {
        if (!s1.equals(s2))
            throw new Exception("两次输入的密码不同!");
    }

    void init() {
        Container c = this.getContentPane();
        c.setLayout(new GridLayout(5, 1, 0, 15));
        JLabel label_1 = new JLabel("请输入你的新密码:");
        this.add(label_1);
        JTextField text_1 = new JTextField();
        this.add(text_1);
        JLabel label_2 = new JLabel("请再次输入你的密码:");
        this.add(label_2);
        JTextField text_2 = new JTextField();
        this.add(text_2);
        JButton bt_save = new JButton("保    存");
    }
}

```



```

        Connection conn = Main.sendConnection();
        String sql = "update ssms.teacher set
teacher.passcode =? where teacher.T_ID=?;";
        bt_save.addActionListener(e -> {
            PreparedStatement pstmt = null;
            try {
                pstmt = conn.prepareStatement(sql);
                try {
                    compare(text_1.getText(),
text_2.getText());

                    pstmt.setString(1, text_1.getText());
                    pstmt.setInt(2,
Integer.parseInt(input_ID));

                    pstmt.executeUpdate();
                    JOptionPane.showMessageDialog(this, "
修改成功!", "Succeed", JOptionPane.INFORMATION_MESSAGE);
                } catch (Exception ex) {
                    JOptionPane.showMessageDialog(this, "
两次输入的密码不同!", "Error", JOptionPane.WARNING_MESSAGE);
                }
            } catch (SQLException ex) {
                ex.printStackTrace();
                JOptionPane.showMessageDialog(this, "格式
有误, 请重新输入!", "Error", JOptionPane.WARNING_MESSAGE);
            }
        });
        this.add(bt_save);

this.setDefaultCloseOperation(JDialog.HIDE_ON_CLOSE);
    }
}

```

//主页面板类

class JPanelHome extends JPanel { //主页面板

```

    JPanel panel_top = null;
    JPanelHome dd = this;
    String sign_id = "";
    DownPanel down = null;

```

```

    public JPanelHome() {
        this.setLayout(new BorderLayout());
        init();
    }

```

```

    }

    void init() {
        panel_top = new JPanel(new GridLayout(1, 2));
        panel_top.setPreferredSize(new Dimension(0, 80));
        panel_top.setBackground(Color.pink);
        JPanel id = new JPanel(new FlowLayout(FlowLayout.CENTER,
10, 30));
        JPanel name = new JPanel(new FlowLayout(FlowLayout.LEFT,
10, 30));

        JTextField input_id = new JTextField(20);
        id.add(input_id);
        JButton button_id = new JButton("按学号查询");
        id.add(button_id);
        JTextField input_name = new JTextField(20);
        name.add(input_name);
        JButton button_name = new JButton("按姓名查询");
        name.add(button_name);
        panel_top.add(id);
        panel_top.add(name);
        add(panel_top, BorderLayout.NORTH);

        button_id.addActionListener(e -> {
            String sql = "SELECT student.*, course.Cname,
score.grade, teacher.name FROM" +
                " ssms.student, ssms.course, ssms.score,
ssms.teacher" +
                " where student.S_ID=" + input_id.getText()
+
                " and student.S_ID = score.S_ID and
score.T_ID = teacher.T_ID and course.Cno = teacher.subject;";
            try {
                search(sql);
            } catch (Exception e1) {
                e1.printStackTrace();
            }
        });
        button_name.addActionListener(e -> {
            String sql = "SELECT student.*, course.Cname,
score.grade, teacher.name FROM" +
                " ssms.student, ssms.course, ssms.score,
ssms.teacher" +

```

```

        " where student.name = '" +
input_name.getText() +
        "' and student.S_ID = score.S_ID and
score.T_ID = teacher.T_ID and course.Cno = teacher.subject;";
        try {
            search(sql);
        } catch (Exception e1) {
            e1.printStackTrace();
        }
    });
}

class DownPanel extends JPanel { //放置查询结果面板

    DownPanel() {
        this.setLayout(new GridLayout(3, 1));
    }

    void init(String sql) {
        String[][] result = null;
        try {
            result = ReturnQueryResult.send(sql);
            sign_id = result[0][0];
        } catch (Exception e) {
            dd.removeAll();
            dd.init();
        }
        String[][] result_1 = new String[1][5];
        try {
            for (int i = 0; i <= 4; i++) {
                result_1[0][i] = result[0][i];
            }
        } catch (Exception e) {
            e.printStackTrace();
            dd.removeAll();
            dd.init();
        }

        String[] head_1 = {"学号", "姓名", "性别", "年龄",
"电话"};

        MyTable table_1 = new MyTable(result_1, head_1);
        JScrollPane sp_1 = new JScrollPane(table_1);
        String[][] result_2 = null;
        try {

```

```

        result_2 = new String[result.length][3];
        for (int i = 0; i < result_2.length; i++) {
            for (int j = 0; j < result_2[i].length; j++)
            {
                result_2[i][j] = result[i][j + 6];
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
        result_2 = new String[0][0];
    }

