****

**软件工程作业**

**——可行性分析**

**姓 名** 毛佳妮、倪毓蔚、羊梦佳、邓春花

**学 号** 201511060206 201511060201

201511060202、201511060208

**学 院** 理学院

**专 业** 应用统计152

**一、源代码**

**代码如下：**

#include<stdio.h>

#include<malloc.h>//动态存储分配函数头文件

#include<math.h>//包含数学函数的文件

#include<string.h>//一个和字符串处理相关的头文件

#include<process.h>//包含用于和宏指令的作用声明与螺纹和过程一起使用的C标头文件

#define ERROR 0 //宏定义

#define OK 1

#define over -2

#define ListInitSize 20

#define ListIncrement 5

#define Namelen 8

#define Majorlen 20

#define INIT\_N 3

typedef int Status; //自定义类型语句 Status i 等价于 int i

typedef char StatusC;

typedef struct{

StatusC number[10]; //学号：081330385

StatusC name[Namelen+1]; //姓名：郭靖

StatusC gender; //性别：男

StatusC telephone[12]; //电话：15890377853

StatusC qqNumber[13]; //QQ:485284320

StatusC major[Majorlen+1];//专业：计算机科学与技术

}Student;//类型为学生

typedef struct{

Student \*elem; //存储空间的基址

Status length; //当前长度

Status listsize; //当前分配的存储容量（以sizeof（Studnet）为单位）

}Sqlist;

//信息初始化，构造一个空的线性表

Status InitList(Sqlist &l){

l.elem=(Student \*)malloc(ListInitSize\*sizeof(Student));

//申请LIST\_INIT\_SIZE个大小为整型（Student）字节的空间，

//把空间指针给L.elem

if(!l.elem)exit(over);

//判断l中的数据成员elem是否为0,为0即执行if下面语句

l.length=0;

//赋值

l.listsize=ListInitSize;

return OK;

}

//插入记录

Status ListInsert(Sqlist &l,Status i,Student s){

Student \*newbase;

if(i<1||i>l.length+1)

return ERROR;//i值不合法

if(l.length>=l.listsize)//当前存储空间已经满了，增加分配空间

{

newbase=(Student\*)realloc(l.elem,(

ListInitSize+ListIncrement)\*(sizeof(Student)));

/\*

先释放原来L.elem所指内存区域，并按照

(LIST\_INIT\_SIZE+ListIncrement)\*sizeof(Student)

的大小重新分配空间其中ListIncrement为20，

同时将原有数据从头到尾拷贝到新分配的内存区域，

并返回该内存区域的首地址。即重新分配存储器块。

\*/

if(!newbase)exit(over);

// 判断newbase是否为0,为0即执行if下面语句

l.elem=newbase;

//赋值

l.listsize+=ListIncrement;

}

for(Status j=l.length;j>=i;j--)

l.elem[j]=l.elem[j-1];

//赋值

l.elem[i-1]=s;

l.length++;

return OK;

}

//打印输出线性表中的信息

void print(Sqlist &l){

if(l.length==0)

printf("空表，无任何记录！\n\n");

printf("学号 姓名 性别 手机号 QQ 专业\n");

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

for(Status i=0;i<l.length;i++){

printf("%-10s %-8s",l.elem[i].number,l.elem[i].name);

//输出学好 姓名

if(l.elem[i].gender=='m')printf("男");

//判断语句

else printf("女");

printf("%14s %9s %10s",l.elem[i].telephone,l.elem[i].qqNumber,l.elem[i].major);

//输出电话号码 qq号码 专业

printf("\n");

}

printf("------------------------------\n");

}

//打印输出线性表中指定学生的信息

void print\_single(Student s){

printf("学号 姓名 性别 手机号 QQ 专业\n");

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

printf("%-10s %-8s",s.number,s.name);

//输出学好 姓名

if(s.gender=='m')printf("男");

else printf("女");

printf("%14s %9s %10s",s.telephone,s.qqNumber,s.major);

//输出电话号码 qq号码 专业

printf("\n");

printf("------------------------------\n");

}

//接受键盘输入信息

void ScanIn(Sqlist &l){

StatusC a[2];

