

“較大型的視窗應用程式:我會整合 *GradeBook2D*+ 檔案存取+*JavaFX*”

GradeBook 成績處理類別與檔案存取

陣列運用

字串處理

自訂類別 *GradeBookD*

基本 *JavaFX* 元件靈活使用

排序

成績計算

其他自行定義的功能!

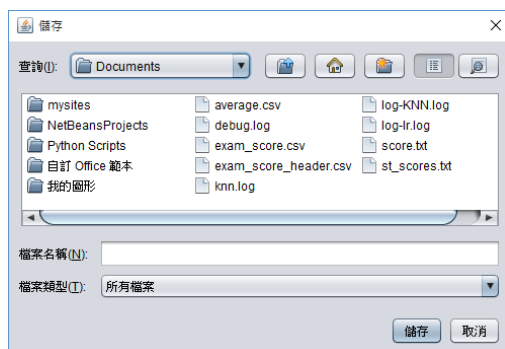
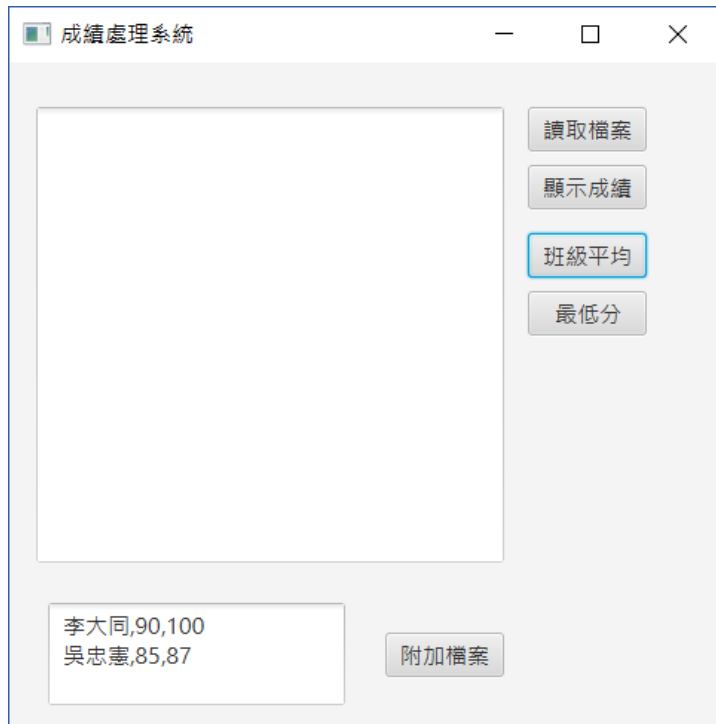
新學到 Java 功能：

