## 1 需 求 分 析

## 2.1 编写目的

## 培养学生用学到的书本知识解决实际问题的能力；培养实际工作所需要的动手能力；培养学生以科学理论和工程上能力的技术，规范地开发大型、复杂、高质量的应用软件和系统软件具有关键性作用；通过课程设计的实践，学生可以在程序设计方法、上机操作等基本技能和科学作风方面受到比较系统和严格的训练。

## 1.2 需求概述

设计一个学生通信录管理系统 使之能提供以下功能：

1. 创建学生信息功能
2. 学生信息浏览功能
3. 学生信息查询功能
4. 学生信息删除功能
5. 学生信息修改功能
6. 学生信息插入功能

## 1.3 需求说明

# （1）学生通信录包括 学号、姓名、系别、邮箱，联系方式。

（2）录入的学生通讯信息要求并可以对其进行浏览、查询、修改、删除 等基本操作。

（3）学生学籍信息的显示要求有一定的规范格式。

（4）对学生学籍信息应能够按学号的方式进行查询

（5）对学生学籍信息的修改应逐个地进行，一个同学通信录信息的更改不应影响其他的职工记录。

（6）所设计的系统应以菜单方式工作，应为用户提供清晰的使用提示，依据用户的选择来

进行各种处理，并要求在此过程中能够尽可能地兼容用户使用过程中的异常情况。

## 2 总 体 设 计

# 功能划分

# 该系统可以按功能进行模块划分，如图 7-1 所示。

学生通讯录管理系统

学生信息查询管理

学生信息修改管理

修改生及其个人

浏览学生及其个人

查找人员及其个人信息

插入学生及其个人

录入学生及其个人信息

删除学生及其个人信息

其中：

（1）菜单选择模块完成用户命令的接收，此模块也是学生通讯录管理系统的入口，用户所要进行的各种操作均需在此模块中进行选择，并进而调用其他模块实现相应的功能。

（2）信息输入模块完成学生通讯录信息的输入功能，输入信息包括学号、姓名、院系、邮箱、联系方式等。

（3）信息浏览模块完成已录入学生通讯录信息的显示。

（4）信息查询模块完成学生通讯录的查询，查询时对应有按学号查询方式。

（5）信息修改模块完成学学生通讯录信息的修改功能。

（6）信息删除模块完成学生通讯录信息的删除功能。

（7）信息插入模块完成学生通讯录信息的插入功能。

## 

## 2．数据结构

本系统中主要的数据结构就是学生学籍的信息，包含学号、姓名、院系、邮箱、联系方式等。在处理过程中各项可以作为一个学生的不同属性来进行 处理。

## 3．程序流程

# 系统的执行应从功能菜单的选择开始，依据用户的选择来进行后续的处理，直到用户 选择退出系统为止，其间应对用户的选择做出判断及异常处理。系统的流程图如图 7-2 所 示。

开始

菜单选择

输入

浏览

查询

修改

删除

退出

信息删除

信息修改

信息查询

信息浏览

信息输入

结束

**图 7-2 程序流程图**

## 3 详 细 设 计

## 3.1 编写目的

根据总体设计说明书，在总体设计的基础之上，对系统进行详细设计，以便下一步进 行程序编码工作。

## 3.2 详细设计

1．数据结构

学生通信录：

typedef struct Node

{

char num[10],

name[20],

xibie[20],

Emil[20],

phone[20]; //num是学号 name是名字

struct Node \*next;

} Node;

各个功能模块的处理流程

对应于总体设计时的系统模块图，各个功能模块的处理流程如下：

（1）信息输入模块

Do{ 读入学生信息;

提示是否继续输入;

}

while(继续输入的判断条件)

（2）信息修改模块

If（没有学生）

输出“没有学生”

Else do{

提示：“请输入你要修改学生的学号”

输入学号

While()

{

If（信息一样）

修改学生通讯录信息

}

}while（判断是否继续的条件）

（3）信息查询模块

If（没有学生）

输出“没有学生”

Else do{

提醒输入学生学号

查找学生信息

显示学生信息

If（判断学生是否存在）

输出 学生不存在

提示 “继续录入请按任意键---退出录入请按N”

}while(判断是否继续的条件)

（4）信息删除模块

Do{

If(学生不存在)

输出“没有学生数据可以删除”

Else

提示输入要删除的学生学号

For（查找相同学生学号）

输出要删除的学生的数据

If（只有一个学生）

直接删除

Else

该学生开始，后面学生的数据依次覆盖前面学生的数据

提示数据已删除

提示 “继续录入请按任意键---退出录入请按N”

