**实 验 报 告**

**课程名称** 数据结构

**实验项目**  实验三 学生成绩管理系统

**系 别\_\_\_ \_计算机学院 \_ \_\_\_\_\_\_**

**专 业\_\_\_\_计算机科学与技术\_ \_\_**

**班级/学号\_\_\_\_1503/2015011227 \_\_**

**学生姓名 \_\_\_\_\_\_\_徐蕾 \_\_\_\_\_\_\_\_\_\_ \_**

**实验日期** \_ 2017年3月27日

**成 绩** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_

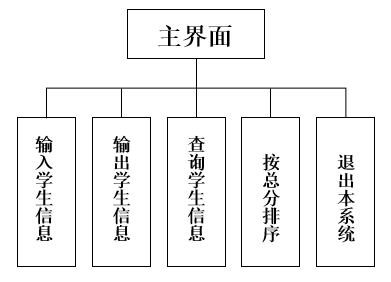
**指导教师** 黄改娟

1. 需求分析

系统主要管理学生信息及成绩信息排序等事项。根据需要可以查询学生的信息，主要功能包括如下。

1. 使用中文菜单，界面设计和用户输入输出要人性化。
2. 将学生信息保存早文本文档中，具体对学生信息进行插入删除操作时，将保存在文本文档中的学生信息提取出来，保存在自己定义的数据结构中，然后再对该数据结构进行操作，所有操作完成，或者在相应的命令完成后，再将学生信息保存到文本中。
3. 具有数据输入功能，输入的数据能最终保存在文件中。
4. 排序功能，根据自己设计的数据结构，设计排序算法。
5. 具有多种查询（如按学号查询、按姓名查询等）及输出功能。
6. 方案设计

总体功能模块结构如图所示



在主界面中包括1. 输入学生信息并保存到文件；2. 读取文件并输出学生信息；3. 按学号查询；4. 按姓名查询；5. 按总分排序。之所以设计这一主界面，一是因为能使用户对程序操作的流程更加清晰简明，二是保证了用户同时只能对一个文件进行操作系统的要求，保证了系统不会打开文件紊乱或者出现致命的错误。

1. 系统实现

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct student

{

char num[10];/\*学号\*/

char name[20];/\*姓名\*/

int fgrade;/\*课程1成绩\*/

int cgrade;/\*课程2成绩\*/

int egrade;/\*课程3成绩\*/

int total;/\*总分\*/

}st[100];

void menu(); //菜单

void input(); //输入学生信息并保存

void save(int m); //保存文件函数

void save(); //保存文件函数

void display(); //浏览学生信息函数

void search(); //查询函数

void search\_num(); //按学号查找

void search\_name(); //按姓名查找

void sort\_total(); //按总分排序

int load();/\*定义各函数\*/

int m;

int main()

{

menu();

int n=0,flag;

char a;

do

{

printf("请选择你需要操作的步骤(1--4):\n");

scanf("%d",&n);

if(n>=0&&n<=4)

{

flag=1;

break;

}

else

{

flag=0;

printf("输入有误，请重新选择");

}

}

while(flag==0);

while(flag==1)

{

switch(n)

{

case 1:printf("\n ◆◆◆输入学生信息◆◆◆\n");

printf("\n");

input();

break;

case 2:printf("\n ◆◆◆浏览学生信息◆◆◆\n");

printf("\n");

display();

break;

case 3:printf("\n ◆◆◆按总分排序◆◆◆\n");

printf("\n");

sort\_total();

break;

case 4:printf("\n ◆◆◆查询学生信息◆◆◆\n");

printf("\n");

search();

break;

case 0:exit(0);

break;

default:break;

}

getchar();

printf("\n");

printf("是否继续运行（y or n）:\n");

scanf("%c",&a);

if(a=='y')

{

flag=1;

system("cls"); /\*清屏\*/

menu(); /\*调用菜单函数\*/

printf("请再次选择你需要操作的步骤(1--4):\n");

scanf("%d",&n);

printf("\n");

}

else

exit(0);

}

}

void menu() /\*菜单函数\*/

{

printf(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf(" 1.录入学生信息\n");

printf(" 2.浏览学生信息\n");

printf(" 3.按总分排序\n");

printf(" 4.查询学生信息\n");

printf(" 0.退出\n");

printf(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n");

printf("\n");