    String[] head_2 = {"科目", "分数", "老师"};
    MyTable table_2 = new MyTable(result_2, head_2);
    JScrollPane sp_2 = new JScrollPane(table_2);
    this.add(sp_1);
    this.add(sp_2);
    JPanel panel_bottom = new JPanel(new
FlowLayout(FlowLayout.CENTER, 30, 10));
    JButton button_alter = new JButton("修改信息");
    button_alter.addActionListener(e -> {
        JDialog alter = new JDialog(father);
        alter.setSize(new Dimension(400, 500));
        alter.setLocationRelativeTo(father);
        Container c = alter.getContentPane();
        c.setLayout(new GridLayout(10, 1, 0, 15));
        JLabel label_age = new JLabel("请输入年龄:");
        alter.add(label_age);
        JTextField text_age = new JTextField();
        alter.add(text_age);
        JLabel label_phone = new JLabel("请输入电话:");
        alter.add(label_phone);
        JTextField text_phone = new JTextField();
        alter.add(text_phone);
        JButton bt_save = new JButton("保    存");
        Connection conn = Main.sendConnection();
        String sql_alter = "update ssms.student set
student.age=?, student.phone=? where student.S_ID=?;";
        bt_save.addActionListener(l -> {
            PreparedStatement pstmt = null;
            try {
                pstmt =
conn.prepareStatement(sql_alter);

```

```

                                pstmt.setInt(1,
Integer.parseInt(text_age.getText()));
                                pstmt.setString(2,
text_phone.getText());
                                pstmt.setInt(3,
Integer.parseInt(sign_id));
                                pstmt.executeUpdate();
                                //修改信息后刷新界面,更新数据
                                update_page();
                                JOptionPane.showMessageDialog(alter,
"修改成功!", "Succeed",

JOptionPane.INFORMATION_MESSAGE);
                                } catch (SQLException |
NumberFormatException ex) {
                                ex.printStackTrace();
                                JOptionPane.showMessageDialog(alter,
"格式有误, 请重新输入!",
                                "Error",
JOptionPane.WARNING_MESSAGE);
                                }
                                });
                                alter.add(bt_save);

alter.setDefaultCloseOperation(JDialog.HIDE_ON_CLOSE);
                                alter.setVisible(true);
                                });

                                JButton button_alter_score = new JButton("修改成绩
");
                                button_alter_score.addActionListener(e -> {
                                String sql_cname = "SELECT course.Cname FROM
ssms.course where course.Cno = " +
                                "( select teacher.subject from
ssms.teacher where teacher.T_ID = " + input_ID + " );";
                                String[][] Cname =
ReturnQueryResult.send(sql_cname);
                                String flag =
JOptionPane.showInputDialog(parent, "请输入" + Cname[0][0] + "成绩:",
                                "修改成绩",
JOptionPane.PLAIN_MESSAGE);
                                if (flag != null && flag != "") {
                                String sql_write_score = "update ssms. score
set score.grade = " + Integer.parseInt(flag)

```

```

+ " where score.S_Id = " + sign_id
+ " and score.T_ID = " + input_ID;
        Connection conn = Main.sendConnection();
        try {
            PreparedStatement pstmt =
conn.prepareStatement(sql_write_score);
            pstmt.executeUpdate();
            update_page();
            JOptionPane.showMessageDialog(parent,
"修改成功!",
                                "Succeed",
JOptionPane.INFORMATION_MESSAGE);
        } catch (SQLException e1) {
            e1.printStackTrace();
            JOptionPane.showMessageDialog(parent,
"修改失败!",
                                "Error",
JOptionPane.WARNING_MESSAGE);
        }
    });
    JButton button_delete = new JButton("彻底删除");
    button_delete.addActionListener(e -> { //删除学生所
有信息
        Object[] options = {"确认", "取消"};
        int flag = JOptionPane.showOptionDialog(parent,
"确认删除该生所有信息吗?", "标题",
            JOptionPane.YES_NO_CANCEL_OPTION,
JOptionPane.QUESTION_MESSAGE, null, options, options[0]);
        if (flag == JOptionPane.YES_OPTION) {
            Connection conn = Main.sendConnection();
            String sql_score = "delete from ssms.score
where S_ID = " + sign_id;
            String sql_student = "delete from
ssms.student where S_ID = " + sign_id;
            try {
                PreparedStatement pstmt =
conn.prepareStatement(sql_score);
                pstmt.executeUpdate();
                pstmt =
conn.prepareStatement(sql_student);
                pstmt.executeUpdate();
                update_page();

```

