

2020-2021 学年第 1 学期

广东外语外贸大学

**《学期综合实践》课程报告**

题 目：设计一个学生成绩管理系统

姓名及学号：

许家彰 20191002867

吕璟源 20191002882

邬坤源 20191003027

具体分工：

吕璟源	具体功能实现，测试优化
邬坤源	UI 设计和界面美化代码实现，编写实践报告
许家彰	可视化设计，包括以表格形式显示所有人成绩和排序

学院：信息科学与技术/网络空间安全学院

年级专业：软件工程 1904

时 间：2020 年 12 月 18 日

目录:

系统介绍..... 3

代码: .....11

[1](#)Student 类 ..... 11

[2](#)开始界面 ..... 12

[3](#)查询界面 ..... 17

[4](#)成绩管理界面 ..... 29

[5](#)打印界面 ..... 37

# 系统介绍

## ● 主界面

首先是开始界面，功能选择窗口，包含 5 个功能按钮



## ● 清除数据功能

清空数据库数据（若出现 bug 可以试试）

## ● 退出程序

点击即退出程序

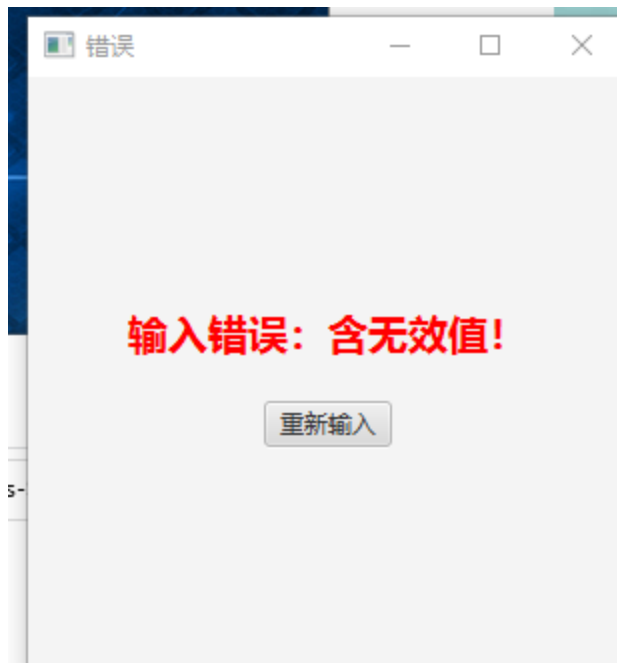
## ● 成绩录入界面



1. 输入学生成绩，点击提交会把信息保存到数据库
2. 返回主页面功能
3. 录入成功提示功能
4. 重复录入检测功能：如果改学生信息已经存在，则弹出提示，询问是否跳转到修改界面



5.空格报错提示功能：若输入无效值（空格）会提示



## ● 查询成绩功能（包含修改）

1. 点击查询进入到修改系统



2. 查询失败（不存在该学生信息）



3.查询成功跳转到修改界面



4.修改成绩功能

(可视化直接操作)，直接修改数据，再点击修改或删除按钮即可实现对数据的管理



5.删除成绩功能



删除和修改

吕璟源	20191002882	78	97	87	68	90	100
许家章	20191002883	87	78	98	89	79	97
吴坤元	20191002884	78	97	99	88	77	66
李倩敏	20191002885	56	65	76	68	99	97
李华	20191002886	78	98	90	23	65	34
小明	20191002888	23	47	68	25	94	85
陈龙	20191002890	23	43	56	74	13	98



姓名	学号	Java程序设计	高等数学	大学英语	操作系统	体育	C++语言	平均分
许家章	20191002883	87	78	98	89	79	97	88.0
吕璟源	20191002882	78	97	87	68	90	100	86.666666666...
吴坤元	20191002884	78	97	99	88	77	66	84.166666666...
李倩敏	20191002885	56	65	76	68	99	97	76.833333333...
李华	20191002886	78	98	90	23	65	34	64.666666666...
小明	20191002888	23	47	68	25	94	85	57.0
陈龙	20191002890	23	43	56	74	13	98	51.166666666...

## ● 程序代码介绍

---

本次学生成绩管理系统采用 Javafx 来编写实现，由郭坤源，许家

彰，吕璟源三人合作开发

代码运行环境为 java jdk-8.0.4\_windows-x64\_bin 或者 jdk-

9.0.4\_windows-x64\_bin

编写工具为 IntelliJ IDEA

具体的功能和模块已经通过注释说明

---

## ● 总结

这次的软工实践选择的题目是设计一个学生成绩管理系统，开发过程中遇到许多难题，加上期末考试将临，没有过多时间去完善和维护代码，代码也有很多不足的地方。我们小组通过讨论和网上查找资料学习解决，增加了学习能力，并且在按时实践内完成了此次实践。通过这次实践，我们也认识到了合作的重要性，分工协作的效率使我们的工作得以快速完成，但是在代码合并上也吸取了很多教训，我们将在后续去学习 git 和 github 的使用，通过 GitHub 平台托管代码建立分支进行开发，最后由组长审核代码并合并到最终版本中。我们也会继续学习前端和后台知识，争取后续在前后端分离的模式下通过使用数据库来体验项目开发模式。

代码:

