C（下）考核方式详细说明

1. **单元测试（占50%）：**
   * 相当于原来的期末考试。
   * 单元测试一共**5次**，分别是**二维数组、字符串、指针与数组、结构体、链表**，每次**20分**。每次5套试卷以上，每套试卷包括1-5题选择题，2道编程题，选择题1分1题，编程题会有不同难度。
   * 如果需要**补考也是机考**，试卷时长2小时，5-10道选择题，3道编程题，分别是字符串，二维数组，链表各一题。
2. **平时成绩（20%）：**
   * 可由作业和课堂互动成绩共同组成。课堂互动成绩由所在班级的任课老师确定。
   * 作业仍然在练习系统上完成，与上学期累计到一起，至少**2000分为满分**，其他分数对应折算。
   * 练习时长明显过短，或练习平均行数明显过少的学生会**酌情扣分**。
3. **大作业（30%）：**
   * 需要完成一个规模较大的程序设计，题目可以任选（例如：小学口算练习系统，一些小型游戏，简单数据库管理系统（成绩、书籍、资产、通讯录管理等）等），至少也要在12章和13章综合例题的基础上所有改进或变化。可以从第1周开始就着手准备。
   * 可以一人一组，也可以2-3人一组，但平均每个学生所交程序的代码量在300行以上。
   * 最后4学时进行现场答辩和评分。多人一组的由教师随机指定一人讲解，作为全组所有成员的分数。
   * 学生答辩前，需要提前填写好附件1，并按表1中答辩要求陈述和展示。答辩后交给任课老师存档。
   * 每一个题目只有一次答辩机会，请提前认真准备。
   * 抄袭者按0分计。

表1 大作业考核要求

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 考核要求 | | | | |
| **创新性（满分10分）：** 题目和算法等的新颖程度  （教材和教师授课内容以外的题目和算法） | **综合性（满分20分）：** 知识点覆盖范围是否全面，一维数组、二维数组、字符串、指针、链表或结构体、文件等知识点，至少覆盖其中4条。否则缺少一项，扣5分 | **规范性（满分10分）：** 例如：标识符和函数命名规范、函数结构划分合理，函数参数设置合理，表达式的书写和语句简单明了，输入输出人性化、适当的注释等。 | **功能性及工作量（满分40分）:** 功能合理、完善； 每人编写代码300行以上 | **答辩成绩（满分20分）**需要清晰陈述创新点，难点，全面展示代码中包含的知识点和规范性，正确回答教师和其他学生的问题。 |