```

JOptionPane.showMessageDialog(parent,
"删除成功!",
                                "Succeed",
JOptionPane.INFORMATION_MESSAGE);
                                } catch (SQLException ex) {
                                    ex.printStackTrace();
JOptionPane.showMessageDialog(parent,
"删除错误!",
                                "Error",
JOptionPane.WARNING_MESSAGE);
                                }
                            }
                        });
                        panel_bottom.add(button_alter_score);
                        panel_bottom.add(button_delete);
                        panel_bottom.add(button_alter);
                        this.add(panel_bottom);
                    }
                }

void update_page() {
    //修改信息后刷新界面,更新数据
    String sql_updata = "SELECT student.*, course.Cname,
score.grade, teacher.name FROM" +
        " ssms.student, ssms.course, ssms.score,
ssms.teacher" +
        " where student.S_ID = " + sign_id + " and
student.S_ID = score.S_ID " +
        "and score.T_ID = teacher.T_ID and course.Cno =
teacher.subject;";
    try {
        search(sql_updata);
    } catch (Exception e1) {
        e1.printStackTrace();
        dd.removeAll();
        dd.init();
    }
}

void search(String sql) { //按学号查询事件
    this.removeAll();
    this.init();
    down = new DownPanel();
    try {

```

```

        down.init(sql);
    } catch (NullPointerException e) {
        e.printStackTrace();
        this.removeAll();
        this.init();
    }
    add(down, BorderLayout.CENTER);
    cl.show(panel_right, "message");//先跳转到其他面板，再
调回来，起到刷新的作用
    cl.show(panel_right, "home");
}
}
}
}

```

### 3.5 学生客户端类

将学生的所有功能封装在 **StudentMenu** 类中。详细代码如下：

```

public class StudentMenu extends JFrame {

    StudentMenu now = this;

    public StudentMenu() {
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setLayout(new BorderLayout());
        this.setSize(700, 600);
        this.setLocationRelativeTo(null);
        this.setResizable(false);
    }

    Login send = null;

    void sendObject(Login o) {
        send = o;
    }

    String input_ID = null;

    void sendID(String ID) {
        input_ID = ID;
    }

    public void init() {
        JPanelUp panel_up = new JPanelUp();

```



```

JPanelBottom panel_bottom = new JPanelBottom();
this.add(panel_up, BorderLayout.NORTH);
this.add(panel_bottom, BorderLayout.CENTER);
this.setVisible(true);
}

class JPanelUp extends JPanel {
    JPanelUp() {
        this.setPreferredSize(new Dimension(800, 480));
        this.setLayout(new GridLayout(1, 2, 0, 0));
        init();
    }

    public void init() {
        String sql_message = "select * from ssms.student where S_ID
= " + input_ID + ";";
        Connection conn = Main.sendConnection();
        PreparedStatement pstmt = null;
        ResultSet resultSet = null;
        String no = "", name = "", sex = "", age = "", phone = "";
        try {
            pstmt = conn.prepareStatement(sql_message);
            resultSet = pstmt.executeQuery();
            resultSet.next();
            no = resultSet.getString(1);
            name = resultSet.getString(2);
            sex = resultSet.getString(3);
            age = resultSet.getString(4);
            phone = resultSet.getString(5);
        } catch (SQLException e) {
            e.printStackTrace();
        }
        JLabel l_message = new JLabel("个人信息:");
        l_message.setFont(new Font("楷体", Font.BOLD, 20));
        JLabel l_no = new JLabel("学 号: " + no);
        JLabel l_name = new JLabel("姓 名: " + name);
        JLabel l_sex = new JLabel("性 别: " + sex);
        JLabel l_age = new JLabel("年 龄: " + age);
        JLabel l_phone = new JLabel("电 话: " + phone);
        JPanel panel_left = new JPanel(new GridLayout(10, 1));
        panel_left.add(l_message);
        panel_left.add(l_no);
        panel_left.add(l_name);
        panel_left.add(l_sex);
    }
}

```

```

        panel_left.add(l_age);
        panel_left.add(l_phone);
        this.add(panel_left);
        JLabel l_score = new JLabel("各科成绩:");
        l_score.setFont(new Font("楷体", Font.BOLD, 20));
        JPanel panel_right = new JPanel(new BorderLayout());
        String sql_score = "select Cname, grade, teacher.name from
ssms.student," +
        " ssms.teacher, ssms.course, ssms.score where
student.S_ID = "
        + input_ID + " and student.S_ID = score.S_ID and "
+
        "score.T_ID = teacher.T_ID and teacher.subject =
course.Cno;";
        String[][] result = ReturnQueryResult.send(sql_score);
        String[] head = {"科目", "分数", "老师"};
        MyTable t_score = new MyTable(result, head);
        JScrollPane sp = new JScrollPane(t_score);
        panel_right.add(l_score, BorderLayout.NORTH);
        panel_right.add(sp, BorderLayout.CENTER);
        this.add(panel_right);
    }
}

class JPanelBottom extends JPanel {

    JPanelBottom() {
        this.setLayout(new FlowLayout(FlowLayout.CENTER, 60, 10));
        init();
    }

    public void init() {
        JButton button_alter = new JButton("修改信息");
        JButton button_password = new JButton("修改密码");
        JButton button_quit = new JButton("退出系统");
        button_alter.setFocusPainted(false);
        button_password.setFocusPainted(false);
        button_quit.setFocusPainted(false);
        button_alter.setPreferredSize(new Dimension(100, 30));
        button_password.setPreferredSize(new Dimension(100, 30));
        button_quit.setPreferredSize(new Dimension(100, 30));
        button_alter.addActionListener(e -> new
AlterDialog(now)); //弹出修改个人信息对话框
        button_password.addActionListener(e -> new

```

