Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

Восточно-Сибирский государственный университет технологий и управления

(ФГБОУ ВСГУТУ)

Электротехнический факультет

Кафедра «Системы Информатики»

Дисциплина

«Программирования»

Отчет по лабораторной работе №3

На тему:

Файловый ввод и вывод

Вариант3

|  |  |
| --- | --- |
| Выполнил: | студент гр. Б661-2 |
|  | Коковихин А.В. |
| Проверил: | преподаватель  Жамбалов Э.Б. |

Улан-Удэ

2022

1. **Словесная постановка задачи**

Измените программу индивидуального задания прошлой лабораторной работы. Добавьте в меню выбора новые действия пользователя «Сохранение» и «Загрузка». Действие «Сохранение» должно выполнять сохранение в файл всех данных вашего списка.

При этом пользователю должен предоставляться выбор: сохранить в файл по умолчанию (придумайте название, к примеру, list.data) или пользователь сам вводит название файла, в который нужно сохранить список. Действие «Загрузить» должно выполнять загрузку данных вашего списка (заполнять список) из файла в память программы.

При этом пользователю также должен предоставляться выбор: загрузить из файла по умолчанию или пользователь сам вводит название файла, из которого нужно загрузить список.

1. **Решение задачи**
   1. **Формальная постановка задачи**

Для выполнения данной работы будут использованы потоки ifstream и ofstream.

Один из потоков используется для записи в файл – ofsteam, а другой для чтения из них ifstream. При вызове одной из оглашённых функций будет осуществляться работа с одним из текущих потоков.

* 1. **Проектирование**

При сохранении файла у пользователя должен быть выбор между сохранение в файл по умолчанию или же в файл с произвольным названием. С загрузкой данных из файла должна быть осуществлена похожая система.

Данные в файлах будут записаны построчно, то есть каждая запись таблицы будет записана на одной строке.

И иметь примерно следующий вид:

Данное1\_Данное2\_Данное3\_и т.д.

При чтении данные будут считываться поочередно и записываться по отдельным полям.

* 1. **Реализация**

Эта работа отличается лишь добавлением 2 новых пунктов меню, поэтому следующие элементы были подвергнуты редактированию:

enum class MenuCommand

{

Exit,

Create,

Sort,

Edit,

Delete,

Show\_All,

Search,

Save\_As,

Load\_As,

Save,

Load,

};

MenuCommand Show\_Menu()

{

cout << "0) Exit" << endl;

cout << "1) Create" << endl;

cout << "2) Sort" << endl;

cout << "3) Edit" << endl;

cout << "4) Delete" << endl;

cout << "5) Show\_All" << endl;

cout << "6) Search" << endl;

cout << "7) Save\_As" << endl;

cout << "8) Load\_As" << endl;

cout << "9) Save" << endl;

cout << "10) Loads" << endl;

cout << "Chouse command: ";

int i;

cin >> i;

return (MenuCommand)i;

}

Сохранение, как и было сказано ранее осуществляется посредством потока ofstream. Первым делом, запрашиваем у пользователя выбрать 1 из пунктов меню, после чего в каждой из процедур, будем создавать поток ofstream, после чего открывать файл в 1-й из процедур это файл по умолчанию «LR\_3\_Individual.txt» а в другой это произвольный файл. После чего проходясь по всему списку с самого начала записываем данные в файл.

Функция сохранения в файл по умолчанию.

void Save(LR\_2::Linked\_PersonPointers\* head, const char filename[] = "LR\_3\_Individual.txt")

{

ofstream out;

out.open(filename);

if (out.is\_open())

{

LR\_2::Linked\_PersonPointers\* curent = head;

while (curent)

{

LR\_2::Person p = \*curent->Person;

out << p.Adress << ' ' << p.Name.FirstName << ' ' << p.Name.SecondName << ' ' << p.Name.Has\_MidleName << ' ' << p.Name.MidleName << ' ' << p.Phone\_Number;

curent = curent->Next;

if (curent)

out << endl;

}

out.close();

}

}

Функция сохранения в произвольный файл.

void Save\_As(LR\_2::Linked\_PersonPointers\* head)

{

if (head == NULL)

{

cout << "List is Empty!" << endl;

return;

}

cout << "Enter SaveFile Name: ";

string filename;

cin >> filename;

filename += ".txt";

Save(head, filename.c\_str());

}

Загрузка, осуществляется с помощью потока для чтения ifstream. Также как и в сохранение требуем выбрать 1 из пунктов меню, после чего создаём поток чтения ifsteam, открываем файл и с помощью цикла пока не встретим конец файла будем считывать данные с файла и записывать их в структуру.

void Load(LR\_2::Linked\_PersonPointers\*& head, LR\_2::Linked\_PersonPointers\*& tail, const char filename[] = "LR\_3\_Individual.txt")

{

ifstream input;

input.open(filename);

if (input.is\_open())

{

while (input.eof() == false)

{

LR\_2::Person\* p = new LR\_2::Person;

input >> p->Adress;

input >> p->Name.FirstName;

input >> p->Name.SecondName;

input >> p->Name.Has\_MidleName;

if (p->Name.Has\_MidleName)

input >> p->Name.MidleName;

input >> p->Phone\_Number;