附件1：

**《C语言程序设计》开放项目实验报告**

|  |  |
| --- | --- |
| 题目： | 学生与课程管理系统 |
| 学号： | 2019210595 |
| 姓名： | 华尹 |
| 亮点（创新性）： | 不仅设计了学生的管理系统，同时也完成了课程的管理系统，并将学生与课程这两中结果关联起来，添加了更多的交叉联系的关系。 |
| 知识点包括： | ☑一维数组 □二维数组 ☑字符串 ☑指针  ☑链表或结构体 □文件 |
| 各功能模块（函数）名称及其功能描述： | // chooseController  void initChooseOS(); 初始化选课系统  void addChooseHandler(); 添加选课处理器  void updateScoreHandler(); 更新选课成绩处理器  void seekChooseHandler(); 查看选课处理器  void printChoose(ReturnedChoose \*returnedChoose); 打印选课信息  // courseController  void initCourseOS(); 初始化课程系统  void addCourseHandler(); 添加课程处理器  void removeCourseHandler(); 删除课程处理器  void seekCourseHandler(); 查看课程处理器  void seeAllCourseHandler(); 查看全部课程处理器  void printCourse(Course \*course); 打印课程信息  // studentController  void initStudentOS(); 初始化学生系统  void addStudentHandler(); 添加学生处理器  void removeStudentHandler(); 删除学生处理器  void seekStudentHandler(); 查看学生信息处理器  void seeAllStudentHandler(); 查看全部学生信息处理器  void printStudent(Student \*student); 打印学生信息  // chooseModal  ReturnedChoose \* newReturnedChoose(); 创建选课返回实例  ReturnedChoose \*addChoose(int studentId, int courseId); 添加选课  ReturnedChoose \*seekStudentScore(int studentId); 查看选课成绩  ReturnedChoose \*updateScore(int studentId, int courseId, int score); 更新选课成绩  // courseModal  ReturnedCourse \* newReturnedCourse(); 创建课程返回实例  ReturnedCourse \*addCourse(int id, char name[nameLen], int teacherId); 添加课程  ReturnedCourse \*removeCourse(int id); 删除课程  ReturnedCourse \*seekCourse(int id); 查看课程  Courses\* getAllCourse(); 获取全部课程  int courseCount(); 获取课程数量  int findCourse(int id); 查找指定id课程所在索引（二分）  int findCourseLowerBound(int id); 查找小于id的最大的课程所在的索引  // studentModal  ReturnedStudent \* newReturnedStudent(); 创建学生返回实例  Students \*initStudentModal(); 初始化学生modal  ReturnedStudent \*addStudent(int id, char name[nameLen], int classNum); 添加学生信息  ReturnedStudent \*removeStudent(int id); 删除学生信息  ReturnedStudent \*updateStudent(int id, char newName[nameLen], int newClassNum); 更新学生信息  ReturnedStudent \*seekStudent(int id); 查看学生信息  ReturnedStudent \*addStudentCourse(int studentId, int courseId); 添加学生选课信息  ReturnedStudent \*updateStudentScore(int studentId, int courseId, int score); 更新学生选课成绩  int hasStudentCourse(int studentIndex, int courseId); 判断学生是否选了指定课程  Students\* getAllStudent(); 获取全部学生信息  int studentCount(); 获取学生数量  int findStudent(int id); 查找指定id学生所在索引（二分）  int findStudentLowerBound(int id); 查找小于id的最大的课程所在的索引  // controllerUtil  int checkoutInputNum(char \*input, int maxNum); 校验输入数字的合法性  void exitSystem(int exitSymbol); 退出系统辅助函数  // dataStruct  CourseStudent \* newCourseStudent(); 创建课程中的学生信息实例  Course \* newCourse(); 创建课程实例  StudentCourse \* newStudentCourse(); 创建学生中的课选课信息实例  Student \* newStudent(); 创建学生实例  // viewUtil  void clearScreen(); 清屏辅助函数  void renderLeft\_(); 渲染左屏函数  void renderPaddingLeft\_(); 渲染padding-left函数  void renderPaddingTop\_(); 渲染padding-top函数  void renderTop\_(); 渲染上屏函数  void renderTransverseLine\_(); 渲染横线函数  // view/main  void renderIndex(); 渲染主页信息  void renderStudent(); 渲染学生系统信息  void renderCourse(); 渲染课程系统信息  void renderChoose(); 渲染选课系统信息 |
| 完成过程中遇到的主要困难： | 1. 在c语言的模块化开发工作中，首先要解决的是构造出类似包级别的命名空间，如何去对外开放部分公开接口，如何去限制部分函数不被外部使用。这里我将可被外界使用的函数签名写在h头文件中，在编写其他模块c文件时，可以导入h头文件，这样编辑器也会帮我们自动导入对应的‘包’。 2. 接上述解决方案之后，在编译时会报错：未定义的symbol，其原因是因为我们导入的h头文件并没有那些函数的定义，仅有他们的签名，编译器在编译时不会自动导入同名的c文件，所以我们需要将导入h头文件改为导入c文件，这样就解决了未定义symbol的问题。 3. 接上述解决方案之后，在编译时会报错：重复定义的symbol，原因是，在h文件中，通常我们会写在预处理器中，防止头文件的内容被重复定义。导入的c文件中，并没有进行重复定义的判断处理，所以我们要在被导入的c文件中添加是否定义的预处理器判断处理。如： #ifndef PROJECT\_CHOOSEMODAL\_C #define PROJECT\_CHOOSEMODAL\_C   #endif //PROJECT\_CHOOSEMODAL\_C   1. 在开放整个系统时，为了稍微贴近实际，尽量结偶各模块，采用mvc架构模式，通过controller作为主要处理器来完成业务逻辑的处理。modal层管理满足业务需求的业务模型。view层管理前台所展示的视图。如果我们需要进行ui的重构，改成GUI界面，只需要更改view层的代码即可。 |
| 收获及心得体会： | 通过一个大作业的形式，尽量多的使用了学到的c语言经典知识，同时对于mvc的理解也更深了一步，通过mvc三层的解耦，让我对项目代码有了更加清晰的认识和把握，更加深刻地理解了mvc三层解耦的思想，给了我更多的在项目中模块解耦上的思路与方向。主要学习了如何去通过原生的c语言去完成一个小项目的流程，也踩了很多的坑，通过不断地去谷歌，不断阅读论坛文章，最终才能解决问题。 |
| 答辩时回答问题情况（答辩时填写） | 正确回答了 个问题中的 个问题。 |

程序全部代码：

// main.c

// main.c

int main() {

initStudentModal();

initCourseModal();

char input[10] = "";

while (1) {

render(IndexView);

printf("Please input the number you'd like to do:\n");

scanf("%s", input);

if (checkoutInputNum(input, 6)) {

// "1. 学生相关",

// "2. 课程相关",

// "3. 选课相关",

// "4. 重新进入（清屏）"

// "5. 退出本系统"

if (strcmp(input, "1") == 0) {

initStudentOS();

} else if (strcmp(input, "2") == 0) {

initCourseOS();

} else if (strcmp(input, "3") == 0) {

initChooseOS();

} else if (strcmp(input, "4") == 0) {

continue;

} else if (strcmp(input, "5") == 0) {

exitSystem(0);

return 0; // 只是为了消除 clion 的警告提示，才加的一个return

}

} else {

puts("Invalid number! Please input valid number:");

}

}

}

// view/main.h

//

// Created by firework on 2021/5/19.

//

#include "../utils/view/viewUtil.h"

#ifndef PROJECT\_MAIN\_H

#define PROJECT\_MAIN\_H

const int IndexView = 0;

const int StudentView = 1;

const int CourseView = 2;

const int ChooseView = 3;

void render(int);

void renderView(int);

void renderIndex();

void renderStudent();

void renderCourse();

void renderChoose();

void renderLabels(char labels[][LabelLen], int labelNum);

#endif //PROJECT\_MAIN\_H

// view/main.c

//

// Created by firework on 2021/5/19.

