通信与信息工程学院

《数据结构》实验报告

|  |  |
| --- | --- |
| 班 级： | 电信1702 |
| 姓 名： | 秦伟钦 |
| 学 号： | 17407110815 |
| 指导教师： | 刘晓佩 |
| 实验时间： |  |

通信与信息工程学院

二〇一九年

**目录**

实验2 线性表的基本操作(2)-------------------------------------------------------------------------------------------页码1

**完成所有实验报告后，按照上述格式形成目录。**

**实验二 线性表的基本操作（2）**

## 一、实验目的：

1、掌握线性表的链式存储结构定义；

2、掌握线性表的链式存储结构的基本操作，如建立、查找、插入和删除等。

## 二、实验环境

1. 装有Visual C＋＋6.0的计算机。

2. 本次实验共计2学时。

## 三、实验内容：

定义一个包含学生信息（学号，姓名，成绩）的链表，使其具有如下功能：

(1) 根据指定学生个数，逐个输入学生信息；

(2) 逐个显示学生表中所有学生的相关信息；

(3) 根据姓名进行查找，返回此学生的学号和成绩；

(4) 根据指定的位置可返回相应的学生信息（学号，姓名，成绩）；

(5) 给定一个学生信息，插入到表中指定的位置；

(6) 删除指定位置的学生记录；

(7) 统计表中学生个数。

## 四、源程序

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#define OK 1

#define ERROR 0

#define OVERFLOW -2

#include<iostream>

using namespace std;

typedef int Status;

typedef struct student

{

char num[10];

char name[20];

double score;

}ElemType;

typedef struct LNode{

ElemType data;

struct LNode \*next;

}LNode,\*LinkList;

Status InitList\_L(LinkList &L)

{

L=new LNode;

L->next=NULL;

return OK;

}

Status DestroyList\_L(LinkList &L)

{

LinkList p;

while(L)

{

p=L;

L=L->next;

delete p;

}

return OK;

}

Status ClearList(LinkList &L)

{

LinkList p,q;

p=L->next;

while(p)

{

q=p->next;

delete p;

p=q;

}

L->next=NULL;

return OK;

}

int ListLength(LinkList L)

{

LinkList p;

p=L->next;

int i=0;

while(p)

{

i++;

p=p->next;

}

return i;

}

int ListEmpty(LinkList L)

{

if(L->next)

return 0;

else

return 1;

}

Status GetElem(LinkList L,int i,ElemType &e)

{

LinkList p;

p=L->next;

int j=1;

while(p&&j<i)

{

p=p->next;

++j;

}

if(!p||j>i) return ERROR;

e=p->data;

return OK;

}

LNode\* LocateElem(LinkList L,char \*nam)

{

LNode \*p;

p=L->next;

while(p&&!strcmp(p->data.name,nam))

p=p->next;

return p;

}

Status ListInsert(LinkList &L,int i,ElemType e)

{

LNode \*p,\*s;

int j=0;

p=L;

while(p&&j<i-1)

{

p=p->next;

++j;

}

if(!p||j>i-1) return ERROR;

s=new LNode;

s->data=e;

s->next=p->next;

p->next=s;

return OK;

}

Status ListDelete\_L(LinkList &L,int i,ElemType &e)

{

LNode \*p,\*q;

int j;

p=L;j=0;

while(p->next && j<i-1){//寻找第i个结点，并令p指向其前驱

p=p->next; ++j;

}

if(!(p->next)||j>i-1)

return ERROR; //删除位置不合理

q=p->next; //临时保存被删结点的地址以备释放

p->next=q->next; //改变删除结点前驱结点的指针域

e=q->data; //保存删除结点的数据域

delete q; //释放删除结点的空间

return OK;

}

//前插法创建链表

void CreateList\_F(LinkList &L,int n)

{

LNode \*p;

int i;

L=new LNode;

L->next=NULL; //先建立一个带头结点的单链表

for(i=n;i>0;--i)

