

國立高雄科技大學-智慧商務系

資料結構

MVC-成績管理

智慧商務系三乙

C109156213 李翊曲

資料結構期末作業

壹、 介紹

一、 動機

藉由本學期的資料結構課程，習得了需多相關的知識和用途，故藉本次期末作業來完成過去 MVC 課程的程式碼修正，來實現資料結構的技能運用。

二、 系統簡介

登入畫面後，可以查看學生資料，師生雙方可以看見課程資料表，進行課程的新增或選課，再由修課資料表來查看選課成果。而學期成績則可以看到學生的上下學期成績和科目，並利用 QuickSort 進行學期成績的排序。

學生資料表		課程資料表(師)		課程資料表(生)		修課資料表		學期成績	
班級	學號	姓名	住址	電話	生日	新增學生資料			
C101	C10101	L1	台中市	096624XXXX	2010-01-07	修改	刪除		
C101	10105	K1	台南市	092915XXXX	2010-04-07	修改	刪除		

貳、 專案規劃：

一、 修改部分

(一)、 QuickSort

```
export default function GradesInfo(){
  Request().get("/index.php?action=getGrades")
  .then(res => {
    let response = res['data'];
    console.log(response);
    switch(response['status']){
      case 200:
        const rows = response['result'];
        let str = `<table border='1'>`;
        str += `<button id='newGrades'>新增成績</button></td></tr>`;
```

```
export default function GradesInfo() {
  Request()
  .get("/index.php?action=getGrades")
  .then((res) => {
    let response = res["data"];
    console.log(response);
    switch (response["status"]) {
      case 200:
        let rows = response["result"];
        rows.sort((a, b) => {
          return b.aavg - a.aavg;
        });
        let str = `<table border='1'>`;
        str += `<button id='newGrades'>新增成績</button></td></tr>`;
```

學生資料表	課程資料表(師)	課程資料表(生)	修課資料表	學期成績
新增成績				
C10101				
科目	上學期成績	下學期成績	科目平均	
國文	0	0	0.0000	
英文	46	24	35.0000	
數學	61	0	30.5000	
平均成績	8.0000			修改 刪除
期末成果	國文應重修	英文應重修	數學應重修	
C10102				
科目	上學期成績	下學期成績	科目平均	
國文	99	99	99.0000	
英文	45	97	71.0000	
數學	85	62	73.5000	
平均成績	86.0000			修改 刪除
期末成果				
C10103				
科目	上學期成績	下學期成績	科目平均	
國文	25	66	45.5000	
英文	35	99	67.0000	
數學	58	85	71.5000	
平均成績	83.3333			修改 刪除
期末成果	國文應重修			

排序前

學生資料表	課程資料表(師)	課程資料表(生)	修課資料表	學期成績
新增成績				
C10102				
科目	上學期成績	下學期成績	科目平均	
國文	99	99	99.0000	
英文	45	97	71.0000	
數學	85	62	73.5000	
平均成績	86.0000			修改 刪除
期末成果				
C10103				
科目	上學期成績	下學期成績	科目平均	
國文	25	66	45.5000	
英文	35	99	67.0000	
數學	58	85	71.5000	
平均成績	83.3333			修改 刪除
期末成果	國文應重修			
C10101				
科目	上學期成績	下學期成績	科目平均	
國文	0	0	0.0000	
英文	46	24	35.0000	
數學	61	0	30.5000	
平均成績	8.0000			修改 刪除
期末成果	國文應重修	英文應重修	數學應重修	

排序後

(二)、堆疊(Stack):

```
rows.forEach((element) => {
  str += `<tr><td id='id' align="center" colspan="4">` + element["stdid"] + `</td></tr>`;
  str += `<td>科目</td><td>上學期成績</td><td>下學期成績</td><td>科目平均</td></tr>`;
  str += `<tr>`;
  str += `<tr>` + `<td>國文</td>` + `<td align="center">` + element["cnmid"] + `<td align="center">` + element["cnfin"] +
    `</td>` + `<td align="center">` + element["cna"] + `</td>`;
  str += `<tr>` + `<td>英文</td>` + `<td align="center">` + element["enmid"] + `</td>` + `<td align="center">` + element["enfin"] +
    `</td>` + `<td align="center">` + element["ena"] + `</td>`;
  str += `<tr>` + `<td>數學</td>` + `<td align="center">` + element["mamid"] +
    `</td>` + `<td align="center">` + element["mafin"] + `</td>` + `<td align="center">` + element["maa"] + `</td>`;
  str += `<tr>` + `<td>平均成績</td>` + `<th align="center" colspan="4">` + element["aver"] + `</th>`;
  str += `<td><button id='updateGrades'>修改</button><button id='deleteGrades'>刪除</button></td>`;
  str += `</tr>`;
  str += `<tr>` + `<td>期末成果</td>`;
```

```
rows.forEach((element) => {
  stack.push(`<tr><td id='id' align="center" colspan="4">` + element["stdid"] + `</td></tr>`);
  stack.push(`<td>科目</td><td>上學期成績</td><td>下學期成績</td><td>科目平均</td></tr>`);
  stack.push(`<tr>`);
  stack.push(`<tr>` + `<td>國文</td>` + `<td align="center">` + element["cnmid"] + `<td align="center">` + element["cnfin"] +
    stack.push(`<tr>` + `<td>英文</td>` + `<td align="center">` + element["enmid"] + `</td>` + `<td align="center">` + element["enfin"] +
    stack.push(`<tr>` + `<td>數學</td>` + `<td align="center">` + element["mamid"] + `</td>` + `<td align="center">` + element["mafin"] +
    stack.push(`<tr>` + `<td>平均成績</td>` + `<th align="center" colspan="4">` + element["aver"] + `</th>`);
  stack.push(`<td><button id='updateGrades'>修改</button><button id='deleteGrades'>刪除</button></td></tr>`);
  stack.push(`<tr>`);
  stack.push(`<tr>` + `<td>期末成果</td>`);
```

利用堆疊的功能將所有的結果項目儲存起來，並在循環完成後一起添加到字符串中。這樣就可以在循環中逐步構建表格，而不是在循環中逐步添加行。

參、專案成果

一、開發歷程

進度表					
	1/24	1/25	1/26	1/27	1/28
資料查找					
程式編寫					
測試及修改					
word 撰寫					

肆、結論與心得

本次修改部分雖然不多，但在查詢和修正錯誤的過程中也是獲益良多。在學習資料結構時，我發現它們對於解決問題和提高程式效率有著重要的作用。例如，使用陣列進行資料儲存時，查詢和修改的效率較低，但如果使用鏈結串列，則可以增加查詢和修改的效率。此外，學習演算法如快速排序和合併排序，可以幫助我們對大量資料進行有效的排序。