LR\_2::Add\_Person(head, tail, p);

}

input.close();

}

}

void Load\_As(LR\_2::Linked\_PersonPointers\*& head, LR\_2::Linked\_PersonPointers\*& tail)

{

cout << "Enter LoadFile Name: ";

string filename;

cin >> filename;

filename += ".txt";

Load(head, tail, filename.c\_str());

}

Главная функция также претерпела небольшие изменения.

void Individual()

{

LR\_2::Linked\_PersonPointers\* head = NULL;

LR\_2::Linked\_PersonPointers\* tail = NULL;

while (true)

{

switch (Show\_Menu())

{

case MenuCommand::Create:

LR\_2::Create\_Person(head, tail);

break;

case MenuCommand::Delete:

LR\_2::Delete\_Person(head, tail);

break;

case MenuCommand::Edit:

LR\_2::Edit\_Person(head, tail);

break;

case MenuCommand::Search:

LR\_2::Search(head);

break;

case MenuCommand::Show\_All:

LR\_2::Print(head);

break;

case MenuCommand::Sort:

LR\_2::Sort(head, tail);

LR\_2::Print(head);

break;

case MenuCommand::Save\_As:

Save\_As(head);

break;

case MenuCommand::Load\_As:

Load\_As(head, tail);

break;

case MenuCommand::Save:

Save(head);

break;

case MenuCommand::Load:

Load(head, tail);

break;

case MenuCommand::Exit:

return;

default:

cout << "Wrong command!";

}

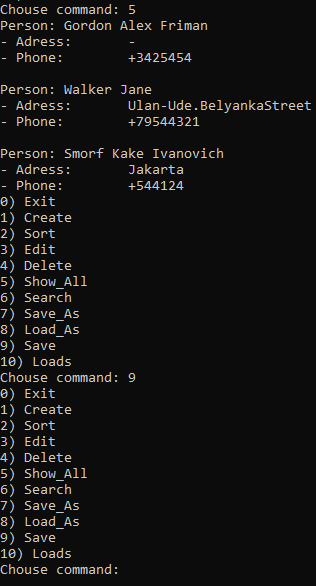
}

}

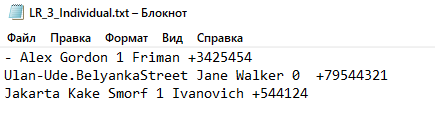
* 1. **Тестирование**

Первым делом проверим сохранение по умолчанию

Сейчас в списке есть следующие записи:

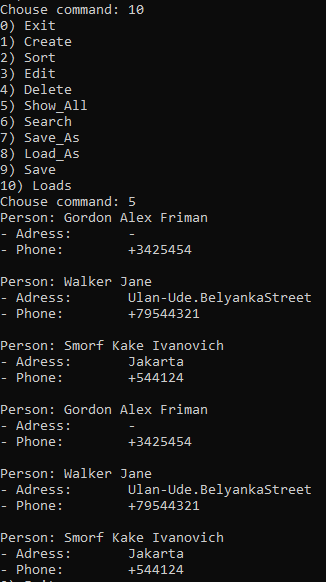






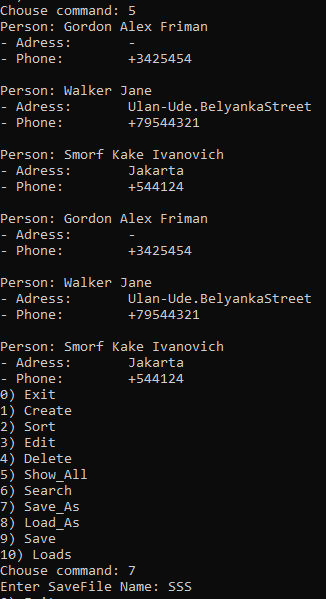
Всё работает корректно.

Теперь, проверим загрузку по умолчанию

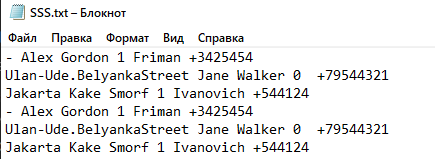


Записи загружаются корректно.

Теперь попробуем создать произвольный файл

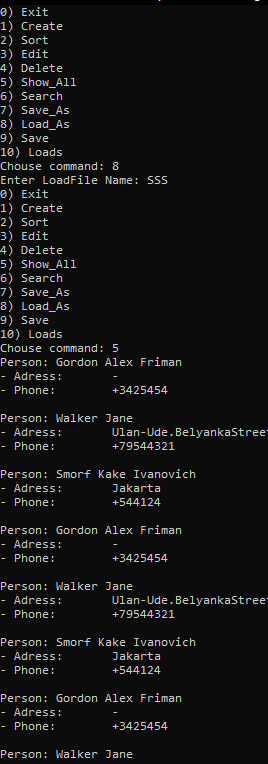






Произвольный файл создался и данные корректно записались в него.

Теперь проделаем тоже самое с загрузкой из произвольного файла.



Все новые функции работаю корректно и тестирование можно считать успешно завершённым.

**Заключение**

В данной работе я познакомился с файловым вводом и выводом, потоками и методами работы с ними. А также подкрепил полученные знания добавив функции сохранения и загрузки.