## Student 类

```
class Student {
    //无参构造方法
    public Student() { }

    /*****
    *****/

    //变量
    String name=null;
    String ID=null;
    String JavaScore=null;
    String MathScore=null;
    String OSScore=null;
    String EngScore=null;
    String SportScore=null;
    String CScore=null;

    /*****
    *****/

    // 提供给外部属性的 set 方法
    public void setCScore(String CScore) {
        this.CScore = CScore;
    }
    public void setMathScore(String mathScore) {
        MathScore = mathScore;
    }
    public void setOSScore(String OSScore) {
        this.OSScore = OSScore;
    }
    public void setEngScore(String engScore) {
        EngScore = engScore;
    }
    public void setSportScore(String sportScore) {
```

```

        SportScore = sportScore;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setID(String ID) {
        this.ID = ID;
    }

    public void setJavaScore(String javaScore) {
        JavaScore = javaScore;
    }
}

/*****
*****/

// 录入文档
void saveToFile(String fileName, Student pupil){
    try{
        BufferedWriter output = new BufferedWriter(new
OutputStreamWriter(new FileOutputStream(fileName, true)));
        output.write(pupil.name+" "+pupil.ID+"
"+pupil.JavaScore+" "+pupil.MathScore+" "
+pupil.EngScore+" "+pupil.OSScore+"
"+pupil.SportScore+" "+pupil.CScore+"\n");
        output.close();
    }catch (IOException e){
        e.printStackTrace();
    }
}
}

```

## 开始界面

```

public class start extends Application {
    @Override
    public void start(Stage primaryStage) throws Exception {
        //主面板
        VBox mainPane = new VBox(20);
        mainPane.setAlignment(Pos.CENTER);
    }
}

```

```

        //最上面的标题
        Label title = new Label("学生成绩管理系统");
        title.setTextFill(Color.RED);
        title.setFont(Font.font("2", FontWeight.BOLD,
FontPosture.ITALIC, 24));

/*****

        //输入
        Button writeScoreButton = new Button("录入成绩");
        //输入按钮引入样式表

writeScoreButton.getStylesheets().add(getClass().getResource("writeScorestyle.css").toExternalForm());
        writeScoreButton.getStyleClass().add("num-button");
        //输入按钮阴影效果
        DropShadow shadow0 = new DropShadow();
        writeScoreButton.setEffect(shadow0);
        //输入功能实现
        writeScoreButton.setOnAction(e->{
            try{
                primaryStage.close();
                new writeScore().start(new Stage());
            } catch (Exception e1){
                e1.printStackTrace();
            }
        }); //跳转到录入成绩界面

/*****

        //查找
        Button searchButton = new Button("查询成绩");
        //查找按钮引入样式表

searchButton.getStylesheets().add(getClass().getResource("writeScorestyle.css").toExternalForm());
        searchButton.getStyleClass().add("num-button");
        //查找按钮阴影效果
        DropShadow shadow1 = new DropShadow();
        searchButton.setEffect(shadow1);
        //查找功能实现

```

```

searchButton.setOnAction(e->{           //跳转到查询成绩界面
    try{
        primaryStage.close();
        new searchScore().start(new Stage());
    } catch (Exception e1){
        e1.printStackTrace();
    }
});

/*****

//打印
Button printScoreButton = new Button("打印成绩");
//打印按钮引入样式表

printScoreButton.getStylesheets().add(getClass().getResource("writeScorestyle.css").toExternalForm());
printScoreButton.setStyleClass().add("num-button");
//打印按钮阴影效果
DropShadow shadow2 = new DropShadow();
printScoreButton.setEffect(shadow2);
printScoreButton.setOnAction(e->{           //跳转到打印成绩界面
    try{
        primaryStage.close();
        new printAllScore().start(new Stage());
    } catch (Exception e1){
        e1.printStackTrace();
    }
});

/*****

//清空数据库数据
Button ClearButton = new Button("清空数据");
//清空按钮引入样式表

ClearButton.getStylesheets().add(getClass().getResource("writeScorestyle.css").toExternalForm());
ClearButton.setStyleClass().add("num-button");
//清空按钮阴影效果
DropShadow shadow4 = new DropShadow();
ClearButton.setEffect(shadow4);

```

```

//清空数据库功能实现
ClearButton.setOnAction(e->{
    File file =new File("data.txt");
    try {
        if(!file.exists()) {
            file.createNewFile();
        }
        FileWriter fileWriter =new FileWriter(file);
        fileWriter.write("");
        fileWriter.flush();
        fileWriter.close();

        //弹出提示框 显示清除成功, 提供 ok 按钮, 按下返回
        {
            Stage stage = new Stage();
            VBox okPane = new VBox(20);
            okPane.setAlignment(Pos.CENTER);
            //最上面的标题
            Label okTitle = new Label("清除成功!");
            okTitle.setTextFill(Color.RED);
            okTitle.setFont(Font.font("2", FontWeight.BOLD,
FontPosture.ITALIC, 20));
            // ok 按钮
            Button writeAgainButton = new Button("OK");
            writeAgainButton.setOnAction(e2->{
                try{
                    stage.close();
                } catch (Exception e3){
                    e3.printStackTrace();
                }
            });
            //舞台布置

            okPane.getChildren().addAll(okTitle,writeAgainButton);
            Scene s = new Scene(okPane,200,100);
            stage.setScene(s);
            stage.show();
        }

    } catch (IOException e6) {
        e6.printStackTrace();
    }
});

```

```

/*****
*****/

//退出
Button exitButton = new Button("退出程序");
//退出按钮引入样式表

exitButton.getStylesheets().add(getClass().getResource("writeScorestyle.css").toExternalForm());
exitButton.getStyleClass().add("num-button");
//退出按钮阴影效果
DropShadow shadow3 = new DropShadow();
exitButton.setEffect(shadow3);
//退出功能实现
exitButton.setOnAction(e->{ //跳转到打印成绩界面
    primaryStage.close();
});

/*****
*****/

//所有部件加入到主面板

mainPane.getChildren().addAll(title,writeScoreButton,searchButton,printScoreButton,ClearButton,exitButton);
mainPane.setId("mainPane");
//搭建舞台
Scene scene = new Scene(mainPane, 400, 500);

scene.getStylesheets().add(getClass().getResource("writeScorestyle.css").toExternalForm());
primaryStage.setTitle("功能选择窗口");
primaryStage.setScene(scene);
primaryStage.show();
}

public static void main(String[] args) {
    launch(args);
}
}