//

#include "main.h"

#include "../utils/view/viewUtil.c"

#include <stdio.h>

#ifndef PROJECT\_MAIN\_C

#define PROJECT\_MAIN\_C

void render(int viewCategory) {

clearScreen();

renderView(viewCategory);

}

void renderView(int viewCategory) {

switch (viewCategory) {

case IndexView:

return renderIndex();

case StudentView:

return renderStudent();

case CourseView:

return renderCourse();

case ChooseView:

return renderChoose();

}

}

void renderLabels(char labels[][LabelLen], int labelNum) {

for (int i = 0; i < labelNum; i ++) {

renderLeft\_();

printf("|");

renderPaddingLeft\_();

for (int j = 0; j < LabelLen; j ++) {

int c = labels[i][j] == '\0' ? ' ' : labels[i][j];

putchar(c);

}

renderPaddingLeft\_();

printf("|\n");

}

}

void renderCategory\_(char labels[][LabelLen], int labelNum) {

renderTop\_();

renderLeft\_();

putchar('|');

renderTransverseLine\_();

puts("|");

renderPaddingTop\_();

renderLabels(labels, labelNum);

renderPaddingTop\_();

renderLeft\_();

putchar('|');

renderTransverseLine\_();

puts("|");

renderTop\_();

}

char IndexViewLabels[][LabelLen] = {

// "1. 学生相关",

// "2. 课程相关",

// "3. 选课相关",

// "4. 重新进入（清屏）"

// "5. 退出本系统"

"1. Student os",

"2. Course os",

"3. Choose course os",

"4. Refresh screen",

"5. Exit system"

};

void renderIndex() {

renderCategory\_(IndexViewLabels, 5);

}

char studentViewLabels[][LabelLen] = {

// "1. 添加学生信息",

// "2. 删除学生信息",

// "3. 查看学生信息",

// "4. 查看全部学生信息",

// "5. 返回到主页",

// "6. 重新进入（清屏）"

// "7. 退出本系统"

"1. add student info",

"2. remove student info",

"3. check student info",

"4. see all students info",

"5. back to index",

"6. refresh screen",

"7. exit system"

};

void renderStudent() {

renderCategory\_(studentViewLabels, 7);

}

char CourseViewLabels[][LabelLen] = {

// 1. 添加课程

//// 2. 删除课程

//// 3. 查看课程

//// 4. 查看全部课程

// "5. 返回到主页",

// "6. 重新进入（清屏）"

// "7. 退出本系统"

"1. add course",

"2. remove course",

"3. check course info",

"4. check all course info",

"5. back to index",

"6. refresh screen",

"7. exit system"

};

void renderCourse() {

renderCategory\_(CourseViewLabels, 7);

}

char ChooseViewLabels[][LabelLen] = {

// 1. 选课

// 2. ~~退选~~

// 3. 成绩录入

// 4. 查看学生选课信息

"1. add choose for student",

"2. update score for student",

"3. seek chose for student",

"4. back to index",

"5. refresh screen",

"6. exit system"

};

void renderChoose() {

renderCategory\_(ChooseViewLabels, 6);

}

#endif //PROJECT\_MAIN\_C

// utils/view/viewUtil.h

//

// Created by firework on 2021/5/19.

//

#ifndef PROJECT\_VIEW\_UTIL\_H

#define PROJECT\_VIEW\_UTIL\_H

const int LabelLen = 30;

const int TOP = 2;

const int LEFT = 20;

const int PADDING = 10;

const int PADDING\_TOP = 1;

void clearScreen();

#endif //PROJECT\_VIEW\_UTIL\_H

// utils/view/viewUtil.c

//

// Created by firework on 2021/5/19.

//

#include "viewUtil.h"

#include <stdio.h>

#include <stdlib.h>

#ifndef PROJECT\_VIEW\_UTIL\_C

#define PROJECT\_VIEW\_UTIL\_C

void clearScreen() {

system("clear");

}

void renderLeft\_() {

for (int i = 0;i < LEFT; i ++)

putchar(' ');

}

void renderPaddingLeft\_() {

for (int i = 0;i < PADDING; i ++)

printf(" ");

}

void renderPaddingTop\_() {

for (int j = 0; j < PADDING\_TOP; j ++) {

renderLeft\_();

putchar('|');

int len = PADDING \* 2 + LabelLen;

for (int i = 0; i < len; i++) {

putchar(' ');

}

puts("|");

}

}

void renderTop\_() {

for (int i = 0; i < TOP; i ++)

printf("\n");

}

void renderTransverseLine\_() {

int len = PADDING \* 2 + LabelLen;

for (int i = 0; i < len ; i ++)

putchar('-');

}

#endif //PROJECT\_VIEW\_UTIL\_C

// utils/modal/dataStruct.h

//

// Created by firework on 2021/5/19.

//

#include <stdlib.h>

#include <string.h>

#ifndef PROJECT\_DATASTRUCT\_H

#define PROJECT\_DATASTRUCT\_H

const int nameLen = 20;

typedef struct CourseStudent {

int studentId;

struct CourseStudent \*next;

} CourseStudent;

CourseStudent \* newCourseStudent() {

CourseStudent \* courseStudent = (CourseStudent \*) malloc(sizeof(CourseStudent));

courseStudent->studentId = -1;

courseStudent->next = NULL;

return courseStudent;

}

typedef struct Course {

int id;

char name[nameLen];

CourseStudent \*student;

int teacherId;

} Course;

Course \* newCourse() {

Course \*course = (Course \*) malloc(sizeof(Course));

strcpy(course->name, "");

course->id = -1;

course->student = newCourseStudent();

course->teacherId = -1;

return course;

}

typedef struct StudentCourse {

int courseId;

int score;

struct StudentCourse \* next;

} StudentCourse;

StudentCourse \* newStudentCourse() {

StudentCourse \* studentCourse = (StudentCourse \*) malloc(sizeof(StudentCourse));

studentCourse->courseId = -1;

studentCourse->score = -1;

studentCourse->next = NULL;

return studentCourse;

}

typedef struct Student {

int id;

int classNum;

char name[nameLen];

StudentCourse \*course;

double totalScore; // 学生各科总成绩

int status; // 学生的状态，0 不存在、1 存在、2 已删除

} Student;

Student \* newStudent() {

Student \*student = (Student \*) malloc(sizeof(Student));

student->id = -1;

student->classNum = -1;

strcpy(student->name, "");

student->course = newStudentCourse();

student->totalScore = -1;

student->status = 0;

return student;

}

#endif //PROJECT\_DATASTRUCT\_H

// utils/controller/controllerUtil.h

//

// Created by firework on 2021/5/19.

