**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Брянская государственная инженерно - технологическая академия**

**Кафедра**

**Информационных технологий**

**Использование списков для хранения информации**

Отчет о лабораторной работе №11.

Дисциплина: «Технологии программирования».

Студент группы ИСТ-201 \_\_\_\_\_\_\_\_\_\_ Д. А. Дюльгер

Преподаватель \_\_\_\_\_\_\_\_\_\_ А. В. Яковлев

Брянск 2014

**Целью лабораторной работы** являетсяполучение навыков работы со списками.

**Краткие теоретические сведения.**

Списком называется упорядоченный набор элементов(структур), количество которых не определено и к которым могут применяться операции включения новых и исключения имеющихся элементов. Для работы со списками, необходимо иметь несколько функций, позволяющих добавлять новый элеент к списку, удалять элемент и производить поиск

Существуют следующие разновидности списков:

* односвязные(однонаправленные) списки;
* двусвязные(двунаправленные) списки;
* кольцевые(циклические) списки.

Кольцевые списки могут быть как односвязными, так и двусвязными.

**Выполнение задания**

Создать базу данных студенческой группы и применить к ней следующие операции:

1. Ввод данных;
2. Просмотр всех данных;
3. Просмотр данных об одном абоненте;
4. Запись данных в файл.

**Листинг 1. Файл Dbase.h**

#ifndef \_\_DBASE\_H\_\_

#define \_\_DBASE\_H\_\_

// структура

struct SItem

{

int ID;

int CoursID;

char GroupID[50];

char Name[25];

char Surname[25];

char Patronymic[25];

char YearOfBir[25];

char ManGen[20];

long long int PhoneNum;

SItem \*next;

};

// Наше пространство имён //

namespace StudBase

{

// =========ПРОТОТИПЫ ФУНКЦИЙ============

void StudElem(int ID, SItem \*Head);

void StudList(SItem \*Head);

void AddItem(SItem \*Head, SItem \*NewItem, SItem \*\*Teil);

void SaveList(SItem \*Head);

}

#endif

**Листинг 2. Файл Dbase.cpp**

#include "stdafx.h"

#include <locale.h>

#include <iostream>

#include <fstream>

#include <windows.h>

#include "Dbase.h"

using namespace std;

///////////////////////////////////////////////////////

// Вывод одного элемента в окно консольного приложения.

void StudBase::StudElem(int ID, SItem \*Head)

{

SItem \*Curr; // Буфер

Curr = Head;

while(Curr->ID != ID)

{

if(Curr->next == NULL)

{

cout << "ОШИБКА! Элемента с таким номером в списке нет." << endl;

cout << endl;

return;

}

Curr = Curr->next;

}

cout << "\t" << "ФИО студента:" << endl;

cout << "\t" << Curr->Surname << " " << Curr->Name << " " << Curr->Patronymic << endl;

cout << "\t" << "Группа студента: " << Curr->GroupID << endl;

cout << "\t" << "Курс студента: " << Curr->CoursID << endl;

cout << "\t" << "Дата рождения: " << Curr->YearOfBir << endl;

cout << "\t" << "Половая принадлежность: " << Curr->ManGen << endl;

cout << "\t" << "Телефонный номер: " << Curr->PhoneNum << endl << endl;

}

////////////////////////////////////////////////////////

// Вывод всех элементов списка

void StudBase::StudList(SItem \*Head)

{

SItem \*Curr;

Curr = Head;

while(Curr->next)

{

cout << "\t" << "Номер элемента: " << Curr->ID << endl;

cout << "\t" << "ФИО студента:" << endl;

cout << "\t" << Curr->Surname << " " << Curr->Name << " " << Curr->Patronymic << endl;

cout << "\t" << "Группа студента: " << Curr->GroupID << endl;

cout << "\t" << "Курс студента: " << Curr->CoursID << endl;

cout << "\t" << "Дата рождения: " << Curr->YearOfBir << endl;

cout << "\t" << "Половая принадлежность: " << Curr->ManGen << endl;

cout << "\t" << "Телефонный номер: " << Curr->PhoneNum << endl << endl;

Curr = Curr->next;

}

cout << "\t" << "Номер элемента: " << Curr->ID << endl;

cout << "\t" << "ФИО студента:" << endl;

cout << "\t" << Curr->Surname << " " << Curr->Name << " " << Curr->Patronymic << endl;

cout << "\t" << "Группа студента: " << Curr->GroupID << endl;

cout << "\t" << "Курс студента: " << Curr->CoursID << endl;

cout << "\t" << "Дата рождения: " << Curr->YearOfBir << endl;

cout << "\t" << "Половая принадлежность: " << Curr->ManGen << endl;

cout << "\t" << "Телефонный номер: " << Curr->PhoneNum << endl << endl;