```

PasswordDialog(now));
        button_quit.addActionListener(e -> System.exit(0));
        this.add(button_alter);
        this.add(button_password);
        this.add(button_quit);
    }
}

class PasswordDialog extends JDialog {
    PasswordDialog(JFrame parents) {
        super(parents);
        this.setSize(300, 260);
        this.setLocationRelativeTo(parents);
        init();
        this.setVisible(true);
    }

    void compare(String s1, String s2) throws Exception {
        if (!s1.equals(s2))
            throw new Exception("两次输入的密码不同!");
    }

    void init() {
        Container c = this.getContentPane();
        c.setLayout(new GridLayout(5, 1, 0, 15));
        JLabel label_1 = new JLabel("请输入你的新密码:");
        this.add(label_1);
        JPasswordField text_1 = new JPasswordField();
        this.add(text_1);
        JLabel label_2 = new JLabel("请再次输入你的密码:");
        this.add(label_2);
        JPasswordField text_2 = new JPasswordField();
        this.add(text_2);
        JButton bt_save = new JButton("保    存");
        Connection conn = Main.sendConnection();
        String sql = "update ssms.student set student.passcode =?
where student.S_ID=?;";
        bt_save.addActionListener(e -> {
            PreparedStatement pstmt = null;
            try {
                pstmt = conn.prepareStatement(sql);
                try {
                    compare(text_1.getText(), text_2.getText());
                    pstmt.setString(1, text_1.getText());

```

```

        pstmt.setInt(2, Integer.parseInt(input_ID));
        pstmt.executeUpdate();
        JOptionPane.showMessageDialog(this, "修改成功!",
",
                                "Succeed",
JOptionPane.INFORMATION_MESSAGE);
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(this, "两次输入
的密码不同!",
                                "Error", JOptionPane.WARNING_MESSAGE);
    }
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
    });
    this.add(bt_save);
    this.setDefaultCloseOperation(JDialog.HIDE_ON_CLOSE);
}
}

```

```

class AlterDialog extends JDialog {
    AlterDialog(JFrame parents) {
        super(parents);
        this.setSize(350, 460);
        this.setLocationRelativeTo(parents);
        init();
        this.setVisible(true);
    }

    void init() {
        Container c = this.getContentPane();
        c.setLayout(new GridLayout(10, 1, 0, 15));
        JLabel label_age = new JLabel("请输入年龄:");
        this.add(label_age);
        JTextField text_age = new JTextField();
        this.add(text_age);
        JLabel label_phone = new JLabel("请输入电话:");
        this.add(label_phone);
        JTextField text_phone = new JTextField();
        this.add(text_phone);
        JButton bt_save = new JButton("保    存");
        Connection conn = Main.sendConnection();
        String sql = "update ssms.student set student.age=?,
student.phone=? where student.S_ID=?;";
    }
}

```