{

p=new LNode; //生成新的学生结点

cin>>p->data.num; //输入学号

cin>>p->data.name; //输入姓名

cin>>p->data.score; //输入成绩

p->next=L->next;L->next=p; //插入到表头

}

}//CreateList\_F

//尾插法创建单链表

void CreateList\_L(LinkList &L,int n)

{

//正位序输入n个元素的值，建立带表头结点的单链表L

LNode \*p,\*r;

int i;

L=new LNode;

L->next=NULL;

r=L; //尾指针r指向头结点

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

{ p=new LNode;//生成新学生结点

cin>>p->data.num; //输入学号

cin>>p->data.name; //输入姓名

cin>>p->data.score; //输入成绩

p->next=NULL;

r->next=p; //插入到表尾

r=p; //r指向新的尾结点

}

}//CreateList\_L

void Input(ElemType \*e)

{

cout<<"姓名:";

cin>>e->name;

cout<<"学号:";

cin>>e->num;

cout<<"成绩:";

cin>>e->score;

cout<<"完成输入\n\n";

}

void Output(ElemType \*e)

{

printf("姓名:%-20s\n学号:%-10s\n成绩:%-10.2lf\n\n",e->name,e->num,e->score);

}

int main()

{

LNode L;

LinkList p;

ElemType a,b,c,d;

cout<<"-------------------------\n";

cout<<"1. 构造顺序表\n";

cout<<"2. 录入指定人数的学生信息\n";

cout<<"3. 显示学生表中的所有信息\n";

cout<<"4. 根据姓名查找该学生，并返回学号和成绩\n";

cout<<"5. 根据某指定位置显示该学生信息\n";

cout<<"6. 在指定位置插入学生信息\n";

cout<<"7. 在指定位置删除学生信息\n";

cout<<"8. 统计学生个数\n";

cout<<"0. 退出\n";

cout<<"------------------------\n";

int n,choose=-1;

while(choose!=0)

{

puts("请输入你要选择的功能前的序号:");

cin>>choose ;

if(choose==0)

break;

else if (choose==1)

{

if(InitList(p))

cout<<"建立顺序表成功\n";

else

cout<<"建立顺序表失败\n";

}

else if (choose==2)

{

cout<<"将要输入学生的人数：";

cin>>n;

for(int i=1;i<=n;i++)

{

printf("第%d个学生:\n",i);

Input(&a);

ListInsert(&L,i,a);

}

}

else if (choose==3)

{

for(int i=1;i<=n;i++)

{

GetElem(&L,i,b);

Output(&b);

}

}

else if (choose==4)

{

char s[20];

cout<<"请输入要查找的学生姓名:";

cin>>s;

if(Search(L,s,p))

Output(&(p->data));

else

cout<<"对不起，查无此人";

puts("");

}

else if (choose==5)

{

cout<<"请输入要查询的位置:";

int id1;

cin>>id1;

GetElem(&L,id1,c);

Output(&c);

}

else if (choose==6)

{

cout<<"请输入要插入的位置:";

int id2;

cin>>id2;

cout<<"请输入学生信息:\n";

Input(&d);

if(ListInsert(&L,id2,d))

{

n++;

cout<<"插入成功"; ;

puts("");

}

else

{

cout<<"插入失败";

puts("");

}

}

else if (choose==7)

{

cout<<"请输入要删除的位置:";

int id3;

cin>>id3;

if(ListDelete(&L,id3))

{

n--;

cout<<"删除成功";

puts("");

}

else

{

cout<<"删除失败";

puts("");

}

}

else if (choose==8)

{

cout<<"已录入的学生个数为:"<<n<<endl;

break;

}

}

cout<<"\n\n谢谢您的使用，请按任意键退出\n\n\n";

system("pause");

return 0; }

总结和体会：

经过本次的数据结构上机实验，让我了解到了线性表在数据存储方面的优越性，通过再次实验让我明白了上机的重要性，也掌握了线性表的顺序存储结构、链式存储结构的设计与操作，对顺序表和单链表的建立、插入、删除等基本操作算法有了一定的了解。