}

////////////////////////////////////////////////////////////////

// Добавление нового элемента в список.

void StudBase::AddItem(SItem \*Head, SItem \*NewItem, SItem \*\*Teil)

{

NewItem->ID = (\*Teil)->ID + 1;

// изменение указателя next первого элемента

if((\*Teil)->ID == 0)

Head->next = NewItem;

NewItem->next = NULL;

(\*Teil)->next = NewItem;

// Ввод данных

cout << "Введите фамилию: ";

cin >> NewItem->Surname;

OemToAnsi(NewItem->Surname, NewItem->Surname);

cout << "Введите имя: ";

cin >> NewItem->Name;

OemToAnsi(NewItem->Name, NewItem->Name);

cout << "Введите отчество: ";

cin >> NewItem->Patronymic;

OemToAnsi(NewItem->Patronymic, NewItem->Patronymic);

cout << "Введите группу: ";

cin >> NewItem->GroupID;

OemToAnsi(NewItem->GroupID, NewItem->GroupID);

cout << "Введите курс: ";

cin >> NewItem->CoursID;

cout << "Введите день, месяц и год рождения: ";

cin >> NewItem->YearOfBir;

OemToAnsi(NewItem->YearOfBir, NewItem->YearOfBir);

cout << "Введите пол: ";

cin >> NewItem->ManGen;

OemToAnsi(NewItem->ManGen, NewItem->ManGen);

cout << "Введите номер телефона: ";

cin >> NewItem->PhoneNum;

\*Teil = NewItem; // Концом списка становится новый элемент.

}

////////////////////////////////////////////////////////////////

// Сохранение списка в текстовый файл.

void StudBase::SaveList(SItem \*Head)

{

SItem \*Curr;

Curr = Head;

// создаём объект txtfile

ofstream data;

data.open("StudDatabase.txt");

while(Curr->next)

{

data << "Номер элемента: " << Curr->ID << endl;

data << "ФИО студента:" << endl;

data << Curr->Surname << " " << Curr->Name << " " << Curr->Patronymic << endl;

data << "Группа студента: " << Curr->GroupID << endl;

data << "Курс студента: " << Curr->CoursID << endl;

data << "Дата рождения: " << Curr->YearOfBir << endl;

data << "Половая принадлежность: " << Curr->ManGen << endl;

data << "Телефонный номер: " << Curr->PhoneNum << endl << endl;

Curr = Curr->next;

}

data << "Номер элемента: " << Curr->ID << endl;

data << "ФИО студента:" << endl;

data << Curr->Surname << " " << Curr->Name << " " << Curr->Patronymic << endl;

data << "Группа студента: " << Curr->GroupID << endl;

data << "Курс студента: " << Curr->CoursID << endl;

data << "Дата рождения: " << Curr->YearOfBir << endl;

data << "Половая принадлежность: " << Curr->ManGen << endl;

data << "Телефонный номер: " << Curr->PhoneNum << endl << endl;

cout << "Запись данных в файл завершена." << endl << endl;

data.close();

}

**Листинг 3. StudDatabase.cpp**

////////////////////////////////////////////////////////////

// Лабораторная работа №11.

// Тема: "Использование списков для хранения информации".

// Вариант №6.

// Студент Дюльгер Д. А.

// Группа ИСТ-201.

////////////////////////////////////////////////////////////

#include "stdafx.h"

#include <locale.h>

#include <iostream>

#include <conio.h>

#include "Dbase.h"

#include <windows.h>

using namespace std;

using namespace StudBase; // наше пространство имён

// нумерация команд меню

enum menu {EXIT, NEW\_ELEM, PRINT\_ELEM, PRINT\_ALL, SAVE\_FILE, CLEAR\_CMD, END};

int Menu();// прототип

void \_tmain()

{

setlocale(LC\_CTYPE, ".1251");

SItem \*Head = NULL;

SItem \*\*Teil;

SItem \*NewItem;

int CmdMenu;

int ID;

// создание первого элемента списка

Head = new SItem;

Head->ID = 0;

Head->next = NULL;

cout << "Введите имя: ";

cin >> Head->Name;

OemToAnsi(Head->Name, Head->Name);

cout << "Введите фамилию: ";

cin >> Head->Surname;

OemToAnsi(Head->Surname, Head->Surname);

cout << "Введите отчество: ";

cin >> Head->Patronymic;