```



# 查询界面

```
public class searchScore extends Application {
    @Override
    public void start(Stage primaryStage) throws Exception {
        //主面板
        VBox searchPane = new VBox(20);
        searchPane.setAlignment(Pos.CENTER);

        //最上面的标题
        Label title = new Label("学生成绩查询系统");
        title.setTextFill(Color.RED);
        title.setFont(Font.font("2", FontWeight.BOLD,
FontPosture.ITALIC, 24));

        //姓名
        HBox namePane = new HBox(20);
        namePane.setAlignment(Pos.CENTER);
        Label name = new Label("姓名");
        TextField nameInput = new TextField();
        namePane.getChildren().addAll(name, nameInput);

        //学号
        HBox IDPane = new HBox(20);
        IDPane.setAlignment(Pos.CENTER);
        Label ID = new Label("学号");
        TextField IDInput = new TextField();
        IDPane.getChildren().addAll(ID, IDInput);

        /*****
        *****/

        //查询按钮
        Button searchButton = new Button("查询");
        //查找按钮引入样式表

        searchButton.getStylesheets().add(getClass().getResource("writeScores
tyle.css").toExternalForm());
        searchButton.getStyleClass().add("num-button");
        //查找按钮阴影效果
        DropShadow shadow1 = new DropShadow();
```

```

searchButton.setEffect (shadow1);
//查询功能实现
searchButton.setOnAction(e->{
    try{
        // 把姓名和学号传给全局变量,所有人都可以使用
        globalUse.name = nameInput.getText().trim();
        globalUse.ID = IDInput.getText().trim();

        //查找信息
        BufferedReader input = new BufferedReader(new
InputStreamReader(new FileInputStream("data.txt")));
        String str;
        boolean CanFind=false; // 用来标记是否
找到该数据项

        // 扫描文档,看是不是能找到该学生信息
        while((str = input.readLine())!= null) // 读取一行
        {
            String[] fields = str.split(" "); //用空格分离
            //这里出bug了 fields[0]
            if( fields[0].equals(globalUse.name) &&
fields[1].equals(globalUse.ID) )
            {
                CanFind = true; //data.txt 中存在该数据
                // 把其他成绩信息传递给全局变量
                globalUse.Java = fields[2] ;
                globalUse.Math = fields[3] ;
                globalUse.English = fields[4] ;
                globalUse.OS = fields[5] ;
                globalUse.Sport = fields[6];
                globalUse.C = fields[7];
            }
        }
        //找得到
        if(CanFind){ //找得到,跳转到修改界面 modifyData
            primaryStage.close();
            new modifyData().start(new Stage());
        }
        // 找不到,data.txt 中没有该数据项,弹出错误框
        else{
            Stage stage = new Stage();
            VBox erroPane = new VBox(20);
            erroPane.setAlignment(Pos.CENTER);
            //最上面的标题

```

```

        Label erroTitle = new Label("查无此人");
        erroTitle.setTextFill(Color.RED);
        erroTitle.setFont(Font.font("2", FontWeight.BOLD,
FontPosture.ITALIC, 20));
        // 重新输入按钮
        Button searchAgainButton = new Button("重新输入");
        searchAgainButton.setOnAction(e2->{
            try{
                stage.close();
                // 清空姓名和学号输入文本框
                nameInput.setText("");
                IDInput.setText("");
            } catch (Exception e3){
                e3.printStackTrace();
            }
        });
        //舞台布置

erroPane.getChildren().addAll(erroTitle,searchAgainButton);
        Scene s = new Scene(erroPane,200,100);
        stage.setScene(s);
        stage.show();
    }

    } catch (Exception e1){
        e1.printStackTrace();
    }
});

/*****
*****/

//返回主页面
Button returnButton = new Button("返回主页面");
//返回主界面按钮引入样式表

returnButton.getStylesheets().add(getClass().getResource("writeScore
style.css").toExternalForm());
        returnButton.getStyleClass().add("num-button");
        //返回主界面按钮阴影效果
        DropShadow shadow3 = new DropShadow();
        returnButton.setEffect(shadow3);
        //返回功能实现

```

```

returnButton.setOnAction(e->{
    try{
        primaryStage.close();
        new start().start(new Stage());
    } catch (Exception e1){
        e1.printStackTrace();
    }
});

/*****
*****/

HBox BtnPane = new HBox(20);
BtnPane.setAlignment(Pos.CENTER);
BtnPane.getChildren().addAll(searchButton, returnButton);

//加入到主面板

searchPane.getChildren().addAll(title, namePane, IDPane, BtnPane);
searchPane.setId("mainPane");
//搭建舞台
Scene scene = new Scene(searchPane, 400, 300);

scene.getStylesheets().add(getClass().getResource("writeScorestyle.css").toExternalForm());
primaryStage.setTitle("查询窗口");
primaryStage.setScene(scene);
primaryStage.show();
}

public static void main(String[] args) {
    launch(args);
}

}

```

成绩录入界面 `import com.sun.org.apache.bcel.internal.generic.SIPUSH;`  
`import javafx.application.Application;`

```

import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.effect.DropShadow;
import javafx.scene.layout.HBox;
import javafx.scene.layout.VBox;
import javafx.scene.paint.Color;
import javafx.scene.text.Font;
import javafx.scene.text.FontPosture;
import javafx.scene.text.FontWeight;
import javafx.stage.Stage;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.InputStreamReader;

public class writeScore extends Application {
    @Override
    public void start(Stage primaryStage) throws Exception {
        //主面板
        VBox mainPane = new VBox(20);
        mainPane.setAlignment(Pos.CENTER);

        //最上面的标题
        Label title = new Label("学生成绩录入");
        title.setTextFill(Color.RED);
        title.setFont(Font.font("2", FontWeight.BOLD,
FontPosture.ITALIC, 24));

        //姓名
        HBox namePane = new HBox(20);
        namePane.setAlignment(Pos.CENTER);
        Label name = new Label("学生姓名");
        TextField nameInput = new TextField();
        namePane.getChildren().addAll(name, nameInput);

        //学号
        HBox IDPane = new HBox(20);
        IDPane.setAlignment(Pos.CENTER);
        Label ID = new Label("学生学号");
        TextField IDInput = new TextField();
        IDPane.getChildren().addAll(ID, IDInput);

        //Java 成绩
        HBox JavaScorePane = new HBox(20);