Status i;

printf("\t 您选择插入一条学生记录\n");

do{

Student s;

printf("\t 请输入学号：");

scanf("%s",s.number);

printf("\t 请输入姓名(<=%d个字符)：",Namelen);

scanf("%s",s.name);

printf("\t 请输入电话号码：");

scanf("%s",s.telephone);

printf("\t 请输入QQ号：");

scanf("%s",s.qqNumber);

printf("\t 请输入专业：");

scanf("%s",s.major);

printf("\t 请输入性别：(m:男，f:女)");

scanf("%\*c%c",&s.gender);

printf("\t 请输入你要插入的位置(1<=i<=%d)：",l.length+1);

scanf("%d",&i);

ListInsert(l,i,s);

printf("\t 您想继续插入记录吗？(y/n)\t");

scanf("%s",a);

}while(strcmp(a,"y")==0||strcmp(a,"Y")==0);

//调用strcmp函数

}

//删除通讯中第i条记录

Status ListDelete(Sqlist &l,int i,Student s){

Status j;

if(i<1||i>l.length)return ERROR;

//判断

else{

s=l.elem[i-1];

for(j=i;j<=l.length;j++)l.elem[j-1]=l.elem[j];

l.length--;

return OK;

}

}

//按照升序排列

void SortorderAscend(Sqlist &l){

system("cls");//清空前面的信息之输出下面信息

Student s;

for(Status i=0;i<l.length-1;i++)

for(Status j=0;j<l.length-i-1;j++){

//循环比较大小 ，排列顺序

if(strcmp(l.elem[j].number,l.elem[j+1].number)>0){

//strcmp() 以二进制的方式进行比较，不会考虑多字节或宽字节字符；

//用来比较字符串（区分大小写）

s=l.elem[j];

l.elem[j]=l.elem[j+1];

l.elem[j+1]=s;

}

}

printf("\*\*\*\*\*按照升序排序成功！\*\*\*\*\*\*\n\n");

}

//判断姓名是否相同

Status EqualName(Student s1,Student s2){

if(strcmp(s1.name,s2.name))return ERROR;

//比较两个字符串是否相同

else return OK;

}

//定位某个元素

int LocateElem(Sqlist l,Student s,Status(\*compare)(Student,Student)) {

Status i=1;

Student \*p=l.elem;

while(i<=l.length&&!compare(\*p++,s))i++;

if(i<=l.length)

return i;

//定位元素的位置

else

return 0;

}

//按照姓名查找

void searchName(Sqlist l,Status(\*equal)(Student,Student)){

system("cls");//清空前面的信息之输出下面信息

Student s;

Status j;

StatusC a[2];

do{

printf("\t 请输入你要查找的学生的姓名：");

scanf("%s",s.name);

j=LocateElem(l,s,equal);

if(!j)printf("没有查找到你所要查找的学生记录！\n");

else print\_single(l.elem[j-1]);

printf("\t 您想继续查找吗？(y/n)\t");

scanf("%s",a);

}while(strcmp(a,"y")==0||strcmp(a,"Y")==0);

}

//修改姓名

void ModifyName(Sqlist &l,Status(\*equal)(Student,Student)){

system("cls");//清空前面的信息之输出下面信息

Student s;

Status j;

StatusC number\_new[10],name\_new[Namelen+1],gender\_new,

telephone\_new[12],qqNumber\_new[13],major\_new[Majorlen+1];

StatusC a[2];

do{

printf("\t 请输入你要更改的学生的姓名：");

scanf("%s",s.name);

j=LocateElem(l,s,equal);

if(!j)printf("没有查找到你所要查找的学生记录！\n");

else {

print\_single(l.elem[j-1]);