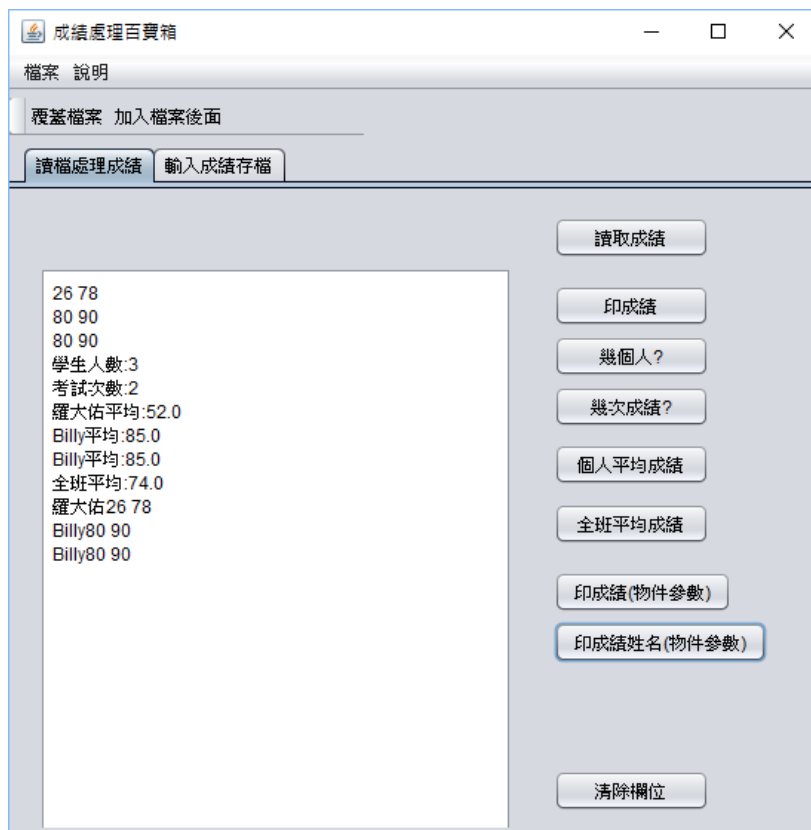
- 檔案存取時的選單
- 檔案附加方式寫檔
- 幾個新的 GUI 元件

學了今天的課程之後，你可以開始嘗試寫一個較大型的視窗應用程式。

成績計算功能程式

參考畫面如下：





兩種方式:

寫法 1: 將計算功能切開給 GradeBook.java 類別負責

寫法 2: 成績計算功能寫在同一個程式裡面(不是個好方式)。

寫法 1: 將計算功能切開給 GradeBook.java 類別負責

整合 GradeBook 成績計算功能程式

● 字串切割

字串切割注意事項:

分隔符號 delimiter: 一個空白

```
String s = new String("20 30 50")
```

```
String[] ss = s.split(" ")
```

分隔符號: 1 個 2 個或多個空白

`s.split(" ")` 這就不管用了!

分隔符號有太多情況了，有沒有更厲害的方法?

```
String s = new String("20 30 50")
```

```
String[] ss = s.split("\\s+")
```

`\s` 表示 space 空白

`+` 代表 1, 2, 3, ... 個

備註:

- `\\` 表示會把 `\s` 當成一般的字串對待，而不是 `\n \t` 的那種溢出字元，若只寫 `\s` 會被當成溢出字元看待
- “`\`”是溢出字元(escape)，有其特殊用途，如果我們存放的文字剛好有這些溢出字元要怎麼辦? 打兩個斜線`\\`，就可以了! 以這種方式告訴系統，這不是溢出字元。

分隔符號: 一個逗號

```
String s = new String("20,30,50")
```

```
String[] ss = s.split(",")
```

分隔符號: 一個逗號 後面可能有空白

```
String s = new String("20, 30, 50")
```

`s.split(",")` 就不管用了!

```
String[] ss = s.split(",\\s*")
```

`\s` 表示 space 空白

* 代表 0,1, 2, 3, ...個

分隔符號:一個逗號 後面可能有空白 前面也可能有空白

```
String[] ss = s.split("\\s*,\\s*")
```

這種一堆符號組成的特殊寫法有其特殊意義，頗為複雜，可以用在文字處理、抓網頁(爬蟲)，這種寫法稱為：**正規表達式 Regular Expression**

FXMLDocumentController

```
public class FXMLDocumentController implements Initializable {

    GradeBook2D gb = new GradeBook2D();

    @FXML
    private TextArea display;

    @FXML
    private TextArea input;
    private String[] names;
    private int[][] grades;
    private String fileName = "score.csv";
    //private File file; //沒用到

    @FXML
    private Button btn_classAvg;

    @FXML
    private Button btn_min;

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        // TODO
    }

    @FXML
    private void read_file(ActionEvent event) {
```

```

        //選取檔案
        FileChooser fileChooser = new FileChooser();
        fileChooser.setTitle("請選取檔案");
        File selectedFile = fileChooser.showOpenDialog(null);

        if (selectedFile != null) {
            //fileName = selectedFile.getAbsolutePath();
            display.setText(fileName + "<--開啟檔名\n");
            openFile(selectedFile);
            gb.setGrades(grades);
        } else {
            return;
        }
    }

    @FXML
    private void read_default_file(ActionEvent event) {
        File file = new File(fileName);
        display.setText(file.getAbsolutePath() + "\n");
        openFile(file);
        gb.setGrades(grades);
    }

    @FXML
    private void print_all(ActionEvent event) {
        display.appendText(gb.outputGrades());
    }

    @FXML
    private void get_classAvg(ActionEvent event) {
        String msg = String.format("班平均分: %.2f\n", gb.get_classAvg());
        display.appendText(msg);
    }

    @FXML
    private void get_min(ActionEvent event) {
        String msg = String.format("最低分: %d\n", gb.getMinimum());
        display.appendText(msg);
    }