}while（判断是否继续的条件）；

1. 信息插入模块

限制输入的范围

Do{

、 提示要插入的学生数据

判断是否存在

提示 “继续录入请按任意键---退出录入请按N”

}while（判断是否继续的条件）；

（6）信息浏览模块

If（判断没有学生）

输出“没有数据，请先输入数据”

Else

输出“打印顺序为：学号、姓名、性别、年龄、系别、专业、班级、籍贯：”

For（依次寻找学生数组学生）

依次打印学生学籍信息

# （7）菜单选择模块

给出信息提示;

绘制菜单(包含录入数据､ 修改数据､ 查找数据､删除数据､ 插入数据､ 浏览数据､ 退出系统);

提示菜单选择并读入到变量;

返回变量的值;

## 4 参 考 程 序

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

typedef struct Node

{

char num[10], name[20], xibie[20],Emil[20],phone[20]; //num是学号 name是名字

struct Node \*next;

} Node;

typedef struct Node \*LinkList; /\* 定义LinkList \*/

int i=0; //定义一个全局变量，用于记录有多少学生

bool InitList(LinkList \*L)

{

\*L=(LinkList)malloc(sizeof(Node)); /\* 产生头结点,并使L指向此头结点 \*/

if(!(\*L)) /\* 存储分配失败 \*/

return false;

(\*L)->next=NULL; /\* 指针域为空 \*/

return true;

}

//录入数据

void Entry(LinkList L) //添加学生成员

{

LinkList p= L;

char ch;

do

{

LinkList NewNode=(LinkList)malloc(sizeof(Node));

i++;// i代表学生人数

printf("请输入学生的学号\n");

scanf("%s",NewNode->num);

printf("请输入学生的姓名\n");

scanf("%s",NewNode->name);

printf("请输入学生的系别\n");

scanf("%s",NewNode->xibie);

printf("请输入学生的邮箱：\n");

scanf("%s",NewNode->Emil);

printf("请输入学生的电话：\n");

scanf("%s",NewNode->phone);

NewNode->next=p->next;

p->next=NewNode;

getchar();

printf("-----继续录入请按任意键---退出录入请按N-----\n");

scanf("%c",&ch);

}

while(ch!='N'&&ch!='n'); //判断是否继续的条件

}

//查找数据

void Find(LinkList L)

{

int t; //xuehaowei0表示没有找到，1为找到相关学生

char ch,x[20];

if(i==0) //如果i=0，则说明没有数据

{

printf("没有学生：\n");

}

else do

{

int xuehao=0;

printf("\t按学号查找：\n");

printf("\t请输入学生学号");

scanf("%s",&x);

getchar();

LinkList p=L->next;

while(p)

{

if(!strcmp(p->num,x)) /\* 找到这样的数据元素 \*/

{

printf("所查学生的学号为：%s\n",p->num);

printf(" 姓名为：%s\n",p->name);

printf(" 系别为：%s\n",p->xibie);

printf(" 邮箱为：%s\n",p->Emil);

printf(" 电话为：%s\n",p->phone);

xuehao=1; //找到一个学生则变1

break;

}

p=p->next;

}

if(xuehao==0) //为0的时候说明没找到

printf("该学生不存在!\n");

printf("-----继续录入请按任意键---退出录入请按N-----\n");

scanf("%c",&ch);

}

while(ch!='N'&&ch!='n'); //判断是否继续的条件

}

void Change(LinkList L) //修改数据

{

long x; //n是输入的学号，q是修改后的学号

int a,b=0,c,ch,N;

char w[20],h[20],T[20],q[20],n[20]; //q是修改后学号 w是修改后的系别 h是修改后的邮箱 T是修改后的电话

if(i==0) //如果i=0，说明没有数据

{

printf("没有学生成员\n");

}

else do

{

printf("请输入你要修改学生的学号：");

scanf("%s",&n);

getchar();

LinkList p=L->next;

while(p)

{

if(!strcmp(n,p->num))

{

printf("1:修改学生的学号：\n");

printf("2:修改学生的系别：\n");

printf("3:修改学生的邮箱：\n");

printf("4:修改学生的电话：\n");

printf("请输入选项：");

scanf("%d",&c);

switch(c)

{

case 1:

printf("该学生学号为：%s\n",p->num);

printf("修改为：\n");

scanf("%s",q);

strcpy(p->num,q);

break;

case 2:

printf("该学生的系别为：%s\n",p->xibie);

printf("修改为：\n");

scanf("%s",w);

strcpy(p->xibie,w);

break;

case 3:

printf("该学生的邮箱为：%s\n",p->Emil);

printf("修改为：\n");

scanf("%s",h);

strcpy(p->Emil,h);

break;

case 4:

printf("该学生的电话为：%s\n",p->phone);

printf("修改为：\n");

scanf("%s",T);

strcpy(p->phone,T);

break;

default :

printf("输入错误！\n");