Status isOrNo=1;

printf("\n修改通讯录：1 学号，2 姓名，3 性别，4 电话，5 QQ号，6 专业， 0 取消\n") ;

while(isOrNo){

printf("--------------------------\n");

printf("请选择需要修改的属性序号：");

scanf("%d",&isOrNo);

switch(isOrNo){

case 1:

printf("请输入更改后的学号：");

scanf("%s",number\_new);

strcpy(l.elem[j-1].number,number\_new);

break;

case 2:

printf("请输入更改后的姓名：");

scanf("%s",name\_new);

strcpy(l.elem[j-1].name,name\_new);

break;

case 3:

printf("请输入更改后的性别：");

scanf("%\*c%c",&l.elem[j-1].gender);

break;

case 4:

printf("请输入更改后的电话：");

scanf("%s",telephone\_new);

strcpy(l.elem[j-1].telephone,telephone\_new);

break;

case 5:

printf("请输入更改后的QQ号：");

scanf("%s",qqNumber\_new);

strcpy(l.elem[j-1].qqNumber,qqNumber\_new);

break;

case 6:

printf("请输入更改后的专业：");

scanf("%s",major\_new);

strcpy(l.elem[j-1].major,major\_new);

break;

}

}

printf("\n\*\*\*\*更改后该条记录变为\*\*\*\*\n\n");

print\_single(l.elem[j-1]);

}

printf("您想继续修改吗？(y/n)\t");

scanf("%s",a);

}while(strcmp(a,"y")==0||strcmp(a,"Y")==0);

}

//主菜单

void menu(){

system("cls");//清空前面的信息之输出下面信息

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

printf("\t\*\*欢迎使用学生通讯录管理系统\*\*\n");

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

printf("\t\*\*请选择如下操作，输入序号\*\*\n");

printf("\t 0：退出系统\n");

printf("\t 1：将初始学生记录插入线性表\n");

printf("\t 2：显示线性表中所有内容\n");

printf("\t 3：向线性表中增加一条记录\n");

printf("\t 4：从线性表中删除一条记录\n");

printf("\t 5：按照姓名修改一条记录\n");

printf("\t 6：按照姓名查找一条记录\n");

printf("\t 7：按照学号升序排列通讯录\n");

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

printf("\t 您想进行什么操作，请选择\n");

}

//主函数

int main(){

Sqlist l;

Status i,select; //select 变量标记用户的选择

if(InitList(l)!=OK)printf("\n\t线性表初始化失败！\n");

Student s[INIT\_N]={{"081404327","马东",'m',"13729947569","87384747" ,"机械工程"},

{"081404527","李东",'m',"13729985469","84786747" ,"软件工程"},

{"081435797","马西",'f',"12329947569","84735777" ,"通讯工程"}};

menu();

scanf("%d",&select);

Student \*s2;

int k;

while(select){

switch(select){

case 1: //初始化数据

system("cls");//清空前面的信息之输出下面信息

for(i=0;i<INIT\_N;i++){

if(ListInsert(l,i+1,s[i])!=OK)

printf("\t初始化数据失败！\n");

}

if(i==INIT\_N)printf("\t初始化数据成功\n\n");

print(l);

printf("\n输入0：退出，输入11：返回主菜单\t");

break;

case 2://显示线性表中的记录

system("cls");//清空前面的信息之输出下面信息

print(l);

printf("\n输入0：退出，输入11：返回主菜单\t");

break;

case 3://接受键盘输入，并把记录插入到线性表中

system("cls");//清空前面的信息之输出下面信息

ScanIn(l);

system("cls");//清空前面的信息之输出下面信息

print(l);

printf("\n输入0：退出，输入11：返回主菜单\t");

break;

case 4://删除通讯录中某条信息

system("cls");//清空前面的信息之输出下面信息

s2=l.elem;

print(l);

printf("您想删除第几条通讯录1~%d:\n",l.length);

scanf("%d",&k);

Student de\_s;

system("cls");//清空前面的信息之输出下面信息

if(ListDelete(l,k,de\_s)==OK){

print(l);

printf("\t%-6s 已经删除成功！\n",de\_s.name);

}

printf("\n输入0：退出，输入11：返回主菜单\t");

break;

case 5://按照姓名修改一条记录

ModifyName(l,EqualName);

printf("\n输入0：退出，输入11：返回主菜单\t");

break;

case 6://按照姓名查找记录

searchName(l,EqualName);

printf("\n输入0：退出，输入11：返回主菜单\t");

break;

case 7://按照学好升序排列

system("cls");//清空前面的信息之输出下面信息

SortorderAscend(l);

print(l);

printf("\n输入0：退出，输入11：返回主菜单\t");

break;

case 11://返回主菜单

menu();

break;