```

```
}

public void openFile(File file) {

    String row = "";
    String line = "";
    int student_size = 0;
    int numgrades = 0; //紀錄有幾次成績

    try {
        Scanner input = new Scanner(file);
        while (input.hasNextLine()) {
            line = input.nextLine();
            System.out.println("---" + line);
            if (!line.isEmpty()) {
                student_size++;
                row = line;
            }
        }
        System.out.printf("學生人數:%d \n", student_size);
        numgrades = row.split(",").length - 1;
        System.out.printf("成績個數:%d \n", numgrades);
        input.close();
    } catch (FileNotFoundException ex) {
        System.out.println("檔案讀取錯誤!");
    }

    names = new String[student_size];
    grades = new int[student_size][numgrades];

    try {
        Scanner input = new Scanner( file );
        int stu = 0;
        while (input.hasNextLine()) {
            line = input.nextLine();
            System.out.println("---" + line);
            if (!line.isEmpty()) {
                String[] rec = line.split(","); //這裡改成各種分隔符號都可以!!
```

```

        names[stu] = rec[0];

        for (int j = 0; j < numgrades; j++) {
            grades[stu][j] = Integer.parseInt(rec[j + 1]);
        }
        stu++;
    }
}

} catch (FileNotFoundException ex) {
    System.out.println("檔案讀取錯誤!");
}

}

```

@FXML

```
private void append_file(ActionEvent event) {
```

```

    //選擇檔案或輸入檔案名稱存檔，若檔案已存在，資料加入檔案後面
    //注意:附加資料通常會出錯，因為沒有換行符號，資料會黏在一起。
    // 若多一個換行符號，卻又產生空白行，需要特別處理
    //輸入資料最行一行請換行，這樣附上資料，才不會黏在一起

```

```

FileChooser fileChooser = new FileChooser();
fileChooser.setTitle("選擇檔案或輸入檔案名稱附加資料");
File savedFile = fileChooser.showSaveDialog(null);

```

```
if (savedFile != null) {
```

```
    try {
```

```
        Formatter output = new Formatter(new FileWriter(savedFile, true)); //資料加入
```

檔案後面

```

        output.format(input.getText() + "\n"); //塞入一個換行符號
        output.flush();
        output.close();
    } catch (IOException e) {

```

```
        display.setText("存檔錯誤!");
    }
}
```



```

        return;
    }
    display.setText("存檔於:" + savedFile.toString());
} else {
    display.setText("File save cancelled.");
}
}

@FXML
private void replace_file(ActionEvent event) {

    //選擇檔案或輸入檔案名稱存檔 若檔案已存在會被覆寫
    FileChooser fileChooser = new FileChooser();
    fileChooser.setTitle("選擇檔案或輸入檔案名稱存檔");
    File savedFile = fileChooser.showSaveDialog(null);
    if (savedFile != null) {
        try {
            Formatter output = new Formatter(savedFile); //資料加入檔案後面
            output.format(input.getText()+ "\n");
            output.flush();
            output.close();
        } catch (IOException e) {
            display.setText("存檔錯誤!");
            return;
        }
        display.setText("存檔於: " + savedFile.toString());
    } else {
        display.setText("File save cancelled.");
    }
}
}

