卡通画

描述已自动生成

**程序设计课程设计报告**

题 目\_\_学生自习打卡管理系统\_

学 院\_\_\_计算机学院\_\_\_\_\_\_\_\_

专 业\_\_\_计算机 \_\_

年级班别\_\_ 2020级（13）班

学 号\_\_\_\_\_3120005209\_\_\_\_\_\_\_

学生姓名\_\_\_\_\_李子浩\_\_\_\_\_\_\_\_\_\_\_

指导教师\_\_\_\_李小妹 \_\_\_\_\_\_\_\_\_

成 绩

|  |  |
| --- | --- |
| 程序功能完成情况 | □完整 □基本完整 □不完整 |
| 测试用例全面情况 | □全面 □基本全面 □不全面 |
| 报告格式是否与要求相符 | □规范 □基本规范 □不规范 |
| 报告内容是否准确全面 | □全面 □基本全面 □不全面 |

**2020年1月5日**

1. **总体设计**

图示

描述已自动生成

1. **详细设计**
2. 功能模块
3. 主函数（管理函数）

图示

描述已自动生成

1. 添加学生打卡信息函数

图示, 示意图

描述已自动生成

1. 按学号查询打卡信息函数（包括学生班级、姓名以及打卡次数）

图示

描述已自动生成

1. 按学号删除打卡信息函数

图示

描述已自动生成

1. 打印班级所有学生所在打卡统计函数

说明：读取结构体studentsClock并打印（studentsClock在输入班级号加载数据时赋值）

1. 读取文件数据函数（readData）

说明：根据输入班级名读取数据文件，并计算统计信息保存到结构体studentsClock

1. 写入文件函数（将学生打卡信息按行写入文件）
2. 读取文件函数（从文件中将打卡信息按行读取）

图示

描述已自动生成

1. 公用数据结构及用法说明

学生打卡数据结构体clockData

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 说明 |
| id | char[] | 学号 |
| class | char[] | 班级 |
| name | char[] | 姓名 |
| clockInData | char[] | 打卡日期（年月日） |

班级内所有学生统计信息结构体ClassClockTotal

|  |  |  |
| --- | --- | --- |
| 字段 | 字段类型 | 说明 |
| id | char\* | 学号 |
| name | char\* | 姓名 |
| clockCount | int | 打卡次数 |

**用法说明：macOS和Linux可直接运行**

1. **试验结果**
2. 进入班级信息

图片包含 图示

描述已自动生成

1. 添加学生打卡信息

文本

描述已自动生成

1. 按学号查询学生

一些文字和图片

描述已自动生成

1. 按学号删除学生打卡信息

文本

描述已自动生成

文本

描述已自动生成

1. 本班级打卡信息统计

一些文字和图案

描述已自动生成

1. **体会总结**

（1）此次课程设计完成了按学号查找学生打卡信息、按学号删除学生打卡信息、扩展功能为添加打卡信息。

（2）需改进的地方有：

1、C语言结构体的使用与C语言函数的互相调用还不能够熟练使用。

2、由于平时写的程序不够多，且使用的是macOS写的程序，使用了windows里没有（linux里有）的头文件，无法很好的解决移植性的问题。

3、程序还无法做到方便快捷的使用。

（3）调试程序：对我来说，几乎不可能一次就通过编译，且通过编译后功能也不一定正确，所以花了很多时间debug。一开始遇到bug时总会心情急躁，但后来逐渐明白出现bug是编程的一个重要部分，出错才能改错，才能知道自己的疏漏与不足。

（4）收获：C语言是计算机程序设计的重要理论基础，在我们以后的学习和工作中都有着十分重要的地位。通过此次课程设计，我明白了要学好这种语言，仅仅靠学习课本上的知识是远远不够的，还需要经常自己动手写程序，只有亲自动手实践，多编写程序，才能发现我们学习上的漏洞和不足，并在实践中解决这些问题，不断提高自己转化知识的能力。

本次的课程设计需要使用文件来存储信息，但我平时也没接触过，因此，刚开始时有较大困难。但我先将书本上的内容认真地看了一遍，然后再通过课后题来增进自己的理解，最终才有了突破。除此之外，遇到的另一个难点就是使用流程图来表达自己的想法，因为只有真正理解这个程序的结构才能作出流程图，这个作业正培养了我这方面的能力。这些就是这次做课程设计的收获。

**附录：**

**main.c:**

#include <stdio.h>

#include <unistd.h>

#include "fileoperate.c"

