**Address List Design Experiment Report**

Class:计科201

Student ID 1: 20401010109 Name 1: 曹敬雨 Experiment Date:2021/11/2

Student ID 2： 20401010104 Name 2：徐汪洋

**Experimental purpose**

Design a classmate's address list.

1. **Experimental environment**

东四510

1. **Experimental content**

address list design

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

Each student in the address list contains the following information: student id、name、telephone number.

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

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.

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

{

int id;

string name;

string tel;

};

typedef student stu;

typedef struct

{

stu data[20];

int length=0;

}banji,\*pbanji;

void dayin(pbanji l);//打印输出学生信息

void exit();//判断数据文件是否存在

void duqu(pbanji l);//读取文件内容

void tianjia(pbanji l,int \_id,string \_name,string \_tel);//添加学生

void shanchu(pbanji l,int id\_);//删除学生

void chazhao(pbanji l,string m);

void save(pbanji l);

void qingchu(pbanji l);

void menu();

void yi\_menu(pbanji l);

void er\_menu(pbanji l);

void san\_menu(pbanji l);

void si\_menu(pbanji l);

void wu\_menu(pbanji l);

void liu\_menu(pbanji l);

void dayin(pbanji l)//打印输出学生信息

{

for(int i=0;i<l->length;i++)

{

cout<<l->data[i].id<<" "<<l->data[i].name<<" "<<l->data[i].tel<<endl;

}

}

void exit()//判断数据文件是否存在

{

cout<<"正在判断文件是否存在："<<endl;

ifstream file;

if(!access("xinxi.txt",0))

{

cout<<"文件存在！"<<endl;

}

else

{

cout<<"文件不存在！";

file.open("xinxi.txt");

cout<<"创建完成！"<<endl;

file.close();

}

}

void duqu(pbanji l)//读取文件内容

{

fstream read;

read.open("xinxi.txt");

int a;

string b;

string c;

while(!read.eof())

{

read>>a;

l->data [l->length].id=a;

read>>b;

l->data [l->length].name=b;

read>>c;

l->data [l->length].tel=c;

l->length++;

}

}

void tianjia(pbanji l,int \_id,string \_name,string \_tel)//添加学生

{

l->length++;

l->data [l->length-1].id=\_id;

l->data [l->length-1].name=\_name;

l->data [l->length-1].tel=\_tel;

cout<<"添加成功"<<endl;

}

void shanchu(pbanji l,int id\_)//删除学生

{

for(int i=0;i<l->length+1;i++)

{

if(id\_ ==l->data[i].id)

{

for(int j=i;j<l->length;j++)

{

l->data[j].id=l->data[j+1].id;

l->data[j].name=l->data[j+1].name;

l->data[j].tel=l->data[j+1].tel;

}

}

}

l->length--;

cout<<"删除成功！"<<endl;

}

void chazhao(pbanji l,string m)

{

for(int i=0;i<l->length+1;i++)

{

if(m ==l->data[i].name)

{

cout<<l->data[i].id<<" "<<l->data[i].name<<" "<<l->data[i].tel<<endl;

}

}

}

void save(pbanji l)

{

ofstream write;

write.open("xinxi.txt");

for(int i=0;i<l->length;i++)

{

write<<l->data[i].id<<" "<<l->data[i].name<<" "<<l->data[i].tel<<" ";

}

write.close();

cout<<"保存成功！"<<endl;

}

void qingchu(pbanji l)

{

for(int i=0;i<l->length;i++)

{

l->data[i].id=0;

l->data[i].name="0";

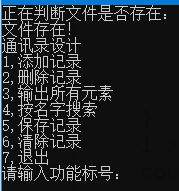
l->data[i].tel="0";

}

cout<<"清除成功！"<<endl;

}

1. **Implementation analysis**



首先判断文件是否存在，若存在，则读取并存入顺序表。后按照要求逐个实现通讯录的基本功能，最后再存入顺序表，后保存退出。

1. **Debugging problem analysis**

存入读取过程中出现了较多差一错误，经由单个调试后整体修改，最终实现目标。

1. **Summary**

该程序能将顺序表中的数据保存至文件中，实现数据的存储。本次顺序表基于静态数组，较少考虑顺序表存储容量。另外有较多差一错误，调试花费较多时间。

1. **Crew Division**

|  |  |  |
| --- | --- | --- |
| **Group division** | | |
| **Member name** | **Work done** | **Completion situation** |
| **曹敬雨** | **数据结构和单一功能函数实现** | **完成** |
| **徐汪洋** | **主函数中对功能函数的整合和排版实现** | **完成** |