//

#include <stdlib.h>

#include <string.h>

#ifndef PROJECT\_DATASTRUCT\_H

#define PROJECT\_DATASTRUCT\_H

const int nameLen = 20;

typedef struct CourseStudent {

int studentId;

struct CourseStudent \*next;

} CourseStudent;

CourseStudent \* newCourseStudent() {

CourseStudent \* courseStudent = (CourseStudent \*) malloc(sizeof(CourseStudent));

courseStudent->studentId = -1;

courseStudent->next = NULL;

return courseStudent;

}

typedef struct Course {

int id;

char name[nameLen];

CourseStudent \*student;

int teacherId;

} Course;

Course \* newCourse() {

Course \*course = (Course \*) malloc(sizeof(Course));

strcpy(course->name, "");

course->id = -1;

course->student = newCourseStudent();

course->teacherId = -1;

return course;

}

typedef struct StudentCourse {

int courseId;

int score;

struct StudentCourse \* next;

} StudentCourse;

StudentCourse \* newStudentCourse() {

StudentCourse \* studentCourse = (StudentCourse \*) malloc(sizeof(StudentCourse));

studentCourse->courseId = -1;

studentCourse->score = -1;

studentCourse->next = NULL;

return studentCourse;

}

typedef struct Student {

int id;

int classNum;

char name[nameLen];

StudentCourse \*course;

double totalScore; // 学生各科总成绩

int status; // 学生的状态，0 不存在、1 存在、2 已删除

} Student;

Student \* newStudent() {

Student \*student = (Student \*) malloc(sizeof(Student));

student->id = -1;

student->classNum = -1;

strcpy(student->name, "");

student->course = newStudentCourse();

student->totalScore = -1;

student->status = 0;

return student;

}

#endif //PROJECT\_DATASTRUCT\_H

// utils/controller/controllerUtil.c

//

// Created by firework on 2021/5/20.

//

#include "controllerUtil.h"

#include <string.h>

#include <stdlib.h>

#include <stdio.h>

#include <unistd.h>

#ifndef PROJECT\_CONTROLLERUTIL\_C

#define PROJECT\_CONTROLLERUTIL\_C

int checkoutInputNum(char \*input, int maxNum) {

for (int i = 1; i <= maxNum; i ++) {

char a = '0';

a += i;

char t[2] = {a};

if (strcmp(input, t) == 0) {

return 1;

}

}

return 0;

}

void exitSystem(int exitSymbol) {

puts("you will exit system after 3 seconds.");

puts("well");

sleep(1);

puts("ok");

sleep(1);

puts("good bye ~");

sleep(1);

exit(exitSymbol);

}

#endif //PROJECT\_CONTROLLERUTIL\_C

// modal/studentModal.h

//

// Created by firework on 2021/5/19.

//

#include "../utils/modal/dataStruct.h"

#ifndef PROJECT\_STUDENTMODAL\_H

#define PROJECT\_STUDENTMODAL\_H

const int studentLen = 100;

// type

typedef struct ReturnedStudent {

Student \*student;

int ok; // 返回的状态，0 失败、1 成功

} ReturnedStudent;

ReturnedStudent \* newReturnedStudent() {

ReturnedStudent \* returnedStudent = (ReturnedStudent \*) malloc(sizeof(ReturnedStudent));

returnedStudent->student = NULL;

returnedStudent->ok = 0;

return returnedStudent;

}

typedef Student\* Students;

Students \*initStudentModal();

ReturnedStudent \*addStudent(int id, char name[nameLen], int classNum);

ReturnedStudent \*removeStudent(int id);

ReturnedStudent \*updateStudent(int id, char newName[nameLen], int newClassNum);

ReturnedStudent \*seekStudent(int id);

ReturnedStudent \*addStudentCourse(int studentId, int courseId);

ReturnedStudent \*updateStudentScore(int studentId, int courseId, int score);

Students\* getAllStudent();

int studentCount();

int findStudent(int id);

int findStudentLowerBound(int id);

#endif //PROJECT\_STUDENTMODAL\_H

// modal/studentModal.c

//

// Created by firework on 2021/5/19.