```

        bt_save.addActionListener(e -> {
            PreparedStatement pstmt = null;
            try {
                pstmt = conn.prepareStatement(sql);
                pstmt.setInt(1,
Integer.parseInt(text_age.getText()));
                pstmt.setString(2, text_phone.getText());
                pstmt.setInt(3, Integer.parseInt(input_ID));
                pstmt.executeUpdate();
                now.init();
                JOptionPane.showMessageDialog(this, "修改成功!",
                    "Succeed",
JOptionPane.INFORMATION_MESSAGE);
            } catch (SQLException | NumberFormatException ex) {
                ex.printStackTrace();
                JOptionPane.showMessageDialog(this, "格式有误, 请
重新输入!",
                    "Error", JOptionPane.WARNING_MESSAGE);
            }
        });
        this.add(bt_save);
        this.setDefaultCloseOperation(JDialog.HIDE_ON_CLOSE);
    }
}
}

```

## 4、运行调试与分析讨论

### (1) 登录

先选择以教师或学生身份登录，输入账号和密码进入相应的客户端。账号分别为老师的职工号和学生的学号。



图 4.1 登录界面

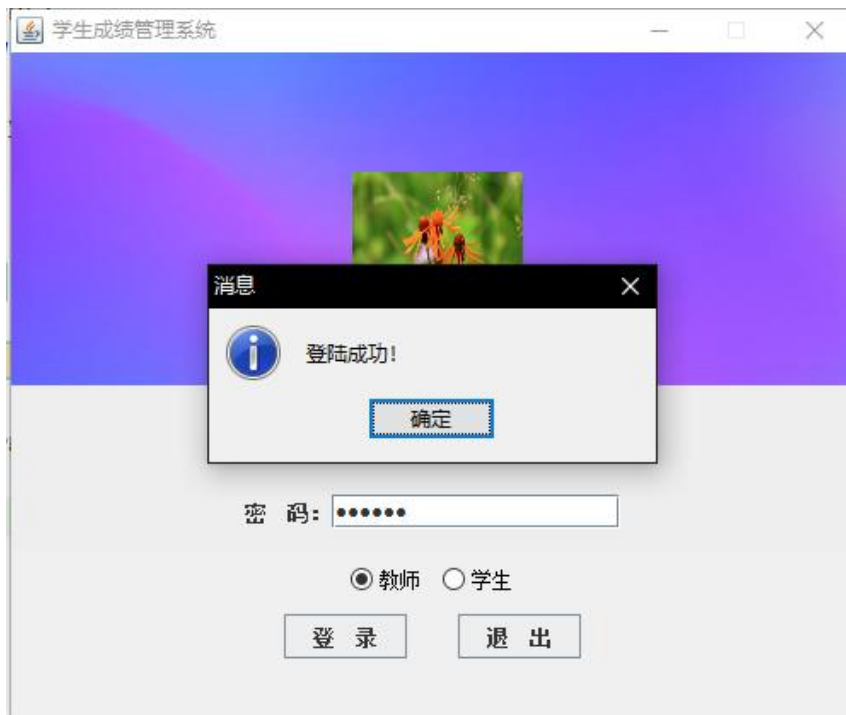


图 4.2 登陆成功界面

## (2) 教师客户端主页面



图 4.3 教师客户端主页面

## (3) 学生客户端主页面



图 4.4 学生客户端主页面

#### (4) 按学号查询

欢迎使用学生成绩管理系统

20001

学号	姓名	性别	年龄	电话
20001	高文君	女	20	12345678901

主页

添加

成绩统计

个人信息

科目	分数	老师
C++	86	徐老师
Java	66	赵老师
数据库原理	77	庞老师
算法与数据结构	99	张老师

图 4.5 按学号查询界面

#### (5) 按姓名查找

欢迎使用学生成绩管理系统

学号	姓名	性别	年龄	电话
20005	张雯	女	23	12345678905

主页

添加

成绩统计

个人信息

科目	分数	老师
C++	85	徐老师
Java	98	赵老师
数据库原理	48	庞老师
算法与数据结构	75	张老师

图 4.6 按姓名查找界面



## (6) 修改成绩



图 4.7 修改成绩界面



图 4.8 修改成功界面

(7) 彻底删除学生信息



图 4.9 删除学生成绩界面



图 4.10 删除信息成功界面

(8) 修改学生信息



图 4.11 修改学生成绩界面

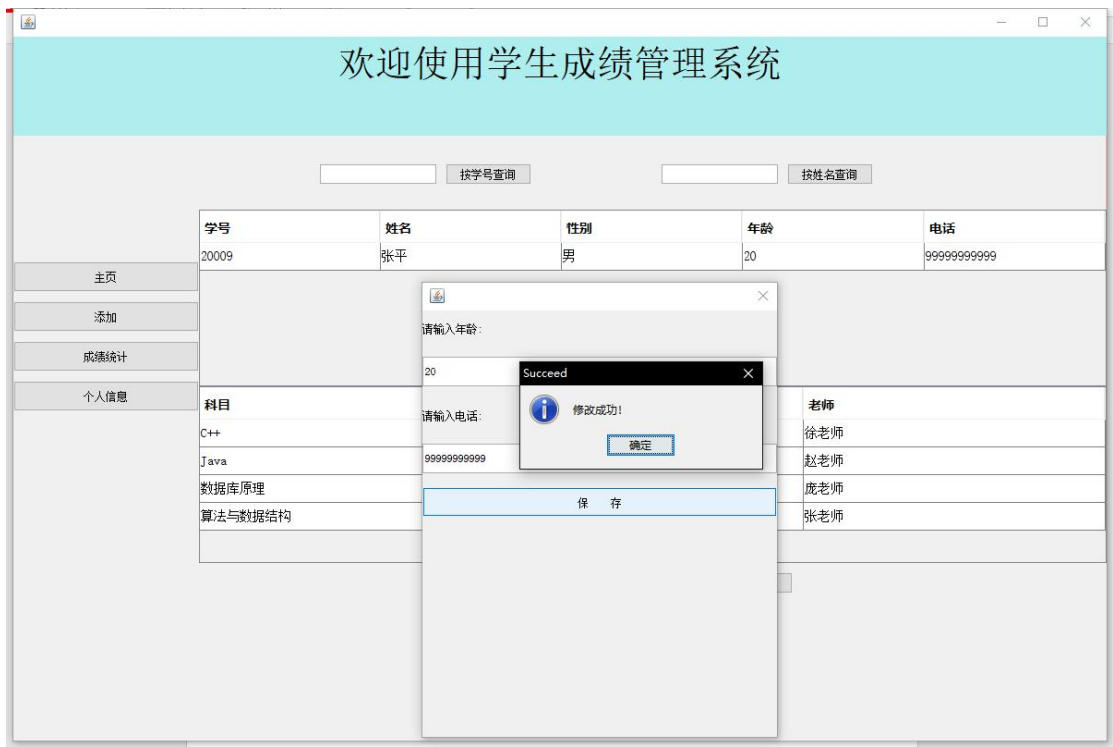


图 4.12 修改成绩成功界面

(9) 添加学生

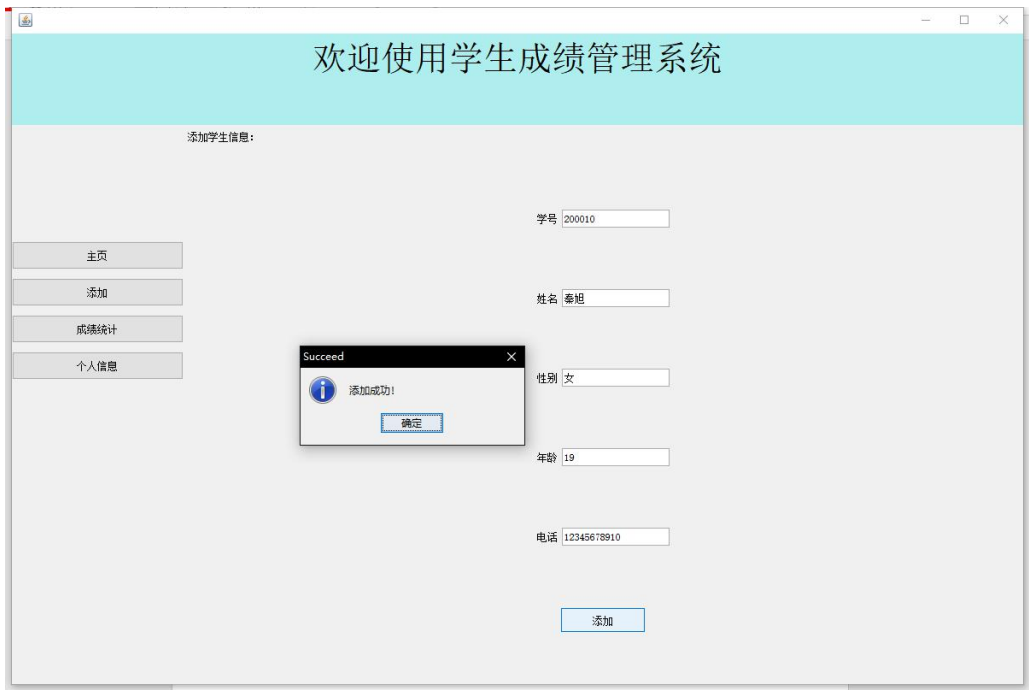


图 4.12 添加学生信息界面

(10) 成绩统计