```

```

        JavaScorePane.setAlignment(Pos.CENTER);
        Label JavaScore = new Label("Java 语言");
        TextField JavaScoreInput = new TextField();
        JavaScorePane.getChildren().addAll(JavaScore, JavaScoreInput);
        //高数成绩
        HBox mathematicsPane = new HBox(20);
        mathematicsPane.setAlignment(Pos.CENTER);
        Label mathematics = new Label("高等数学");
        TextField mathematicsInput = new TextField();

        mathematicsPane.getChildren().addAll(mathematics, mathematicsInput);
        //大学英语
        HBox EnglishPane = new HBox(20);
        EnglishPane.setAlignment(Pos.CENTER);
        Label English = new Label("大学英语");
        TextField EnglishInput = new TextField();
        EnglishPane.getChildren().addAll(English, EnglishInput);
        //操作系统
        HBox OSPane = new HBox(20);
        OSPane.setAlignment(Pos.CENTER);
        Label OS = new Label("操作系统");
        TextField OSInput = new TextField();
        OSPane.getChildren().addAll(OS, OSInput);
        //体育
        HBox SportsPane = new HBox(20);
        SportsPane.setAlignment(Pos.CENTER);
        Label Sports = new Label("大学体育");
        TextField SportsInput = new TextField();
        SportsPane.getChildren().addAll(Sports, SportsInput);
        //C 语言
        HBox CPane = new HBox(20);
        CPane.setAlignment(Pos.CENTER);
        Label C = new Label("C++语言");
        TextField CInput = new TextField();
        CPane.getChildren().addAll(C, CInput);

        /*****
        *****/

        //提交
        Button submitButton = new Button("提交");
        //提交按钮引入样式表

        submitButton.getStylesheets().add(getClass().getResource("writeScores

```

```

tyle.css").toExternalForm());
    submitButton.setId("submitButton");
    //提交按钮阴影效果
    DropShadow shadow = new DropShadow();
    submitButton.setEffect(shadow);
    //功能实现: 实例化一个 Student 对象, 从文本框中获取值, 值传给函数
    saveToFile, 录入信息到文档中
    submitButton.setOnAction(e->{    //成绩录入
        boolean ifError = false ;
        //检查是否有空
        {

            if(nameInput.getText().trim().equals("") )    ifError
= true ;
            if(IDInput.getText().trim().equals("") )    ifError =
true ;
            if(JavaScoreInput.getText().trim().equals("") )
ifError = true ;
            if(mathematicsInput.getText().trim().equals("") )
ifError = true ;
            if(EnglishInput.getText().trim().equals("") )
ifError = true ;
            if(OSInput.getText().trim().equals("") )    ifError =
true ;
            if(SportsInput.getText().trim().equals("") )
ifError = true ;
            if(CInput.getText().trim().equals("") )    ifError =
true;

            if(ifError)
            {
                primaryStage.close();
                Stage stage = new Stage();
                VBox okPane = new VBox(20);
                okPane.setAlignment(Pos.CENTER);
                //最上面的标题
                Label okTitle = new Label("输入错误: 含无效值!");
                okTitle.setTextFill(Color.RED);
                okTitle.setFont(Font.font("2", FontWeight.BOLD,
FontPosture.ITALIC, 20));
                // 重新输入按钮
                Button writeAgainButton = new Button("重新输入");
                writeAgainButton.setOnAction(e2->{
                    try{

```

```

        stage.close();
        new writeScore().start(new Stage());
    } catch (Exception e3) {
        e3.printStackTrace();
    }
});
//舞台布置

okPane.getChildren().addAll(okTitle, writeAgainButton);
Scene s = new Scene(okPane, 300, 300);
stage.setScene(s);
stage.setTitle("错误");
stage.show();
}

}

if(!ifError) {
    System.out.println(1);
    Student pupil = new Student();
    pupil.setName(nameInput.getText().trim());
    pupil.setID(IDInput.getText().trim());

pupil.setJavaScore(JavaScoreInput.getText().trim());

pupil.setMathScore(mathematicsInput.getText().trim());
pupil.setEngScore(EnglishInput.getText().trim());
pupil.setOSScore(OSInput.getText().trim());
pupil.setSportScore(SportsInput.getText().trim());
pupil.setCScore(CInput.getText().trim());

//检查该学号和姓名是否已存在于数据库中，若存在，提示并跳转
到修改界面

    try {
        BufferedReader input = new BufferedReader(new
InputStreamReader(new FileInputStream("data.txt")));
        String str;
        boolean CanFind = false; //
用来标记是否找到该数据项
        while ((str = input.readLine()) != null) // 读取
一行
        {
            String[] fields = str.split(" "); //用空格
分离
            //这里出bug了 fields 的长度必须到达8，也就是说

```



*fields[7]一定要有值*

```
        if (fields[0].equals(pupil.name) &&
fields[1].equals(pupil.ID)) {
            CanFind = true; //data.txt 中存在该数据

            //把这个学生的数据传给全局变量，以便在修改界面中显
```

示

```
        {
            globalUse.name = pupil.name;
            globalUse.ID = pupil.ID;
            globalUse.Java = fields[2];
            globalUse.Math = fields[3];
            globalUse.English = fields[4];
            globalUse.OS = fields[5];
            globalUse.Sport = fields[6];
            globalUse.C = fields[7];
        }
    }
}
```