//

#include "studentModal.h"

#include "courseModal.c"

#include <string.h>

#ifndef PROJECT\_STUDENTMODAL\_C

#define PROJECT\_STUDENTMODAL\_C

int studentCount\_ = 0;

Student \*students\_[studentLen];

Students \*initStudentModal() {

for (int i = 0; i < studentLen; i ++)

students\_[i] = newStudent();

return students\_;

};

// 暂时不考虑学生数量过多的情况，仅在一个 studentLen 大小的 数组中存储

ReturnedStudent \*addStudent(int id, char name[nameLen], int classNum) {

if (findStudent(id) != -1) {

return newReturnedStudent();

}

Student \*student = newStudent();

student->id = id;

student->classNum = classNum;

strcpy(student->name, name);

int index = findStudentLowerBound(id), cnt = studentCount();

while (cnt > index) {

students\_[cnt] = students\_[cnt - 1];

cnt --;

}

students\_[index] = student;

studentCount\_ ++;

// 这里不写 students\_[studentCount\_ ++] 是为了保险起见，减少风险

// 其实可以单独为针对 studentCount\_ 的操作写几个函数来尽量确保 其原子性及安全性，但是懒

ReturnedStudent \*res = newReturnedStudent();

res->student = student;

res->ok = 1;

return res;

}

ReturnedStudent \*removeStudent(int id) {

int index = findStudent(id), cnt = studentCount();

ReturnedStudent \*res = newReturnedStudent();

if (index == -1) {

return res;

}

res->student = students\_[index];

res->ok = 1;

while (index < cnt-1) {

students\_[index] = students\_[index + 1];

index ++;

}

studentCount\_ --;

students\_[studentCount\_] = newStudent();

return res;

};

ReturnedStudent \*updateStudent(int id, char newName[nameLen], int newClassNum) {

int index = findStudent(id);

ReturnedStudent \*res = seekStudent(id);

if (index == -1) {

return res;

}

res->student->classNum = newClassNum;

strcpy(res->student->name, newName);

return res;

};

// 二分

ReturnedStudent \*seekStudent(int id) {

int index = findStudent(id);

ReturnedStudent \* res = newReturnedStudent();

if (index == -1) {

return res;

}

res->student = students\_[index];

res->ok = 1;

return res;

};

ReturnedStudent \*addStudentCourse(int studentId, int courseId) {

int index = findStudent(studentId);

Student \*student = students\_[index];

StudentCourse \* studentCourse = student->course;

while (studentCourse->next != NULL)

studentCourse = studentCourse->next;

studentCourse->courseId = courseId;

studentCourse->next = newStudentCourse();

ReturnedStudent \*returnedStudent = newReturnedStudent();

returnedStudent->student = student;

returnedStudent->ok = 1;

return returnedStudent;

}

ReturnedStudent \*updateStudentScore(int studentId, int courseId, int score) {

int index = findStudent(studentId);

Student \*student = students\_[index];

StudentCourse \* studentCourse = student->course;

while (studentCourse->courseId != courseId)

studentCourse = studentCourse->next;

if (studentCourse->next == NULL) {

return newReturnedStudent();

}

studentCourse->score = score;

ReturnedStudent \*returnedStudent = newReturnedStudent();

returnedStudent->student = student;

returnedStudent->ok = 1;

return returnedStudent;

};

int hasStudentCourse(int studentIndex, int courseId) {

StudentCourse \* studentCourse = students\_[studentIndex]->course;

while (studentCourse->courseId != courseId && studentCourse->next != NULL)

studentCourse = studentCourse->next;

return studentCourse->courseId == courseId;

}

Students\* getAllStudent() {

return students\_;

};

int studentCount() {

if (studentCount\_ == -1) {

int i = 0;

studentCount\_ = 0;

while (students\_[i]->status == 1) {

studentCount\_ ++;

i ++;

}

}

return studentCount\_;

}

// 二分

int findStudent(int id) {

if (studentCount() <= 0) {

return -1;

}

int l = 0, r = studentCount() - 1, mid;

// [l ,r]

while (l <= r) {

mid = l + (r - l) / 2;

if (students\_[mid]->id == id) {

return mid;

}

else if (students\_[mid]->id < id) {

l = mid + 1;

} else {

r = mid - 1;

}

}

return -1;

}

int findStudentLowerBound(int id) {

if (studentCount() <= 0) {

return 0;

}

int i = 0, cnt = studentCount\_;

while (i < cnt && students\_[i]->id < id) {

i ++;

}

return i;

}

#endif //PROJECT\_STUDENTMODAL\_C

// modal/courseModal.h

//

// Created by firework on 2021/5/20.

//

#include "../utils/modal/dataStruct.h"

#ifndef PROJECT\_COURSEMODAL\_H

#define PROJECT\_COURSEMODAL\_H

const int courseLen = 100;

// type

typedef struct ReturnedCourse {

Course \*course;

int ok; // 返回的状态，0 失败、1 成功

} ReturnedCourse;

ReturnedCourse \* newReturnedCourse() {

ReturnedCourse \* returnedCourse = (ReturnedCourse \*) malloc(sizeof(ReturnedCourse));

returnedCourse->course = NULL;

returnedCourse->ok = 0;

return returnedCourse;

}

typedef Course\* Courses;

Courses \*initCourseModal();

ReturnedCourse \*addCourse(int id, char name[nameLen], int teacherId);

ReturnedCourse \*removeCourse(int id);

ReturnedCourse \*seekCourse(int id);

Courses\* getAllCourse();

int courseCount();

int findCourse(int id);

int findCourseLowerBound(int id);

#endif //PROJECT\_COURSEMODAL\_H

// modal/courseModal.c

//

// Created by firework on 2021/5/20.