}

void input() /\*输入学生信息并保存\*/

{

int i;

printf("请输入需要创建信息的学生人数(1--10):\n");

scanf("%d",&m);

for(i=0;i<m;i++)

{

printf("请输入学号:");

scanf("%s",st[i].num);

printf("请输入姓名:");

scanf("%s",st[i].name);

printf("请输入课程1成绩:");

scanf("%d",&st[i].fgrade);

printf("请输入课程2成绩:");

scanf("%d",&st[i].cgrade);

printf("请输入课程3成绩:");

scanf("%d",&st[i].egrade);

st[i].total=st[i].fgrade+st[i].cgrade+st[i].egrade;

/\*信息输入已经完成\*/

}

save(m);

return;/\*返回主界面\*/

}

void display()/\*浏览学生信息\*/

{

int i;

m=load();

printf("\n学号\t姓名\t课程1成绩\t课程2成绩\t课程3成绩\t总分\n");/\*输出学生信息\*/

for(i=0;i<m;i++)

{

printf("\n%s\t%s\t%d\t\t%d\t\t%d\t\t%d\n",st[i].num,st[i].name,st[i].fgrade,st[i].cgrade,st[i].egrade,st[i].total);

}

}

void sort\_total() /\*按总分排序\*/

{

int i,j;

struct student temp; /\*声明结构体变量\*/

printf("\t学号\t姓名\t课程1成绩\t课程2成绩\t课程3成绩\t总分\n");

for(i=0;i<m;i++)/\*对数据进行排序\*/

{

for(j=i+1;j<m;j++)

if(st[i].total<st[j].total)

{

temp=st[i];

st[i]= st[j];

st[j]=temp;

}

printf("\n\t%s\t%s\t%d\t\t%d\t\t%d\t\t%d\n",st[i].num,st[i].name,st[i].fgrade,st[i].cgrade,st[i].egrade,st[i].total);

}

}

void search() /\*查询函数\*/

{

int k; /\*定义局部变量\*/

m=load();

printf("\n按学号查询请按1，\t按姓名查询请按2\n");

printf("\n请输入查询方式:");

scanf("%d",&k);

switch(k)/\*查询方式的选择\*/

{

case 1:search\_num();break;/\*按学号查询\*/

case 2:search\_name();break;/\*按姓名查询\*/

}

printf("\nPress any key to enter menu......");

getchar();

return; /\*返回主界面\*/

}

void search\_num()/\*(1)按学号查找\*/

{

int i;

char temp[30];

printf("\n 请输入要查找的学号:");

scanf("%s",temp);

for( i=0;i<m;i++)

{

if(strcmp(temp,st[i].num)==0)

{

printf("\n学号\t姓名\t课程1成绩\t课程2成绩\t课程3成绩\t总分\n");

printf("\n%s\t%s\t%d\t\t%d\t\t%d\t\t%d\n",st[i].num,st[i].name,st[i].fgrade,st[i].cgrade,st[i].egrade,st[i].total);

}

}

}

void search\_name()/\*(2)按姓名查找\*/

{

int i;

char temp[30];

printf("\n 请输入要查找的姓名:");

scanf("%s",temp);

for( i=0;i<m;i++)

{

if(strcmp(temp,st[i].name)==0)

{

printf("\n学号\t姓名\t课程1成绩\t课程2成绩\t课程3成绩\t总分\n");

printf("\n%s\t%s\t%d\t\t%d\t\t%d\t\t%d\n",st[i].num,st[i].name,st[i].fgrade,st[i].cgrade,st[i].egrade,st[i].total);

}

}

}

int load() /\*导入函数\*/

{

FILE\*fp;

int n=0;

if((fp=fopen("student\_list","rb"))==NULL)

{

printf("cannot open file\n");

exit(0);

}

else

{

do

{

fread(&st[n],sizeof(struct student),1,fp);

n++;

}

while(feof(fp)==0);

}

fclose(fp);

return(n-1);

}

void save(int m)/\*保存文件函数\*/

{

int i;

FILE\*fp;

if((fp=fopen("student\_list","wb"))==NULL)/\*创建文件并判断是否能打开\*/

{

printf("cannot open file\n");

exit(0);

}

for(i=0;i<m;i++)/\*将内存中学生的信息输出到磁盘文件中去\*/

if(fwrite(&st[i],sizeof(struct student),1,fp)!=1)

printf("file write error\n");

fclose(fp);

}

1. 结果测试



