**程序设计报告**

**项目名称**

**学生成绩管理系统**

**项目简介**

学生成绩管理系统是一款基于 C 语言开发的简易成绩管理软件，旨在帮助教育管理者对学生成绩进行有效的录入、查询、修改、删除、以及排序和保存。系统支持多班级的学生成绩管理，具有数据文件的动态保存功能，用户友好的交互界面，以及严格的输入校验，确保数据完整性和稳定性。

**功能描述**

**主要功能**

1. **录入学生成绩**：
   * 用户可以为指定班级录入学生信息，包括姓名、学号、性别、课程及其对应成绩。
   * 系统限制最多录入 10 个班级，每个班级最多容纳 100 名学生。
2. **查询课程平均分**：
   * 用户可查询某门课程在指定班级的平均成绩。
3. **查询学生成绩及排名**：
   * 用户可以通过学号查询某学生的所有课程成绩、总成绩及班级排名。
4. **修改学生成绩**：
   * 修改某学生的某门课程成绩，并更新总成绩。
5. **删除学生成绩**：
   * 删除某学生的某门课程成绩，并更新总成绩。
6. **按学号排序输出学生总分及排名**：
   * 按学号升序排列学生，并输出班级总成绩及排名。
7. **动态文件保存**：
   * 程序在每次修改后，自动将班级成绩数据保存到对应班级的文件中，文件命名规则为 class\_X.txt。

**代码实现**

**目录结构**

项目代码模块化设计，按功能划分为多个模块，各模块具有独立的头文件和源文件：

StuGradeSystem/

├── main.c // 主程序

├── data.h // 全局数据结构定义

├── data.c // 全局变量定义

├── utils.h // 工具函数声明

├── utils.c // 工具函数实现

├── input.h // 成绩录入模块声明

├── input.c // 成绩录入模块实现

├── query.h // 成绩查询模块声明

├── query.c // 成绩查询模块实现

├── modify.h // 成绩修改模块声明

├── modify.c // 成绩修改模块实现

├── delete.h // 成绩删除模块声明

├── delete.c // 成绩删除模块实现

├── sort.h // 排序模块声明

├── sort.c // 排序模块实现

**主要代码示例**

**1. 数据结构定义 (data.h)**

#ifndef DATA\_H

#define DATA\_H

#define MAX\_CLASSES 10 // 最大班级数量

#define MAX\_STUDENTS 100 // 每个班级最大学生人数

#define MAX\_COURSES 5 // 每个学生最多课程数

#define MAX\_NAME\_LEN 50 // 学生姓名最大长度

#define MAX\_ID\_LEN 15 // 学号最大长度

#define MAX\_SUBJECT\_LEN 50 // 科目名称最大长度

#define DATA\_FOLDER "Data" // 数据保存文件夹

// 成绩结构体

typedef struct {

char subject[MAX\_SUBJECT\_LEN]; // 科目名称

float score; // 成绩

} Course;

// 学生结构体

typedef struct {

char name[MAX\_NAME\_LEN]; // 姓名

char id[MAX\_ID\_LEN]; // 学号

char gender; // 性别

Course courses[MAX\_COURSES]; // 课程数组

int courseCount; // 课程数量

float totalScore; // 总成绩

} Student;

// 班级结构体

typedef struct {

char className[MAX\_NAME\_LEN]; // 班级名称

Student students[MAX\_STUDENTS]; // 学生数组

int studentCount; // 学生数量

} Class;

// 全局班级数组

extern Class classes[MAX\_CLASSES];

#endif

**2. 文件夹创建及保存功能 (utils.c)**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/stat.h>

#include <direct.h>

#include "data.h"

#include "utils.h"

// 创建 Data 文件夹

void createDataFolder() {

struct stat st = {0};

if (stat(DATA\_FOLDER, &st) == -1) {

\_mkdir(DATA\_FOLDER);

printf("Data 文件夹已创建！\n");

}

}

// 保存班级数据到文件

void saveClassDataToFile(int classIndex) {

if (classIndex < 0 || classIndex >= MAX\_CLASSES) {

printf("无效的班级索引。\n");

return;

}

char filePath[150];

snprintf(filePath, sizeof(filePath), "%s/class\_%d.txt", DATA\_FOLDER, classIndex);

FILE\* file = fopen(filePath, "w");

if (!file) {

printf("无法保存数据到文件。\n");

return;

}

fprintf(file, "Class: %s\n", classes[classIndex].className);

for (int i = 0; i < classes[classIndex].studentCount; i++) {

fprintf(file, "Student: %s %s %c %.2f\n", classes[classIndex].students[i].id,

classes[classIndex].students[i].name, classes[classIndex].students[i].gender,

classes[classIndex].students[i].totalScore);

for (int j = 0; j < classes[classIndex].students[i].courseCount; j++) {

fprintf(file, " Course: %s %.2f\n", classes[classIndex].students[i].courses[j].subject,

classes[classIndex].students[i].courses[j].score);