*//找得到*

```
if (CanFind) { //找得到，跳转到修改界面 modifyData
    //弹出提示框 显示录入成功，提供 ok 按钮，按下返回
```

并清除文本框内容

```
{

    Stage stage = new Stage();
    VBox okPane = new VBox(20);
    okPane.setAlignment(Pos.CENTER);
    //最上面的标题
    Label okTitle = new Label("该学生成绩已存在！

");

    okTitle.setTextFill(Color.RED);
    okTitle.setFont(Font.font("2",
FontWeight.BOLD, FontPosture.ITALIC, 20));
    // 重新输入按钮
    Button writeAgainButton = new Button("重新输

入");

    writeAgainButton.setOnAction(e2 -> {
        try {
            stage.close();
            nameInput.setText("");
            IDInput.setText("");
            JavaScoreInput.setText("");
        }
    });
}
```

```

        mathematicsInput.setText("");
        OSInput.setText("");
        EnglishInput.setText("");
        SportsInput.setText("");
        CInput.setText("");
    } catch (Exception e3) {
        e3.printStackTrace();
    }
});
//跳转到修改界面
Button JumpToModifyButton = new Button("修
改该学生信息");

JumpToModifyButton.setOnAction(e2 -> {
    try {
        stage.close();
        primaryStage.close();
        new modifyData().start(new
Stage());
    } catch (Exception e3) {
        e3.printStackTrace();
    }
});
//舞台布置
okPane.getChildren().addAll(okTitle,
writeAgainButton, JumpToModifyButton);

Scene s = new Scene(okPane, 300, 300);
stage.setScene(s);
stage.setTitle("错误");
stage.show();
}
}

// 找不到,data.txt 中没有该数据项, 正常录入
else {
    pupil.saveToFile("data.txt", pupil);
    //弹出提示框 显示录入成功, 提供 ok 按钮, 按下返回
并清除文本框内容
{

    Stage stage = new Stage();
    VBox okPane = new VBox(20);
    okPane.setAlignment(Pos.CENTER);
    //最上面的标题

```

```

        Label okTitle = new Label("录入成功!");
        okTitle.setTextFill(Color.RED);
        okTitle.setFont(Font.font("2",
FontWeight.BOLD, FontPosture.ITALIC, 20));
        // 重新输入按钮
        Button writeAgainButton = new Button("OK");
        writeAgainButton.setOnAction(e2 -> {
            try {
                stage.close();
                nameInput.setText("");
                IDInput.setText("");
                JavaScoreInput.setText("");
                mathematicsInput.setText("");
                OSInput.setText("");
                EnglishInput.setText("");
                SportsInput.setText("");
                CInput.setText("");
            } catch (Exception e3) {
                e3.printStackTrace();
            }
        });
        //舞台布置
        okPane.getChildren().addAll(okTitle,
writeAgainButton);

        Scene s = new Scene(okPane, 200, 100);
        stage.setScene(s);
        stage.show();

    }

} catch (Exception e1) {
    e1.printStackTrace();
}

} //如果没错误则执行 if 块的代码
});

/*****
*****/

//返回
Button returnButton = new Button("返回主页面");

```

```
returnButton.getStylesheets().add(getClass().getResource("writeScores
tyle.css").toExternalForm());

returnButton.getStyleClass().add("num-button");
//返回主界面按钮阴影效果
DropShadow shadow3 = new DropShadow();
returnButton.setEffect(shadow3);
//功能实现：返回主页面
returnButton.setOnAction(e->{
    try{
        primaryStage.close();
        new start().start(new Stage());
    } catch (Exception e1){
        e1.printStackTrace();
    }
});

/*****
*****/

HBox BtnPane = new HBox(20);
BtnPane.setAlignment(Pos.CENTER);
BtnPane.getChildren().addAll(submitButton,returnButton);

//加入到主面板

mainPane.getChildren().addAll(title,namePane,IDPane,JavaScorePane,mat
hematicsPane,EnglishPane,OSPane,SportsPane,CPane,BtnPane);
mainPane.setId("mainPane");
//搭建舞台
Scene scene = new Scene(mainPane, 500, 500);

scene.getStylesheets().add(getClass().getResource("writeScorestyle.cs
s").toExternalForm());

primaryStage.setTitle("录入成绩窗口");
primaryStage.setScene(scene);
primaryStage.show();
}

public static void main(String[] args) {
    launch(args);
}

}
```