}

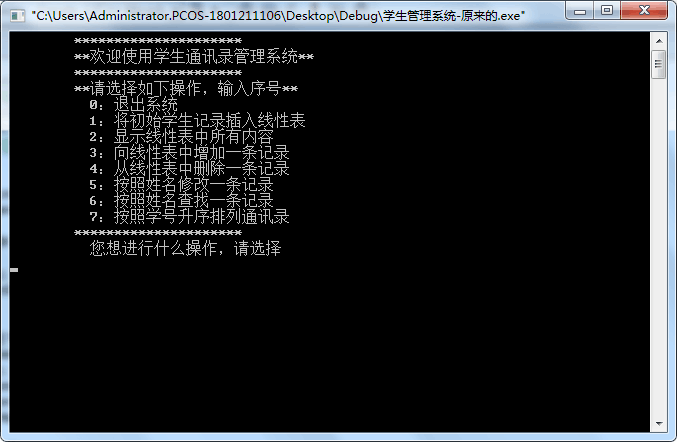
scanf("%d",&select);

}

return 0;

}

**运行结果如下：**



**由于已经获取了可以运行的源代码，我们将在源代码的基础上进行修改，增加有关成绩和选课的功能，该项目具有较高的可行性。**

**二、现阶段功能及改进**

**现阶段功能：**

1. 增加学生信息，包括学号、性别、姓名、专业、电话号码、qq号，插入位置。
2. 按照姓名修改学生信息。
3. 删除学生信息记录
4. 按姓名查找学生信息
5. 读取程序数据
6. 读取线性表中数据
7. 排序

**其程序思维导图如下：**

**学生管理系统**

4

3

1

0

7

6

5

2

排序

查找（按姓名）

修改（按姓名）

删除记录

增加记录

显示线性表中内容

初始线性表

退出系统

读取程序中数据 读取线性表中数据 学号 通讯 学号 继续查找 返回主菜单

返回主菜单 返回主菜单 性别 返回主菜 性别 返回主菜单

姓名 姓名

专业 专业

电话号码 电话号码

qq号 qq号

插入位置 插入位置

继续插入 继续插入

返回主菜单 返回主菜单

**改进：**

**在上述功能的基础上，我们增加了如下功能。**

1、登录功能，输入正确密码即可进入操作界面

2、增加课程，课程包括课程名、学分、上课时间、任课老师

3、删除课程

4、修改课程

5、给学生选的课录入成绩

5、统计功能，包括：统计各门课程的平均分

每个学生个人的平均分

各门成绩排序

**其程序思维导图如下：**

**学生管理系统**

退出系统

0

初始线性表

1

读取程序中数据

返回主菜单

2

显示线性表中内容

读取线性表中数据返回主菜单

3

增加记录

学号

性别

姓名

专业

电话号码

qq号

插入位置

继续插入

返回主菜单

4

删除记录

通讯

返回主菜单

5

修改（按姓名）

学号

性别

姓名

专业

电话号码

qq号

插入位置

继续插入

返回主菜单

6

删除记录

继续查找

返回主菜单

7

排序

返回主菜单

8

录入、修改、删除成绩

分数

绩点

9

增加课程

学分

上课时间

任课老师

10

删除课程

学分

上课时间

任课老师

11

修改课程

学分

上课时间

任课老师

12

统计

课程平均分

个人平均分

成绩排名

**三、系统流程图**

输入正确密码登录

增加学生课程

修改学生课程

删除学生课程

增加成绩信息

统计成绩信息

统计各课程平均分

学生个人平均分

学生按各成绩排序

学生成绩表

学生排名表

学生课程表

退出系统

**四、数据流图**

选课信息表

成绩信息表

选课信息

录入成绩

管理员

添加学生信息

学生信息表

查询学生信息

修改学生信息

删除学生信息

学生

学号

学生信息

学生信息

学生信息

学生信息

成绩录入

选课信息

选课查询

成绩信息

登录

密码