#include "myfunctions.c"

/\*主函数（管理函数）\*/

int main()

{

printf("\t\t★ --欢迎进入学生自习打卡管理系统-- ★\n");

// 1.首先创建数据文件夹（其中保存打卡数据）

createFolder("data");

// 2.输入要管理的班级

Flag1:;

char className[30];

char judgeClassName[40];

printf("请输入您的班级:");

scanf("%s", className);

// 根据路径创建文件夹名

sprintf(judgeClassName, "data/%s", className);

// 3.判断班级文件夹是否存在

if (access(judgeClassName, 0) == -1) {

// 班级不存在则请求是否根据班级名创建文件夹

printf("抱歉，您输入的班级[%s]不存在\n是否创建文件夹?(1.是 0.否)", className);

getchar();

char choice;

scanf("%c", &choice);

// 是则创建文件夹

if (choice == '1')

{

char fileName[20];

sprintf(fileName, "data/%s", className);

createFolder(fileName);

printf("文件夹%s创建成功!\n", fileName);

}

else

{

// 否 返回输入班级名

goto Flag1;

}

}

else

{

// 班级存在则读取班级文件数据

printf("正在读取文件数据...\n");

readData(judgeClassName);

}

// 选择要进行的操作

int choice;

char studentId[20];

while (1)

{

printMenu();

printf("\t请选择您需要的操作:");

scanf("%d", &choice);

switch (choice)

{

case 1:

printf("添加学生打卡信息\n");

addStudentData(judgeClassName);

readData(judgeClassName); // 每次添加完成后重新读取数据

break;

case 2:

printf("按学号查询学生的打卡信息\n请输入学生学号:");

scanf("%s", studentId);

showDataById(judgeClassName, studentId);

break;

case 3:

printf("按学号删除学生打卡记录\n请输入学生学号:");

scanf("%s", studentId);

deleteDataById(judgeClassName, studentId);

readData(judgeClassName); // 每次删除完成后重新读取数据

break;

case 4:

printf("查询班级的打卡情况（缺卡人数及其缺卡次数）\n");

printTotalData();

break;

case 0:

printf("退出系统\n");

return 0;

default:

printf("输入错误!");

break;

}

}

}

**myfunctions.c:**

#include <stdio.h>

// 打卡系统用到的其它函数

/\*打印菜单\*/

void printMenu()

{

printf("\t\t★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★\n");

printf("\t\t★ --欢迎进入学生自习打卡管理系统-- ★\n");

printf("\t\t★ 1.添加学生打卡信息 ★\n");

printf("\t\t★ 2.按学号查询学生打卡信息 ★\n");

printf("\t\t★ 3.按学号删除学生打卡记录 ★\n");

printf("\t\t★ 4.本班级打卡信息统计 ★\n");

printf("\t\t★ 0.退出系统 ★\n");

printf("\t\t★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★\n");

}

/\*打印学生打卡信息\*/

void printData(struct clockData student)

{

printf("学号:%s 姓名:%s 班级:%s 打卡日期:%s\n",

student.id, student.name, student.className, student.clockInDate);

}

/\*打印班级学生所有打卡统计\*/

void printTotalData() {

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

{

printf("学号:%s 姓名:%s 打卡次数:%d\n",classClockTotal[i].id,classClockTotal[i].name,classClockTotal[i].clockCount);

}

}

/\*根据学生id查询学生打卡信息\*/

void showDataById(char \*fileName, char \*studentId)

{

char name[40];

sprintf(name, "%s/%s.txt", fileName, studentId);

// 若信息不存在

if (access(name, 0) == -1)

{

printf("该学号信息不存在!(按任意键继续)");

getchar();

}

else

{

FILE \*pFile;

pFile = fopen(name, "r");

if (pFile == NULL) perror("Error opening file");

else

{

int dataCount = 0;

char mystring[100];

while (fgets(mystring, 100, pFile) != NULL)

{

char \*p;

p = strtok(mystring, " ");

sscanf(p, "%s", cData.id);

p = strtok(NULL, " ");

sscanf(p, "%s", cData.name);

p = strtok(NULL, " ");

sscanf(p, "%s", cData.className);

p = strtok(NULL, " ");

sscanf(p, "%s", cData.clockInDate);

printData(cData);

dataCount++;

}

printf("统计：学生%s共打卡%d次\n",cData.name,dataCount);

fclose(pFile);