# 成绩管理界面

```
public class modifyData extends Application {  
    @Override  
    public void start(Stage primaryStage) throws Exception {  
        //布局设计: 主面板用 VBox, 上面用 HBox, 每一个 HBox 面板包含两个数据项  
        VBox modifyPane = new VBox(20);  
        modifyPane.setAlignment(Pos.CENTER);  
  
        //最上面的标题  
        Label title = new Label("学生成绩修改系统");  
        title.setTextFill(Color.RED);  
        title.setFont(Font.font("2", FontWeight.BOLD,  
FontPosture.ITALIC, 24));  
  
        //第一行: 名字和学号  
        HBox hBox1 = new HBox(20);  
        hBox1.setAlignment(Pos.CENTER);  
        Label name = new Label("学生姓名");  
        TextField nameInput = new TextField(globalUse.name);  
        Label ID = new Label("学生学号");  
        TextField IDInput = new TextField(globalUse.ID);  
        hBox1.getChildren().addAll(name, nameInput, ID, IDInput);  
  
        //第二行: Java 和 体育  
        HBox hBox2 = new HBox(20);  
        hBox2.setAlignment(Pos.CENTER);  
        Label JavaScore = new Label("Java 成绩");  
        TextField JavaScoreInput = new TextField(globalUse.Java);  
        Label SportScore = new Label("大学体育");  
        TextField SportScoreInput = new TextField(globalUse.Sport);  
  
        hBox2.getChildren().addAll(JavaScore, JavaScoreInput, SportScore, Sports  
coreInput);  
  
        //第三行: 操作系统 和 高等数学  
        HBox hBox3 = new HBox(20);  
        hBox3.setAlignment(Pos.CENTER);  
        Label OSScore = new Label("操作系统");  
        TextField OSScoreInput = new TextField(globalUse.OS);  
        Label mathScore = new Label("高等数学");
```

```

        TextField mathScoreInput = new TextField(globalUse.Math);

hBox3.getChildren().addAll(OSScore,OSScoreInput,mathScore,mathScoreInput);

        //第四行： 英语 和 C 语言
HBox hBox4 = new HBox(20);
        hBox4.setAlignment(Pos.CENTER);
        Label EnglishScore = new Label("英语成绩");
        TextField EnglishScoreInput = new
TextField(globalUse.English);
        Label CScore = new Label("C++语言");
        TextField CScoreInput = new TextField(globalUse.C);

hBox4.getChildren().addAll(EnglishScore,EnglishScoreInput,CScore,CScoreInput);

/*****
*****/

        //第五行
HBox hBox5 = new HBox(20);
        hBox5.setAlignment(Pos.CENTER);
        //修改成绩按钮
Button modifyButton = new Button("修改成绩");
        //修改成绩按钮引入样式表

modifyButton.getStylesheets().add(getClass().getResource("writeScorestyle.css").toExternalForm());
        modifyButton.getStyleClass().add("num-button");
        //修改成绩按钮阴影效果
DropShadow shadow5 = new DropShadow();
        modifyButton.setEffect(shadow5);
        //修改功能实现：新建一个temp 中转文件，将修改后的信息写入temp，再把temp 复制给 data.txt
        modifyButton.setOnAction(e->{
            try{
                //从 data.txt 读取信息
                BufferedReader input = new BufferedReader(new
InputStreamReader(new FileInputStream("data.txt")));
                //写入到中转文件
                BufferedWriter outputToTemp = new BufferedWriter(new
OutputStreamWriter(new FileOutputStream("temp.txt")));

```

```

        String str;
        //扫描文件, 找到目标数据项的位置
        // 循环读取一行
        while((str = input.readLine()) != null)
        {
            String[] fields = str.split(" "); //用空格分离, 存入
数组
            // 找到需要修改的数据项
            if( fields[0].equals(globalUse.name) &&
fields[1].equals(globalUse.ID) )
                { //找到该数据项在文件中的位置, 修改应该改的信息, 再存入到中
转文件
                    outputToTemp.write(nameInput.getText()+"
"+IDInput.getText()+" "+JavaScoreInput.getText()+"
"+mathScoreInput.getText()+" "+EnglishScoreInput.getText()+"
"+OSScoreInput.getText()+" "+SportScoreInput.getText()+"
"+CScoreInput.getText()+"\n");
                }
                // 其他信息原封不动存入中转文件
            else {
                outputToTemp.write(str+"\n");
            }
        }
        //关闭文件流
        input.close();
        outputToTemp.close();

        //从中转文件 temp.txt 读取信息
        BufferedReader inputFromTemp = new BufferedReader(new
InputStreamReader(new FileInputStream("temp.txt")));
        //新建一个 data.txt 写入信息
        BufferedWriter outputToData = new BufferedWriter(new
OutputStreamWriter(new FileOutputStream(new File("data.txt"))));

        while((str = inputFromTemp.readLine()) != null)
        {
            outputToData.write(str+"\n");
        }
        //关闭文件流
        inputFromTemp.close();
        outputToData.close();

        //弹出提示框 显示修改成功, 提供 ok 按钮, 按下返回
        {

```

```

        Stage stage = new Stage();
        VBox okPane = new VBox(20);
        okPane.setAlignment(Pos.CENTER);
        //最上面的标题
        Label okTitle = new Label("修改成功!");
        okTitle.setTextFill(Color.RED);
        okTitle.setFont(Font.font("2", FontWeight.BOLD,
FontPosture.ITALIC, 20));
        // ok 按钮
        Button modifyAgainButton = new Button("OK");
        modifyAgainButton.setOnAction(e2->{
            try{
                stage.close();
            } catch (Exception e3){
                e3.printStackTrace();
            }
        });
        //舞台布置

        okPane.getChildren().addAll(okTitle,modifyAgainButton);
        Scene s = new Scene(okPane,200,100);
        stage.setScene(s);
        stage.show();
        //2000millis 后自动关闭
        Thread thread = new Thread(() -> {
            try {
                Thread.sleep(2000);
                if (stage.isShowing()) {
                    Platform.runLater(() ->
stage.close());
                }
            } catch (Exception exp) {
                exp.printStackTrace();
            }
        });
        thread.setDaemon(true);
        thread.start();

    }

} catch (IOException a){
    a.printStackTrace();
}

});

```



```

/*****
*****/

//删除数据项按钮
Button deleteButton = new Button("删除数据项");
//删除数据项按钮引入样式表

deleteButton.getStylesheets().add(getClass().getResource("writeScorestyle.css").toExternalForm());
deleteButton.getStyleClass().add("num-button");
//删除数据项按钮阴影效果
DropShadow shadow6 = new DropShadow();
deleteButton.setEffect(shadow6);
//删除功能实现：新建一个 temp 中转文件，将修改后的信息写入 temp，再把
temp 复制给 data.txt
deleteButton.setOnAction(e->{
    try{
        //从 data.txt 读取信息
        BufferedReader input = new BufferedReader(new
InputStreamReader(new FileInputStream("data.txt")));
        //写入到中转文件
        BufferedWriter outputToTemp = new BufferedWriter(new
OutputStreamWriter(new FileOutputStream("temp.txt")));

        String str;
        //扫描文件，找到目标数据项的位置
        // 循环读取一行
        while((str = input.readLine()) != null)
        {
            String[] fields = str.split(" "); //用空格分离,存入
数组

            // 找到需要修改的数据项
            if( fields[0].equals(globalUse.name) &&
fields[1].equals(globalUse.ID) )
            {
                continue;
            }
            // 其他信息原封不动存入中转文件
            else {
                outputToTemp.write(str+"\n");
            }
        }
    }
}