}

}

p=p->next;

}

printf("-----继续录入请按任意键---退出录入请按N-----\n");

getchar();

ch=getchar();

}

while(ch!='N'&&ch!='n'); //判断是否继续的条件

}

//删除数据

void Del(LinkList L)

{

char ch;

long t,j,s,q=0;

char n[20];

do

{

int x=0;

if(i==0)

printf("没有学生数据可以删除！\n");

else

{

printf("请输入要删除学生的学号：");

scanf("%s",&n);

LinkList p=L;

LinkList q;

while(p->next)

{

if(!strcmp(p->next->num,n)) /\* 找到这样的数据元素 \*/

{

x=1;

printf("您将要删除学生的数据如下：\n");

printf("%s %s %s %s %s \n",p->next->num,p->next->name,p->next->xibie,p->next->Emil,p->next->phone);

if(i==1) //如果只有一个学生，则直接删除即可

{

i--;

L->next=NULL;

printf("1数据已删除！\n");

}

else //如果有多个学生，则从该学生开始，后面学生的数据依次覆盖前面学生的数据

{

i--;

q=p->next;

p->next=q->next;

free(q);

printf("2数据已删除！\n");

}

break;

}

p=p->next;

}

if(x==0)

{

printf("没有该同学\n");

}

}

getchar();

printf("-----继续录入请按任意键---退出录入请按N-----\n");

scanf("%c",&ch);

}

while(ch!='N'&&ch!='n'); //判断是否继续的条件

}

//浏览数据

void Show(LinkList L)

{

long d;

if(i==0)

printf("没有数据，请先输入数据：\n");

else

{

LinkList p=L->next;

printf("打印顺序为：\n学号、 姓名、 系别、 邮箱、 电话：\n");

while(p)

{

printf("%s\t%s\t%s\t%s\t%s\t\n",p->num,p->name,p->xibie,p->Emil,p->phone);

p=p->next;

}

}

}

bool ListInsert(LinkList L)

{

int j=1,k;

printf("请输入要插入的位置:");

scanf("%d",&k);

if(k<=0)

{

printf("不能小于等于0\n");

return false;

}

if(k>i+1)

{

printf("当前学生只有 %d 人小于%d 人\n",i,k);

return false;

}

LinkList p;

p = L;

LinkList NewNode=(LinkList)malloc(sizeof(Node));

printf("请输入学生的学号\n");

scanf("%s",NewNode->num);

printf("请输入学生的姓名\n");

scanf("%s",NewNode->name);

printf("请输入学生的系别\n");

scanf("%s",NewNode->xibie);

printf("请输入学生的邮箱：\n");

scanf("%s",NewNode->Emil);

printf("请输入学生的电话：\n");

scanf("%s",NewNode->phone);

while (p && j < i) /\* 寻找第i个结点 \*/

{

p = p->next;

++j;

}

if (!p || j > i)

return false; /\* 第i个元素不存在 \*/

NewNode->next = p->next; /\* 将p的后继结点赋值给s的后继 \*/

p->next = NewNode; /\* 将s赋值给p的后继 \*/

i++;// i代表学生人数

return true;

}

//主函数 1.学生通信录管理系统（包括创建，查询，插入，删除，修改，查看）

int main()

{

LinkList L;

int w;

printf("您所使用的是南阳理工学院--------------------------\n");

printf("---------------计算机与信息工程学院\n");

printf("++++++++++++学生通信录管理系统++++++++++++++++++++++++++\n");

printf("\n");

printf("\n");

printf("++++++++++++++++++++为了能有效保存您的数据，请根据提示操作++++++++\n");

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

do

{

printf("\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*【主菜单】\*\n");

printf("\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~+[1]:录入数据：\n");

printf("\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~+[2]:修改数据：\n");

printf("\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~+[3]:查找数据：\n");

printf("\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~+[4]:删除数据：\n");

printf("\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~+[5]:插入数据：\n");

printf("\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~+[6]:浏览数据：\n");

printf("\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~+[0]:退出系统：\n");

printf("请输入选项：");

scanf("%d",&w);

switch(w)

{

case 1:

InitList(&L);

Entry(L);

break;

case 2:

Change(L);

break;

case 3:

Find(L);

break;

case 4:

Del(L);

break;

case 5:

if(ListInsert(L))

{

printf("插入成功！\n");

};

break;

case 6:

Show(L);

break;

case 0:

w=0;

break;

default:

printf("输入错误！\n");

break;

}

}

while(w!=0);

return 0;

}