}

}

fclose(file);

printf("班级数据已保存到：%s\n", filePath);

}

**3. 成绩录入功能 (input.c)**

void addStudentScore(int classIndex) {

if (classIndex < 0 || classIndex >= MAX\_CLASSES) {

printf("无效的班级索引。\n");

return;

}

if (classes[classIndex].studentCount >= MAX\_STUDENTS) {

printf("班级已满，无法添加更多学生。\n");

return;

}

Student\* student = &classes[classIndex].students[classes[classIndex].studentCount];

printf("请输入学生姓名：");

scanf\_s("%s", student->name, (unsigned int)sizeof(student->name));

printf("请输入学号：");

scanf\_s("%s", student->id, (unsigned int)sizeof(student->id));

while (1) {

printf("请输入性别（M/F）：");

scanf\_s(" %c", &student->gender, 1);

if (student->gender == 'M' || student->gender == 'F' || student->gender == 'm' || student->gender == 'f') {

break;

}

printf("输入无效，请输入 M 或 F。\n");

}

printf("请输入课程数量（最多 %d 门）：", MAX\_COURSES);

if (scanf\_s("%d", &student->courseCount) != 1 || student->courseCount < 1 || student->courseCount > MAX\_COURSES) {

printf("课程数量无效。\n");

return;

}

float totalScore = 0;

for (int i = 0; i < student->courseCount; i++) {

printf("请输入课程 %d 名称：", i + 1);

scanf\_s("%s", student->courses[i].subject, (unsigned int)sizeof(student->courses[i].subject));

printf("请输入课程 %d 成绩：", i + 1);

if (scanf\_s("%f", &student->courses[i].score) != 1) {

printf("成绩输入无效。\n");

return;

}

totalScore += student->courses[i].score;

}

student->totalScore = totalScore;

classes[classIndex].studentCount++;

saveClassDataToFile(classIndex);

printf("学生成绩录入成功！\n");

}

**测试结果**

**测试数据**

1. 班级数量：10，学生数量：100。
2. 每个学生录入 2-5 门课程。

**测试场景**

1. 正常录入数据：测试录入完整性和文件生成。
2. 输入非法值（超出范围的索引、无效的性别、课程数量不符）：验证错误提示。
3. 修改与删除学生成绩：验证数据更新。

**测试结果**

| **功能** | **测试用例** | **预期结果** | **实际结果** |
| --- | --- | --- | --- |
| 创建文件夹 | Data 文件夹首次运行时不存在 | 自动创建 Data 文件夹 | 通过 |
| 录入学生成绩 | 输入有效数据 | 成功录入数据并保存到文件 | 通过 |
| 输入性别非法值 | 输入除 M/F 外的字符 | 提示错误并重新输入 | 通过 |
| 删除学生成绩 | 输入学号和课程名称 | 删除课程并更新总成绩 | 通过 |
| 班级已满 | 超过 100 名学生 | 提示班级已满，无法录入 | 通过 |

**可读性与注释**

1. **模块化设计**：每个功能独立实现，减少耦合。
2. **注释全面**：每个函数都带有功能说明。
3. **变量命名清晰**：遵循命名规范，体现用途。

**开源声明**

本项目 **“学生成绩管理系统”** 已在 GitHub 平台开源，遵循 **Apache License 2.0** 协议进行发布。

**版权声明**

Copyright 2024 Jiacheng Ni

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

**项目仓库**

项目源码及相关文档托管在 GitHub 仓库，欢迎访问：

**GitHub 仓库地址**: <https://github.com/XXYoLoong/StuGradeSystem.git>

**许可说明**

* 您可以自由地复制、修改、分发、合并、发布本项目的代码。
* 使用时需遵守 **Apache License 2.0** 协议，需保留原作者版权声明和许可协议文本。
* 若您对本项目代码进行修改并再发布，请提供适当的说明以告知变更内容。

**使用声明**

1. **免责声明**
   * 本项目为学习和研究目的开发，代码中的任何功能仅为演示，使用时请根据具体需求修改。
   * 作者对因使用本项目代码所导致的任何直接或间接后果概不负责。
2. **社区支持**
   * 欢迎提交 Issue 或 Pull Request 对项目提出改进建议。

**联系我们**

如有任何问题、建议或需求，您可以通过以下方式联系作者：

* GitHub Issues: <https://github.com/XXYoLoong/StuGradeSystem/issues>
* 邮箱:yoloong05@gmail.com

**致谢**

感谢所有为本项目贡献代码、提交 Issue 或提出建议的开发者，您的支持是本项目持续改进的动力！