```

```

        //关闭文件流
input.close();
outputToTemp.close();

        //从中转文件 temp.txt 读取信息
BufferedReader inputFromTemp = new BufferedReader(new
InputStreamReader(new FileInputStream("temp.txt")));
        //新建一个 data.txt 写入信息
BufferedWriter outputToData = new BufferedWriter(new
OutputStreamWriter(new FileOutputStream(new File("data.txt"))));
        //文件复制
while((str = inputFromTemp.readLine()) != null)
{
    outputToData.write(str+"\n");
}
        //关闭文件流
inputFromTemp.close();
outputToData.close();

        //弹出提示框 显示删除成功, 提供 ok 按钮, 按下返回
{
    Stage stage = new Stage();
    VBox okPane = new VBox(20);
    okPane.setAlignment(Pos.CENTER);
    //最上面的标题
    Label okTitle = new Label("删除成功!");
    okTitle.setTextFill(Color.RED);
    okTitle.setFont(Font.font("2", FontWeight.BOLD,
FontPosture.ITALIC, 20));
    // ok 按钮
    Button modifyAgainButton = new Button("OK");
    modifyAgainButton.setOnAction(e2->{
        try{
            stage.close();
            primaryStage.close();
            new searchScore().start(new Stage());

        } catch (Exception e3){
            e3.printStackTrace();
        }
    });
    //提示框舞台布置

```

```

okPane.getChildren().addAll(okTitle, modifyAgainButton);
    Scene s = new Scene(okPane, 200, 100);
    stage.setScene(s);
    stage.show();
    //2000millis 后自动关闭提示框
    Thread thread = new Thread(() -> {
        try {
            Thread.sleep(2000);
            if (stage.isShowing()) {
                Platform.runLater(() ->
stage.close());
            }
        } catch (Exception exp) {
            exp.printStackTrace();
        }
    });
    thread.setDaemon(true);
    thread.start();
}

} catch (IOException a) {
    a.printStackTrace();
}
});

hBox5.getChildren().addAll(modifyButton, deleteButton);

/*****
*****/

//第6行： 返回主查询页面按钮
Button backButton = new Button("返回查询页面");
//返回主查询页面按钮引入样式表

BackButton.getStylesheets().add(getClass().getResource("writeScorestyle.css").toExternalForm());
BackButton.getStyleClass().add("num-button");
//返回主查询页面按钮阴影效果
DropShadow shadow3 = new DropShadow();
BackButton.setEffect(shadow3);
//返回按钮功能实现
BackButton.setOnAction(e -> {
    try {

```

```

        primaryStage.close();
        new searchScore().start(new Stage());
    } catch (Exception e1) {
        e1.printStackTrace();
    }
});

/*****
*****/

//第7行： 返回主页面按钮
Button returnButton = new Button("返回主页面");
//返回主界面按钮引入样式表

returnButton.getStylesheets().add(getClass().getResource("writeScore
style.css").toExternalForm());
returnButton.getStyleClass().add("num-button");
//返回主界面按钮阴影效果
DropShadow shadow4 = new DropShadow();
returnButton.setEffect(shadow4);
//返回主页面按钮功能实现
returnButton.setOnAction(e->{
    try{
        primaryStage.close();
        new start().start(new Stage());
    } catch (Exception e1) {
        e1.printStackTrace();
    }
});

//加入到主面板

modifyPane.getChildren().addAll(title,hBox1,hBox2,hBox3,hBox4,hBox5,B
ackButton,returnButton);
modifyPane.setId("mainPane");
//搭建舞台
Scene scene = new Scene(modifyPane, 600, 500);

```

```

scene.getStylesheets().add(getClass().getResource("writeScorestyle.css").toExternalForm());

primaryStage.setTitle("成绩管理窗口");
primaryStage.setScene(scene);
primaryStage.show();
}

public static void main(String[] args) {
    launch(args);
}
}

```

## 打印界面

```

public class printAllScore extends Application {

    TableView<Student> table = new TableView<>(); //表控件是通过实例化
    TableView 类创建的。

    public static void main(String[] args) {
        launch(args);
    }

    @Override
    public void start(Stage primaryStage) throws Exception {

        Scene scene = new Scene(new Group());
        primaryStage.setWidth(930);
        primaryStage.setHeight(500);
        primaryStage.setTitle("成绩表");
        //最上面的标题
        Label title = new Label("学生成绩单");
        title.setTextFill(Color.RED);
        title.setFont(Font.font("2", FontWeight.BOLD,
        FontPosture.ITALIC, 24));

        // 加载表格
        TableColumn name = new TableColumn("姓名");
        name.setMinWidth(100);
        name.setCellValueFactory(new PropertyValueFactory<>("name"));
    }
}