//

#include "courseModal.h"

#include <string.h>

#ifndef PROJECT\_COURSEMODAL\_C

#define PROJECT\_COURSEMODAL\_C

int courseCount\_ = 0;

Course \*courses\_[courseLen];

Courses \*initCourseModal() {

for (int i = 0; i < courseLen; i ++)

courses\_[i] = newCourse();

return courses\_;

};

// 暂时不考虑学生数量过多的情况，仅在一个 courseLen 大小的 数组中存储

ReturnedCourse \*addCourse(int id, char name[nameLen], int teacherId) {

if (findCourse(id) != -1) {

return newReturnedCourse();

}

Course \*course = newCourse();

course->id = id;

course->teacherId = teacherId;

strcpy(course->name, name);

int index = findCourseLowerBound(id), cnt = courseCount();

while (cnt > index) {

courses\_[cnt] = courses\_[cnt - 1];

cnt --;

}

courses\_[index] = course;

courseCount\_ ++;

// 这里不写 courses\_[courseCount\_ ++] 是为了保险起见，减少风险

// 其实可以单独为针对 courseCount\_ 的操作写几个函数来尽量确保 其原子性及安全性，但是懒

ReturnedCourse \*res = newReturnedCourse();

res->course = course;

res->ok = 1;

return res;

}

ReturnedCourse \*removeCourse(int id) {

int index = findCourse(id), cnt = courseCount();

ReturnedCourse \*res = newReturnedCourse();

if (index == -1) {

return res;

}

res->course = courses\_[index];

res->ok = 1;

while (index < cnt-1) {

courses\_[index] = courses\_[index + 1];

index ++;

}

courseCount\_ --;

courses\_[courseCount\_] = newCourse();

return res;

};

// 二分

ReturnedCourse \*seekCourse(int id) {

int index = findCourse(id);

ReturnedCourse \* res = newReturnedCourse();

if (index == -1) {

return res;

}

res->course = courses\_[index];

res->ok = 1;

return res;

};

Courses\* getAllCourse() {

return courses\_;

};

int courseCount() {

if (courseCount\_ == -1) {

int i = 0;

courseCount\_ = 0;

while (courses\_[i]->id != -1) {

courseCount\_ ++;

i ++;

}

}

return courseCount\_;

}

// 二分

int findCourse(int id) {

if (courseCount() <= 0) {

return -1;

}

int l = 0, r = courseCount() - 1, mid;

// [l ,r]

while (l <= r) {

mid = l + (r - l) / 2;

if (courses\_[mid]->id == id) {

return mid;

}

else if (courses\_[mid]->id < id) {

l = mid + 1;

} else {

r = mid - 1;

}

}

return -1;

}

int findCourseLowerBound(int id) {

if (courseCount() <= 0) {

return 0;

}

int i = 0, cnt = courseCount\_;

while (i < cnt && courses\_[i]->id < id) {

i ++;

}

return i;

}

#endif //PROJECT\_COURSEMODAL\_C

// modal/chooseModal.h

//

// Created by firework on 2021/5/20.

//

#include "../utils/modal/dataStruct.h"

#ifndef PROJECT\_CHOOSEMODAL\_H

#define PROJECT\_CHOOSEMODAL\_H

// type

typedef struct ReturnedChoose {

Student \*student;

int ok; // 返回的状态，0 失败、1 成功

} ReturnedChoose;

ReturnedChoose \* newReturnedChoose() {

ReturnedChoose \* returnedChoose = (ReturnedChoose \*) malloc(sizeof(ReturnedChoose));

returnedChoose->student = NULL;

returnedChoose->ok = 0;

return returnedChoose;

}

ReturnedChoose \*addChoose(int studentId, int courseId);

ReturnedChoose \*seekStudentScore(int studentId);

ReturnedChoose \*updateScore(int studentId, int courseId, int score);

#endif //PROJECT\_CHOOSEMODAL\_H

// modal/chooseModal.c

//

// Created by firework on 2021/5/20.

//

#include "chooseModal.h"

#include "studentModal.c"

#include "courseModal.c"

#ifndef PROJECT\_CHOOSEMODAL\_C

#define PROJECT\_CHOOSEMODAL\_C

ReturnedChoose \*addChoose(int studentId, int courseId) {

int stuIndex = findStudent(studentId);

if (stuIndex == -1) {

return newReturnedChoose();

}

int courseIndex = findCourse(courseId);

if (courseIndex == -1) {

return newReturnedChoose();

}

ReturnedChoose \* res = newReturnedChoose();

ReturnedStudent \*returnedStudent = addStudentCourse(studentId, courseId);

res->student = returnedStudent->student;

res->ok = returnedStudent->ok;

return res;

}

ReturnedChoose \*seekStudentScore(int studentId) {

int stuIndex = findStudent(studentId);

if (stuIndex == -1) {

return newReturnedChoose();

}

ReturnedStudent \*returnedStudent = seekStudent(studentId);

ReturnedChoose \*res = newReturnedChoose();

res->student = returnedStudent->student;

res->ok = returnedStudent->ok;

return res;

}

ReturnedChoose \*updateScore(int studentId, int courseId, int score) {

int stuIndex = findStudent(studentId);

if (stuIndex == -1) {

return newReturnedChoose();

}

if (!hasStudentCourse(stuIndex, courseId)) {

return newReturnedChoose();

}

ReturnedStudent \* returnedStudent = updateStudentScore(studentId, courseId, score);

ReturnedChoose \*res = newReturnedChoose();

res->student = returnedStudent->student;

res->ok = returnedStudent->ok;

return res;

};

#endif //PROJECT\_CHOOSEMODAL\_C

// controller/studentController.h

//

// Created by firework on 2021/5/20.

