# 线性表实验报告

Class: 计科201

StudentNo.: 20401180118 Name: 申宝龙.

StudentNo.: 20403070214 Name: 王镓玮

1. **Experimental purpose**

1. Use basic operation to realize the specific operation of linear table;

2. Master the application of file operation;

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**

**Visual Studio 2019**

1. **Experimental content**

To design a classmate's address book, the requirements are as follows:

* Each student in the address list contains the following information: student ID card, name and telephone number. If you need more fields, please add them yourself.
* This program has a main menu with the following functions:

1. Add record: add a student record from the input.
2. Delete record: Delete the student record according to the entered student ID card.
3. Output all records: displays all records in the address list.
4. Search by name: enter the name of the student, and then output all the information of the student.
5. Save records: save all records in the address book to a file.
6. Clear records: delete all records in the address list, and then delete the file.
7. give up

**Hint:**

* When the program starts, it should be determined whether there is a record file. If the file exists, read each record from the list.
* After the user selects and completes a function of the main menu, the program should return to the main menu.
* When adding a record, it should be inserted at the end of the list.
* If the record does not exist when deleting or searching, the program should output some information to the user.
* When adding or deleting records, there is no need to write files.
* When you want to keep records, you'd better overwrite the files. (or delete the original file first, and then save all records)
* Each module is written as a function and called by the main function.

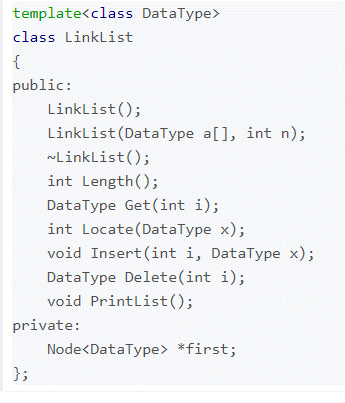
**Optional:**

* Add a sorting function in the main menu, and the sorting results should be arranged in ascending order according to the student number. Sorting can be done by bubbling sorting or inserting sorting.

1. **Important data structure.**

Sruct://The node consists of a data field for storing data elements and a pointer field for storing addresses of subsequent nodes.

|  |
| --- |
| string stuPhone; //student phone number. |
| string stuNumber; //student id. |
| string stuPhone; //student phone number. |
| Node<DataType> \* next |



1. **Implementation analysis**
   * + 1. Create a node dedicated to storing student information, struct.
       2. A class is a linked list, node < datatype > \* first;

As an empty linked list of first->NULL, it is also a pseudo-node, which does not store any information.

* + - 1. Add "add, delete, insert and look up" to the class, and have traversal functions.
      2. The main function calls the function realization function of the class.
      3. And we solved the main functions to be realized by using the list container.

1. **Debugging problem analysis.**

When programming, no information is added to store the number of students, which leads to a blank student information when saving the file, resulting in imperfection.

A screen clearing operation without writing resulted in messy interface.

For example, the sort function of list can be used for simple sorting, but some systems do not have this sort of nodes, so they can be defined.

bool myCompare(StudentInfo val1, StudentInfo val2)

{

return val1.stuNumber < val2.stuNumber;

}

void fun7(list<StudentInfo>& x)

{

x.sort(myCompare);

Cout < < "done!" ;

system("pause");

system("cls");

}

1. **abstract**

Understand the structure of linear table and how to store data; Familiar with the operation of writing files.

Understand that c++ can also call the < list > container of stl database, which can realize functions more easily. Modular programming is:

Very important, the availability is quite high, which greatly saves time and improves efficiency.

1. **member**

|  |  |  |
| --- | --- | --- |
| **group** | | |
| **Member name** | **Complete the work.** | **performance** |
| **申宝龙** | **The functional experiment of the main class of.** | **complete** |
| 王镓玮 | **Menu reading and writing file function of main function** | **complete** |
|  |  |  |