OemToAnsi(Head->Patronymic, Head->Patronymic);

cout << "Введите группу: ";

cin >> Head->GroupID;

OemToAnsi(Head->GroupID, Head->GroupID);

cout << "Введите курс: ";

cin >> Head->CoursID;

cout << "Введите день, месяц и год рождения: ";

cin >> Head->YearOfBir;

OemToAnsi(Head->YearOfBir, Head->YearOfBir);

cout << "Введите пол: ";

cin >> Head->ManGen;

OemToAnsi(Head->ManGen, Head->ManGen);

cout << "Введите номер телефона: ";

cin >> Head->PhoneNum;

Teil = new SItem\*;

\*Teil = Head;

//==========ВЫЗОВ МЕНЮ=========//

for(;;)

{

CmdMenu = Menu();

switch (CmdMenu)

{

case EXIT:

return;

break;

case NEW\_ELEM:

NewItem = new SItem;

AddItem(Head, NewItem, Teil);

break;

case PRINT\_ELEM:

cout << "Номер элемента: ";

cin >> ID;

StudElem(ID, Head);

break;

case PRINT\_ALL:

StudList(Head);

break;

case SAVE\_FILE:

SaveList(Head);

break;

case CLEAR\_CMD:

system("cls");

break;

}

}

}

// программирование меню

int Menu()

{

int CmdMenu;

cout << endl << "МЕНЮ:" << endl;

for(;;)

{

cout << EXIT << " - выход" << endl;

cout << NEW\_ELEM << " - новый элемент" << endl;

cout << PRINT\_ELEM << " - вывести на экран элемент" << endl;

cout << PRINT\_ALL << " - вывести на экран все элементы" << endl;

cout << SAVE\_FILE << " - сохранить в файле" << endl;

cout << CLEAR\_CMD << " - очистка экрана" << endl;

cout << "Номер команды меню: ";

cin >> CmdMenu;

cout << endl;

if(CmdMenu < 0 || CmdMenu >= END)

{

cout << "ОШИБКА - такой команды нет." << endl;

continue;

}

else

return CmdMenu;

}

}

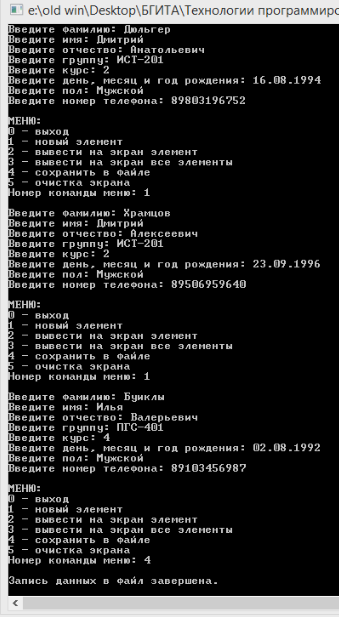


Рис.1 – Результат работы программы **StudDatabase.exe**

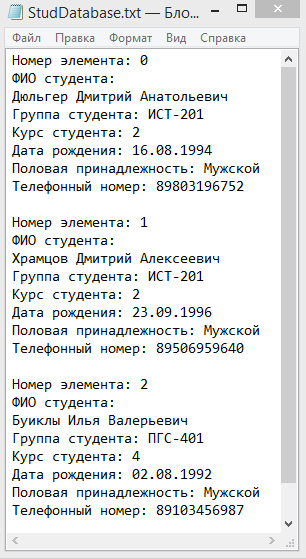


Рис.2 – Текстовый файл, содержащий данные списка.

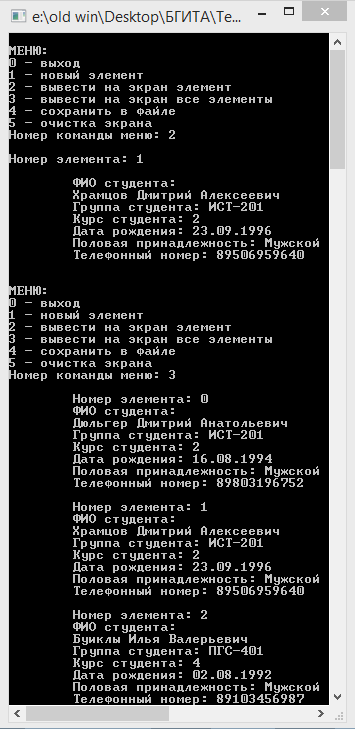


Рис.3 – Результат работы программы **StudDatabase.exe**