//

#include "../modal/studentModal.h"

#ifndef PROJECT\_STUDENTCONTROLLER\_H

#define PROJECT\_STUDENTCONTROLLER\_H

void initStudentOS();

void addStudentHandler();

void removeStudentHandler();

void seekStudentHandler();

void seeAllStudentHandler();

void printStudent(Student \*student);

#endif //PROJECT\_STUDENTCONTROLLER\_H

// controller/studentController.c

//

// Created by firework on 2021/5/20.

//

#include "studentController.h"

#include "../view/main.c"

#include "../modal/studentModal.c"

#include "../utils/controller/controllerUtil.c"

#include <stdio.h>

#include <string.h>

#ifndef PROJECT\_STUDENTCONTROLLER\_C

#define PROJECT\_STUDENTCONTROLLER\_C

void initStudentOS() {

render(StudentView);

char input[10] = "";

while (1) {

printf("Please input the number you'd like to do:\n");

scanf("%s", input);

if (checkoutInputNum(input, 6)) {

// "1. 添加学生信息",

// "2. 删除学生信息",

// "3. 查看学生信息",

// "4. 查看全部学生信息",

// "5. 返回到主页",

// "6. 重新进入（清屏）"

// "7. 退出本系统"

if (strcmp(input, "1") == 0) {

addStudentHandler();

} else if (strcmp(input, "2") == 0) {

removeStudentHandler();

} else if (strcmp(input, "3") == 0) {

seekStudentHandler();

} else if (strcmp(input, "4") == 0) {

seeAllStudentHandler();

} else if (strcmp(input, "5") == 0) {

return;

} else if (strcmp(input, "6") == 0) {

puts("Will refresh screen 2 seconds later");

sleep(2);

render(StudentView);

} else if (strcmp(input, "7") == 0) {

return exitSystem(0);

}

} else {

puts("Invalid number! Please input valid number:");

}

}

}

void addStudentHandler() {

int id, classNum;

char name[nameLen];

puts("Please input info of student with the format: '{id}-{classNum}-{name}'");

puts("ex. 00-01-Jhon");

scanf("%d-%d-%s", &id, &classNum, name);

ReturnedStudent \*res = addStudent(id, name, classNum);

if (!res->ok) {

puts("\nFailed to add student. err: repetitive id.\n");

} else {

puts("\nSuccessfully added student");

puts("Student:");

printStudent(res->student);

puts("");

}

}

void removeStudentHandler() {

int id;

puts("Please input id of student to remove with the format: '{id}'");

puts("ex. 00");

scanf("%d", &id);

ReturnedStudent \*res = removeStudent(id);

if (!res->ok) {

puts("\nFailed to add student. err: id not found.\n");

} else {

puts("\nSuccessfully removed student");

puts("Student:");

printStudent(res->student);

puts("");

}

}

void seekStudentHandler() {

int id;

puts("Please input id of student to seek with the format: '{id}'");

puts("ex. 00");

scanf("%d", &id);

ReturnedStudent \*res = seekStudent(id);

if (!res->ok) {

puts("\nFailed to seek student. err: id not found.\n");

} else {

puts("\nSuccessfully seek student");

puts("Student:");

printStudent(res->student);

puts("");

}

}

void seeAllStudentHandler() {

Students\* students = getAllStudent();

Student student;

int cnt = studentCount();

if (cnt == 0) {

puts("There is no student.\n");

return;

}

for (int i = 0; i < cnt; i ++) {

student = \*students[i];

puts("Student:");

printStudent(&student);

}

}

void printStudent(Student \*student) {

printf("id:%d\tname:%s\tclassNum:%d\n", student->id, student->name, student->classNum);

}

#endif //PROJECT\_STUDENTCONTROLLER\_C

// controller/courseController.h

//

// Created by firework on 2021/5/20.

//

#include "../modal/courseModal.h"

#ifndef PROJECT\_COURSECONTROLLER\_H

#define PROJECT\_COURSECONTROLLER\_H

void initCourseOS();

void addCourseHandler();

void removeCourseHandler();

void seekCourseHandler();

void seeAllCourseHandler();

void printCourse(Course \*course);

#endif //PROJECT\_COURSECONTROLLER\_H

// controller/courseController.c

//

// Created by firework on 2021/5/20.

//

#include "courseController.h"

#include "../view/main.c"

#include "../modal/courseModal.c"

#include "../utils/controller/controllerUtil.c"

#ifndef PROJECT\_COURSECONTROLLER\_C

#define PROJECT\_COURSECONTROLLER\_C

void initCourseOS() {

render(CourseView);

char input[10] = "";

while (1) {

printf("Please input the number you'd like to do:\n");

scanf("%s", input);

if (checkoutInputNum(input, 6)) {

// 1. 添加课程

//// 2. 删除课程

//// 3. 查看课程

//// 4. 查看全部课程

// "5. 返回到主页",

// "6. 重新进入（清屏）"

// "7. 退出本系统"

if (strcmp(input, "1") == 0) {

addCourseHandler();

} else if (strcmp(input, "2") == 0) {

removeCourseHandler();

} else if (strcmp(input, "3") == 0) {

seekCourseHandler();

} else if (strcmp(input, "4") == 0) {

seeAllCourseHandler();

} else if (strcmp(input, "5") == 0) {

return;

} else if (strcmp(input, "6") == 0) {

puts("Will refresh screen 2 seconds later");

sleep(2);

render(CourseView);

} else if (strcmp(input, "7") == 0) {

return exitSystem(0);

}

} else {

puts("Invalid number! Please input valid number:");