```

```

TableColumn ID = new TableColumn("学号");
ID.setMinWidth(100);
ID.setCellValueFactory(new PropertyValueFactory<>("id"));

TableColumn Gmath = new TableColumn("高等数学");
Gmath.setMinWidth(100);
Gmath.setCellValueFactory(new
PropertyValueFactory<>("gaomath"));

TableColumn eng = new TableColumn("大学英语");
eng.setMinWidth(100);
eng.setCellValueFactory(new PropertyValueFactory<>("Eng"));

TableColumn JavaScore = new TableColumn("Java 程序设计");
JavaScore.setMinWidth(100);
JavaScore.setCellValueFactory(new
PropertyValueFactory<>("JavaScores"));

TableColumn os = new TableColumn("操作系统");
os.setMinWidth(100);
os.setCellValueFactory(new PropertyValueFactory<>("OS"));

TableColumn AVERAGE = new TableColumn("平均分");
AVERAGE.setMinWidth(100);
AVERAGE.setCellValueFactory(new
PropertyValueFactory<>("average"));

TableColumn RANK = new TableColumn("排名");
RANK.setMinWidth(100);
RANK.setCellValueFactory(new PropertyValueFactory<>("rank"));

TableColumn Sport = new TableColumn("体育");
Sport.setMinWidth(100);
Sport.setCellValueFactory(new
PropertyValueFactory<>("sport"));

TableColumn CPP = new TableColumn("C++语言");
CPP.setMinWidth(100);
CPP.setCellValueFactory(new PropertyValueFactory<>("cpp"));

getProductsFromFile();

```

```
table.getColumns().addAll(name, ID, JavaScore, Gmath, eng, os, Sport, CPP, AVERAGE);
```

```
//提示文本
```

```
Label tips = new Label("点击最上面成绩标签可排序");
```

```
//返回按钮
```

```
Button returnButton = new Button("返回主页面");
```

```
//返回主界面按钮引入样式表
```

```
returnButton.getStylesheets().add(getClass().getResource("writeScorestyle.css").toExternalForm());
```

```
returnButton.getStyleClass().add("num-button");
```

```
//返回主界面按钮阴影效果
```

```
DropShadow shadow3 = new DropShadow();
```

```
returnButton.setEffect(shadow3);
```

```
//返回按钮功能实现
```

```
returnButton.setOnAction(e -> {  
    try {  
        primaryStage.close();  
        new start().start(new Stage());  
    } catch (Exception e1) {  
        e1.printStackTrace();  
    }  
});
```

```
final VBox vbox = new VBox();
```

```
vbox.setSpacing(5);
```

```
vbox.setPadding(new Insets(10, 0, 0, 10));
```

```
vbox.getChildren().addAll(tips, table, returnButton);
```

```
vbox.setAlignment(Pos.CENTER);
```

```
vbox.setId("mainPane");
```

```
//加入到主面板
```

```
((Group) scene.getRoot()).getChildren().addAll(vbox);
```

```
scene.getStylesheets().add(getClass().getResource("writeScorestyle.css").toExternalForm());
```

```
//搭建舞台
```

```
primaryStage.setScene(scene);
```

```
primaryStage.show();
```

```
}
```

```

/*****
*****/

```

```

public class Student {
    private final SimpleStringProperty id;
    private final SimpleStringProperty name;
    private final SimpleStringProperty gaomath;
    private final SimpleStringProperty Eng;
    private final SimpleStringProperty cpp;
    private final SimpleStringProperty JavaScores;
    private final SimpleStringProperty OS;
    private final SimpleStringProperty sport;
    private final SimpleStringProperty average;
    private final SimpleStringProperty rank;

    Student(String id, String name,String JavaScores, String
gaomath,String Eng, String OS, String sport,
            String cpp, String average,String rank)
    {
        this.id = new SimpleStringProperty(id);
        this.name = new SimpleStringProperty(name);
        this.cpp = new SimpleStringProperty(cpp);
        this.gaomath = new SimpleStringProperty(gaomath);
        this.Eng = new SimpleStringProperty(Eng);
        this.JavaScores =new SimpleStringProperty(JavaScores);
        this.OS=new SimpleStringProperty(OS);
        this.sport=new SimpleStringProperty(sport);
        this.average=new SimpleStringProperty(average);
        this.rank=new SimpleStringProperty(rank);
    }

    public String getId() {
        return this.id.get();
    }
    public void setId(String id) {
        this.id.set(id);
    }
    public String getCpp() {
        return this.cpp.get();
    }
    public void setCpp(String cpp) {
        this.id.set(cpp);
    }
    public String getName() {

```



```

        return this.name.get();
    }
    public void setName(String name) {
        this.name.set(name);
    }
    public String getGaomath() {
        return this.gaomath.get();
    }
    public void setGaomath(String gaomath) {
        this.gaomath.set(gaomath);
    }
    public String getEng() {
        return this.Eng.get();
    }
    public void setEng(String gaomath) {
        this.Eng.set(gaomath);
    }
    public String getJavaScores() {
        return this.JavaScores.get();
    }
    public void setJavaScores(String JavaScores) {
        this.JavaScores.set(JavaScores);
    }
    public String getOS() {
        return this.OS.get();
    }
    public void setOS(String OS) {
        this.OS.set(OS);
    }
    public String getSport() {
        return this.sport.get();
    }
    public void setSport(String sport) {
        this.sport.set(sport);
    }
    public String getAverage() {
        return this.average.get();
    }
    public void setAverage(String average) {
        this.average.set(average);
    }
    public String getRank() {
        return this.rank.get();
    }
}

```

```

        public void setRank(String rank) {
            this.rank.set(rank);
        }
    }

    /*****
    *****/

    //从 data.txt 中读取数据传给表格
    private void getProductsFromFile() {
        try {
            BufferedReader br = new BufferedReader(new
InputStreamReader(new FileInputStream("data.txt")));
            String line;
            String[] fields;
            while ((line = br.readLine()) != null) {
                fields = line.split(" ");
                double temp =
Double.parseDouble(fields[2])+Double.parseDouble(fields[3])+Double.pa
rseDouble(fields[4])

+Double.parseDouble(fields[5])+Double.parseDouble(fields[6])+Double.p
arseDouble(fields[7]);
                temp = temp/6;
                temp = Math.round(temp*100)/100;
                table.getItems().add(new Student(fields[1],
fields[0], fields[2], fields[3], fields[4], fields[5],
                fields[6],
fields[7],String.valueOf(temp),"1"));
            }
            br.close();
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}

```