}

}

}

/\*根据学生id删除学生打卡信息\*/

void deleteDataById(char \*fileName, char \*studentId)

{

char name[40];

sprintf(name, "%s/%s.txt", fileName, studentId);

// 若信息不存在

if (access(name, 0) == -1)

{

printf("该学号信息不存在!(按任意键继续)");

getchar();

}

else

{

remove(name);

printf("删除学生打卡信息成功!(按任意键继续)");

getchar();

}

}

**fileoperate.c:**

#include <stdio.h>

#include <unistd.h>

#include <sys/stat.h>

#include <sys/dirent.h>

#include <dirent.h>

#include <memory.h>

#include "mystructs.h"

// 打卡系统用到的文件操作相关函数

/\*创建文件夹\*/

void createFolder(char \*folderName)

{

// 文件夹不存在则创建文件夹

if (access(folderName, 0) == -1)

{

mkdir(folderName, 0775);

}

}

/\*将数据写入文件\*/

int writeFile(char \*fileName, char \*data, int mode)

{

FILE \*file;

if (mode == 0)

{

file = fopen(fileName, "a+");

}

else

{

file = fopen(fileName, "w+");

}

if (file == NULL)

{

printf("文件打开错误!\n");

return 1;

}

fputs(data, file);

fclose(file);

return 0;

}

/\*读取文件数据并保存到打卡信息结构体\*/

void readData(char \*fileName) {

// 遍历数据文件目录，打开文件并加载数据到结构体studentsData

DIR \*dir;

struct dirent \*ptr;

dir = opendir(fileName);

FILE \*pFile;

classClockTotalCount = 0;

while ((ptr = readdir(dir)) != NULL)

{

if (strcmp(ptr->d\_name, ".") != 0 && strcmp(ptr->d\_name, "..") != 0)

{

char readFileName[40];

sprintf(readFileName, "./%s/%s", fileName, ptr->d\_name);

printf("正在读取%s\n", readFileName);

// 读取文件数据并保存到结构体

char mystring[100];

int dataCount = 0;

pFile = fopen(readFileName, "r");

if (pFile == NULL) perror("文件打开错误");

else {

while (fgets(mystring, 100, pFile) != NULL)

{

char \*p;

p = strtok(mystring, " ");

sscanf(p, "%s", cData.id);

p = strtok(NULL, " ");

sscanf(p, "%s", cData.name);

p = strtok(NULL, " ");

sscanf(p, "%s", cData.className);

p = strtok(NULL, " ");

sscanf(p, "%s", cData.clockInDate);

dataCount++;

}

fclose(pFile);

}

sscanf(cData.id, "%s", classClockTotal[classClockTotalCount].id);

sscanf(cData.name, "%s", classClockTotal[classClockTotalCount].name);

classClockTotal[classClockTotalCount].clockCount = dataCount;

classClockTotalCount++;

}

}

closedir(dir);

}

/\*添加学生打卡信息\*/

void addStudentData(char \*filePath) {

printf("格式：学号 姓名（注：姓名以最后一次打卡输入的为准） 班级 打卡日期\n请输入:");

struct clockData stu;

scanf("%s %s %s %s", stu.id, stu.name, stu.className, stu.clockInDate);

// 根据学生的班级生成文件名，打开数据文件并写入

char fileName[40];

sprintf(fileName, "%s/%s.txt", filePath, stu.id);

printf("%s", fileName);

// 拼接学生打卡信息

char data[80];

sprintf(data, "%s %s %s %s\n", stu.id, stu.name, stu.className, stu.clockInDate);

// 追加方式写入文件

writeFile(fileName, data, 0);

printf("添加成功!(按任意键继续)");

getchar();

}

**mystructs.h:**

// 本系统所用结构体

#ifndef TASK5\_HEA\_H // 防止头文件在同一编译单元被重复引用

#define TASK5\_HEA\_H

/\*学生打卡信息结构体\*/

struct clockData {

char id[20]; // 学号

char className[20]; // 班级

char name[20]; // 姓名

char clockInDate[30]; // 打卡日期 年.月.日 2020.12.21

} cData;

// 班级内学生打卡统计

struct ClassClockTotal{

char id[20]; // 学号

char name[20]; // 学生姓名

int clockCount; // 打卡次数

}classClockTotal[50];

int classClockTotalCount = 0; // 统计个数

#endif //TASK5\_HEA\_H

**参考文献**

1. 谭浩强.C程序设计（第五版）.北京：清华大学出版社，2018.