```

****寫檔案有兩種方式:覆寫與附加**

****存檔附加在檔案的後面**

```
File savedFile = new File("score.csv");
try {
    Formatter output = new Formatter(new FileWriter(savedFile, true)); //資料加入檔案後面
    output.format("%s\n", input.getText());
    output.close();
} catch (IOException ex) {
    System.out.println("存檔錯誤");
}
```

split()用法

計算有幾次考試成績

```
numScore = row.split(",").length - 1;
```

上面一行可以寫成以下兩行

```
String s[] = row.split(",");
numScore = s.length-1;
```

計算之功能寫在 GradeBook 類別內

其他有趣的功能請自行添加

GradeBook.java

```
package gradebook;

public class GradeBook2D {

    private int[][] grades;
    private String[] names;
    private double[] averages;

    public GradeBook2D() { //建構子 1
    }

    public GradeBook2D(int[][] grades) { //建構子 2
        this.grades = grades;
    }
}
```

```
public GradeBook2D(int grades[[]], String[] names) //建構子 3
{
    this.names = names;
    this.grades = grades;
}

public void setGrades(int[][] grades) {
    this.grades = grades;
}

public int getNumberOfScore() {
    return this.grades[0].length;
}

public int getNumberOfStudent() {
    return grades.length;
}

public String outputGrades() {
    String msg = "";
    for (int student = 0; student < grades.length; student++) {
        msg += String.format("學生序號:%2d", student + 1);
        for (int test : grades[student]) {
            msg += String.format("%3d", test);
        }
        double average = getAvg(grades[student]);
        msg += String.format("%9.2f\n", average);
    }
    return msg;
}

public double get_classAvg() {
    int total = 0;
    int count = 0;

    for (int i = 0; i < grades.length; i++) {
        for (int j = 0; j < grades[i].length; j++) {
```

```
        total += grades[i][j];
        count++;
    }
}
return (double) total / count;
}

public double getAvg(int[] setOfGrades) {
    int total = 0;

    for (int grade : setOfGrades) {
        total += grade;
    }
    return (double) total / setOfGrades.length;
}

public int getMinimum() {
    int lowGrade = grades[0][0];
    for (int[] studentGrades : grades) {
        for (int grade : studentGrades) {
            if (grade < lowGrade) {
                lowGrade = grade;
            }
        }
    }

    return lowGrade;
}

public int getFailed() {
    int count = 0;
    for (int i = 0; i < grades.length; i++) {
        for (int j = 0; j < grades[i].length; j++) {
            if (grades[i][j] < 60) {
                count++;
            }
        }
    }
}
```

```

        return count;
    }

    //計算每位學生平均，以一維陣列回傳
    public double[] getStuAverages() {
        for (int i = 0; i < grades.length; i++) {
            int total = 0;
            for (int j = 0; j < grades[i].length; j++) {
                total = total + grades[i][j];
            }
            averages[i] = (double) total / grades[i].length;
        }
        return averages;
    }
}

```

回家作業：

讀入一個學生成績的 CSV 檔案，挑一個版本去處理即可：

版本 1: 只有各次考試成績沒有姓名

版本 2: 姓名加上各次考試成績

按鈕按下去之後，在文字顯示區域顯示出以下分數資訊：

- * 有多少個學生？
- * 每個學生有多少次成績？
- * 計算每個學生的平均分數
- * 計算每個學生的最高分數
- * 計算每個學生的最低分數
- * 計算全班的最高分數
- * 計算全班的最低分數
- * 計算第 1 次考試的全班平均、計算第 2 次考試的全班平均、....
- * 計算全班的平均分數
- * 平均成績分布表 以 10 分為等地 或以 ABCDF 為等第

你可以設計好看的功能按鈕：

整合 GradeBook 成績計算功能程式--擴充版

每位學生的測驗次數不一定一樣，請修改程式可以應付此情況。