图 4.13 总成绩统计界面



图 4.14 单科成绩统计界面

## (11) 个人信息



图 4.15 教师个人信息界面

## (12) 学生客户端主页面

The screenshot shows a window titled '学生客户端主页面'. It is divided into two main sections: '个人信息:' (Personal Information) on the left and '各科成绩:' (Exam Scores) on the right. The '个人信息:' section contains fields for '学号:' (Student ID: 20005), '姓名:' (Name: 张雯), '性别:' (Gender: 女), '年龄:' (Age: 23), and '电话:' (Phone: 12345678905). Below these fields are three buttons: '修改信息' (Modify Information), '修改密码' (Modify Password), and '退出系统' (Exit System). The '各科成绩:' section contains a table with three columns: '科目' (Subject), '分数' (Score), and '老师' (Teacher). The table lists four subjects: Java (100, 赵老师), C++ (85, 徐老师), 数据库原理 (48, 庞老师), and 算法与数据结构 (75, 张老师).

科目	分数	老师
Java	100	赵老师
C++	85	徐老师
数据库原理	48	庞老师
算法与数据结构	75	张老师

图 4.16 学生个人信息界面

## (13) 修改信息

The screenshot shows the '修改信息' (Modify Information) dialog box. It contains fields for '请输入年龄:' (Please enter age: 20) and '请输入电话:' (Please enter phone: 12345678966). Below these fields is a '保存' (Save) button. A 'Succeed' dialog box is also visible, displaying '修改成功!' (Modification successful!) and a '确定' (Confirm) button. The background window shows the '个人信息:' section with updated values: '学号:' (20005), '姓名:' (张雯), '性别:' (女), '年龄:' (20), and '电话:' (12345678966). The '各科成绩:' section remains the same as in Figure 4.16.

科目	分数	老师
Java	100	赵老师
C++	85	徐老师
数据库原理	48	庞老师
算法与数据结构	75	张老师

图 4.17 修改学生信息界面

#### (14) 修改密码

**个人信息:**

学 号: 20005

姓 名: 张雯

性 别: 女

年 龄: 20

电 话: 12345678966

**各科成绩:**

科目	分数	老师
Java	100	赵老师
C++	85	徐老师
数据库原理		庞老师
		张老师

请输入你的新密码:

123456

请再次输入你的密码:

123456

保 存

Succeed

修改成功!

确定

修改信息 修改密码 退出系统

图 4.18 学生修改密码界面

## 5、数据结构基本算法实现

### 5.1 利用 SQL 查询语句 **select from** 返回查询结果