}

}

};

void addCourseHandler() {

int id, teacherId;

char name[nameLen];

puts("Please input info of course with the format: '{id}-{name}-{teacherName}'");

puts("ex. 00-01-Jhon");

scanf("%d-%d-%s", &id, &teacherId, name);

ReturnedCourse \*res = addCourse(id, name, teacherId);

if (!res->ok) {

puts("\nFailed to add course. err: repetitive id.\n");

} else {

puts("\nSuccessfully added course");

puts("course:");

printCourse(res->course);

puts("");

}

};

void removeCourseHandler() {

int id;

puts("Please input id of course to remove with the format: '{id}'");

puts("ex. 00");

scanf("%d", &id);

ReturnedCourse \*res = removeCourse(id);

if (!res->ok) {

puts("\nFailed to add course. err: id not found.\n");

} else {

puts("\nSuccessfully removed course");

puts("course:");

printCourse(res->course);

puts("");

}

};

void seekCourseHandler() {

int id;

puts("Please input id of course to seek with the format: '{id}'");

puts("ex. 00");

scanf("%d", &id);

ReturnedCourse \*res = seekCourse(id);

if (!res->ok) {

puts("\nFailed to seek course. err: id not found.\n");

} else {

puts("\nSuccessfully seek course");

puts("course:");

printCourse(res->course);

puts("");

}

};

void seeAllCourseHandler() {

Courses\* courses = getAllCourse();

Course course;

int cnt = courseCount();

if (cnt == 0) {

puts("There is no course.\n");

return;

}

for (int i = 0; i < cnt; i ++) {

course = \*courses[i];

puts("course:");

printCourse(&course);

}

puts("");

};

void printCourse(Course \*course) {

printf("id:%d\tname:%s\tteacherId:%d\n", course->id, course->name, course->teacherId);

};

#endif //PROJECT\_COURSECONTROLLER\_C

// controller/chooseController.h

//

// Created by firework on 2021/5/20.

//

#include "../modal/studentModal.h"

#include "../modal/chooseModal.h"

#ifndef PROJECT\_CHOOSECONTROLLER\_H

#define PROJECT\_CHOOSECONTROLLER\_H

void initChooseOS();

void addChooseHandler();

void updateScoreHandler();

void seekChooseHandler();

void printChoose(ReturnedChoose \*returnedChoose);

#endif //PROJECT\_CHOOSECONTROLLER\_H

// controller/chooseController.c

//

// Created by firework on 2021/5/20.

//

#include "chooseController.h"

#include "../view/main.c"

#include "../modal/chooseModal.c"

#include "../utils/controller/controllerUtil.c"

#ifndef PROJECT\_CHOOSECONTROLLER\_C

#define PROJECT\_CHOOSECONTROLLER\_C

void initChooseOS() {

render(ChooseView);

char input[10] = "";

while (1) {

printf("Please input the number you'd like to do:\n");

scanf("%s", input);

if (checkoutInputNum(input, 6)) {

// "1. add choose for student",

// "2. update score for student",

// "3. seek chose for student",

// "4. back to index",

// "5. refresh screen",

// "6. exit system"

if (strcmp(input, "1") == 0) {

addChooseHandler();

} else if (strcmp(input, "2") == 0) {

updateScoreHandler();

} else if (strcmp(input, "3") == 0) {

seekChooseHandler();

} else if (strcmp(input, "4") == 0) {

return;

} else if (strcmp(input, "5") == 0) {

puts("Will refresh screen 2 seconds later");

sleep(2);

render(ChooseView);

} else if (strcmp(input, "6") == 0) {

return exitSystem(0);

}

} else {

puts("Invalid number! Please input valid number:");

}

}

}

void addChooseHandler() {

int stuId, courseId;

puts("Please input info of studentId and courseId with the format: '{stuId}-{courseId}'");

puts("ex. 00-01");

scanf("%d-%d", &stuId, &courseId);

ReturnedChoose \*res = addChoose(stuId, courseId);

if (!res->ok) {

puts("\nFailed to choose course for student. err: id not found.\n");

} else {

puts("\nSuccessfully choose course for student.");

puts("Chose:");

printChoose(res);

puts("");

}

};

void updateScoreHandler() {

int stuId, courseId, score;

puts("Please input info of studentId and courseId with the format: '{stuId}-{courseId}-{score}'");

puts("ex. 00-01-98");

scanf("%d-%d-%d", &stuId, &courseId, &score);

ReturnedChoose \*res = updateScore(stuId, courseId, score);

if (!res->ok) {

puts("\nFailed to update score for student. err: id not found.\n");

} else {

puts("\nSuccessfully update score for student.");

puts("Chose:");

printChoose(res);

puts("");

}

};

void seekChooseHandler() {

int stuId;

puts("Please input info of studentId with the format: '{stuId}'");

puts("ex. 00");

scanf("%d", &stuId);

ReturnedChoose \*res = seekStudentScore(stuId);

if (!res->ok) {

puts("\nFailed to seek student's scores. err: id not found.\n");

} else {

puts("\nSuccessfully choose seek student's scores.");

puts("Chose:");

printChoose(res);

puts("");

}

};

void printChoose(ReturnedChoose \*returnedChoose) {

int stuId = returnedChoose->student->id;

StudentCourse \*studentCourse = returnedChoose->student->course;

while (studentCourse->next != NULL) {

printf("studentId:%d, courseId:%d, score:%d\n", stuId, studentCourse->courseId, studentCourse->score);

studentCourse = studentCourse->next;

}

};

#endif //PROJECT\_CHOOSECONTROLLER\_C