**Application of linear list Experiment Report**

Class:计科201班 Student ID 1: 周宸 Name 1: 20401010119

Experiment Date:2021/11/6

Student ID 2：徐彤 Name 2：20401010110

Student ID 3：张雅康 Name 3：20401010126

1. **Experimental purpose**

1、Use the basic operations to implement the specific operations for the linear table;

2、Master the application of file operations;

3、Improve the understanding of the data structure of linked storage structure, and gradually cultivate the programming ability to solve practical problems.

1. **Experimental environment**

A computer with Visual C ++ 6.0 / CFree.

This experiment has 4 class hours in all.

1. **Experimental content**

Design a classmate's address list, requested as follows:

* Each student in the address list contains the following information: student id、name、telephone number. If you need more fields, please add them yourself.
* The program has a main menu containing the following functions:

1. Add a record: Add a student record from the input.
2. Delete a record: Delete a student record according to the student id from the input.
3. Output all records: Display all the records in the address list.
4. Search by name: Input the student name and then output the whole information of the student.
5. Save records: Save all the records in the address list to a certain file.
6. Clear records: Delete all the records in the address list and then delete the file.
7. Quit

**hint：**

* When the program starts, it should be determined whether there is a record file. If the file exists, read each record from it to the list.
* After the user selects and completes a function of the main menu, the program should return to the main menu.
* When a record is added, it should be inserted into the tail of the list.
* If a record does not exist when performing delete or and search operation, the program should output some information to the user.
* You do not need to write files when adding records or deleting records.
* When you want to save a record you’d better overwrite the file. (Or delete the original file first, and then save all the records)
* Each module is written in the form of a function, called by the main function.

**optional：**

* Add a sorting function in the main menu, the sorting result should be in an ascending order according to the student number. Sorting methods can be done by bubble sort or insert sort.

1. **Important data structures**

struct Student//创建Student结构体

{

string id;//Id名

string name;//姓名

string tel;//电话

};

typedef struct SNode//创建结点

{

Student data;//数据域

SNode \*next;//指针域

}SNode,\*SList;

1. **Implementation analysis**

SList creatList()//创建链表

{

SNode \*head=new SNode();//创建头结点

head->next=NULL;

return head;//返回头结点

}

函数部分

void AddStudent(SList &S,string name,string tel,string id)//加入学生

{

SNode \*P=new SNode();//创建新结点

SList K=S;

P->next=NULL;

while(K->next!=NULL)

{

K=K->next; //一个一个查找

}

K->next=P;//K作为头指针一般不放数据，所以K->next再指向数据

void DelStudent(SList &S,string id)//删除学生

void printStudent(SList &S)//打印链表

void SearchStudent(SList &S,string name)//查找

void SaveFile(SList &S)//存数据

void clear(SList &S)//清除链表

主函数部分

while(!input.eof())//读数据

{

input>>a;

ai++;

if(ai%3==1)//判断类型

name=a;

if(ai%3==2)

tel=a;

else

{

id=a;

AddStudent(S,name,tel,id);

}

}

1. **Debugging problem analysis**

Problems:Variable types in function call and function definition are easy to be confused

Solution:When defining a function, you need to define the types of formal parameters in the function; When calling a function, you only need to write the name of the function and the variables to be passed. The definition of constants and the role of basic library functions are best written at the top of the program. The precompiled instruction beginning .with # is not followed by a semicolon because this line is a precompiled instruction Surface Non statement; But typedef is a statement. Be sure to use a semicolon as the end of the statement.

1. **Summary**

The number of elements in the table is limited

The elements in the table have logical order, and each element in the sequence has its order

The data types of the elements in the table are the same, which means that each element occupies the same size of storage space

The elements in the table are abstract, that is, the one-to-one logical relationship between elements is discussed without considering the content represented by the elements

Linear table is a logical structure that represents the one-to-one adjacent relationship between elements

**Crew Division**

|  |  |  |
| --- | --- | --- |
| **Group division** | | |
| **Member name** | **Work done** | **Completion situation** |
| **周宸** | **代码与报告** | **100%** |
| **徐彤** | **代码与报告** | **100%** |
| **张雅康** | **代码与报告** | **100%** |