```
public static String[][] send(String sql) {
    String[][] result = null;
    Connection conn = null;
    Statement statement = null;
    try {
        conn = Main.sendConnection();
        statement = conn.createStatement();

        ResultSet rs = statement.executeQuery(sql);
        ResultSetMetaData rsmd = rs.getMetaData();
        int column = rsmd.getColumnCount(); // 获取列数
        rs.last();
        int row = rs.getRow(); // 获取行数
        rs.beforeFirst();
        result = new String[row][column];
        int count = 0;
        while (rs.next()) {
            for (int i = 1; i <= column; i++) {
                result[count][i - 1] = rs.getString(i);
            }
            count++;
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return result;
}
```

### 5.2 用 **row\_number() over(order by)** 实现按照分数排名, **sum()** 实现求学生成绩总分, **case when then end** 统计学生各科成绩。

```
void show_all() { // 统计总成绩
    String[] head = {"学号", "姓名", "Java", "数据库原理", "算法与数据结构", "C++", "总分", "平均分", "排名"};
    String sql = "select r.*, row_number() over(order by '平均分') as row_rank\n" +
        "from(\n" +
        "select score.S_ID, student.name, \n" +
```



```

        "sum(case when T_ID = 10001 then score.grade else 0 end) as
' java', \n" +
        "sum(case when T_ID = 10002 then score.grade else 0 end) as
' 数据库', \n" +
        "sum(case when T_ID = 10003 then score.grade else 0 end) as
' 算法', \n" +
        "sum(case when T_ID = 10004 then score.grade else 0 end) as
' C++', \n" +
        "sum(score.grade) as '总分', \n" +
        "format(sum(score.grade)/count(*), 2) as '平均分' \n" +
        "from ssms.score, ssms.student \n" +
        "where score.S_ID = student.S_ID \n" +
        "group by score.S_ID \n" +
        "order by 平均分 desc) r";
String[][] result = ReturnQueryResult.send(sql);
MyTable table = new MyTable(result, head);
JScrollPane jsp = new JScrollPane(table);
panels[0].add(jsp, BorderLayout.CENTER);
String[] head_1 = {"分数段", "优秀(90~100)", "良好(80~90)", "中等
(70~80)", "及格(60~70)", "不及格(0~60)"};
String[][] result_1 = new String[1][6];
result_1[0][0] = "人数";
String sql_1 = "select count(case when ave >= 90 then 1 end) as '
优秀', \n" +
        "count(case when ave < 90 and ave >= 80 then 1 end) as '
良好', \n" +
        "count(case when ave < 80 and ave >= 70 then 1 end) as '
中等', \n" +
        "count(case when ave < 70 and ave >= 60 then 1 end) as '
及格', \n" +
        "count(case when ave < 60 then 1 end) as '不及格' \n" +
        "from(select score.S_ID, student.name, \n" +
        "sum(case when T_ID = 10001 then score.grade else 0 end) as
' java', \n" +
        "sum(case when T_ID = 10002 then score.grade else 0 end) as
' 数据库', \n" +
        "sum(case when T_ID = 10003 then score.grade else 0 end) as
' 算法', \n" +
        "sum(case when T_ID = 10004 then score.grade else 0 end) as
' C++', \n" +
        "sum(score.grade) as '总分', \n" +
        "format(sum(score.grade)/count(*), 2) as ave, \n" +
        "format(@r := @r + 1, 0) as '排名' \n" +
        "from ssms.score, ssms.student, (select @r := 0) alias \n" +

```

```

        "where score.S_ID = student.S_ID\n" +
        "group by score.S_ID) t;";
String[][] resultSet = ReturnQueryResult.send(sql_1);
for(int i = 1; i<6; i++) {
    result_1[0][i] = resultSet[0][i-1];
}
MyTable table_1 = new MyTable(result_1, head_1);
JScrollPane jsp_1 = new JScrollPane(table_1);
jsp_1.setPreferredSize(new Dimension(0, 100));
panels[0].add(jsp_1, BorderLayout.SOUTH);
}
void show_single(int no, String t_no) { //统计每科成绩
    String[] head = {"学号", "姓名", "成绩", "排名"};
    String sql = "select r.*, row_number() over(order by 'subject') as
row_rank\n" +
        "from(\n" +
        "select score.S_ID, student.name,\n" +
        "sum(score.grade) as 'subject'\n" +
        "from ssms.score, ssms.student\n" +
        "where score.S_ID = student.S_ID and score.T_ID = " + t_no
+
        " group by score.S_ID\n" +
        "order by score.grade desc) r";
String[][] result = ReturnQueryResult.send(sql);
MyTable table = new MyTable(result, head);
JScrollPane jsp = new JScrollPane(table);
panels[no].add(jsp, BorderLayout.CENTER);
String[] head_1 = {"分数段", "优秀(90~100)", "良好(80~90)", "中等
(70~80)", "及格(60~70)", "不及格(0~60)"};
String[][] result_1 = new String[1][6];
result_1[0][0] = "人数";
String sql_1 = "select count(case when score.grade >= 90 then 1 end)
as '优秀',\n" +
        "count(case when score.grade < 90 and score.grade >= 80 then
1 end) as '良好',\n" +
        "count(case when score.grade < 80 and score.grade >= 70 then
1 end) as '中等',\n" +
        "count(case when score.grade < 70 and score.grade >= 60 then
1 end) as '及格',\n" +
        "count(case when score.grade < 60 then 1 end) as '不及格'\n"
+
        "from ssms.score\n" +
        "where score.T_ID = " + t_no;
String[][] resultSet = ReturnQueryResult.send(sql_1);

```

```
for(int i = 1; i<6; i++) {  
    result_1[0][i] = resultSet[0][i-1];  
}  
MyTable table_1 = new MyTable(result_1, head_1);  
JScrollPane jsp_1 = new JScrollPane(table_1);  
jsp_1.setPreferredSize(new Dimension(0, 100));  
panels[no].add(jsp_1, BorderLayout.SOUTH);  
}
```

## 6、课程设计总结

经过两周的课设，我所做的课题学生成绩管理系统部分功能基本完成。其功能基本符合的用户要求，能够对学生成绩的基本信息进行查询、修改、添加、删除。通过此课程设计，使我们学到了以前没有学过的很多知识。熟悉了 MYSQL 和 SQL 语句，Swing GUI 工具包以前没用过的组件和 IntelliJ IDEA 开发工具等。编程过程中遇到了很多问题，最后通过求助老师和同学以及上网查询成功解决，我认为编程一定要细心，由于粗心大意导致出现了许多奇怪的问题，浪费了很多时间。课本的知识还是不够的，我应该扩展自己的课外知识，多多阅读课外的相关知识，这样才能对 Java 更加熟悉。最后感谢我的老师和同学们在课程设计过程中给我的帮助。

## 参考文献

- [1]张跃平等. JSP 程序设计[M]. 北京: 清华大学出版社, 2009, 8
- [2]严蔚敏, 吴伟民. 数据结构 (C 语言版). 北京: 清华大学出版社.
- [3]严蔚敏, 吴伟民. 数据结构题集 (C 语言版). 北京: 清华大学出版社.
- [4]郝春梅, 齐景嘉. 数据结构 (C 语言版). 北京: 清华大学出版社, 2010. 6
- [5] William Ford, William Topp .DATA STRUCTURE WITH C++. 北京: 清华大学出版社 (影印版).
- [6]丁振凡. Java 语言实用教程. 北京: 北京邮电大学出版社
- [7]耿祥义. Java2 实用教程. 北京: 清华大学出版社
- [8]赵付青. Java 面向对象程序设计. 北京: 国防工业出版社.
- [9]耿祥义. Java 课程设计. 北京: 清华大学出版社
- [10]陈元春. 实用数据结构基础. 北京: 中国铁道出版社

## 致谢

本次 Java 课设是在我们的指导老师的悉心指导下进行的。老师在 Java 开发有很多的实践经验，在我们面对问题时对我们的悉心指导及其严谨的工作态度，使我们受益匪浅，在此特别向老师表示深深的感谢。在系统完善过程中，我们也遇到了这样或那样的技术问题，但经过自己和同学们的不懈努力及查阅大量的资料，最终都得到了基本满意的答案。同时，其他同学也给了我们许多有益的启示，促动和帮助，使我能够顺利的完成课题。这也表明，我们班是一个团结、互助的集体。谢谢同学们对我的帮助，你们是最可爱的。感谢课设期间每个在机房给予我帮助的老师，给我提供上机的机会，使我们能够和同学在一起上机讨论和学习交流，也增强了我对计算机的兴趣。通过两周的实训，我学到了很多知识。我懂得了如何设计图形界面、如何存储数据等。这一切结果都离不开老师的谆谆教导。老师不但在课上对我们认真负责，把自身的知识传授给我们。这些无不代表着老师对我们的关心和爱护。在做报告期间，老师渊博的学识、严谨求实的科学精神、一丝不苟的治学态度和高尚的品格，深深的感染了我 and 每一个同学。使我认真的做好报告的每一个细节。