Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Кафедра ИС

Отчет

По дисциплине: “Алгоритмизация и программирование”

Лабораторная работа №5

“Программирование линейных

списков на языке С/С++”

Выполнил:

ст.гр. ИС/б-12

Долженко И.А.

Проверил:

Сметанина Т.И.

Севастополь

2018

1 ЦЕЛЬ РАБОТЫ

Изучение списковых структур данных и приобретение навыков разработки и отладки программ, использующих динамическую память. Исследование особенностей организации списков средствами языка С/С++

2 ПОСТАНОВКА ЗАДАЧИ

Представить одну из приведенных ниже таблиц в виде линейного списка L, элементами которого являются строки таблицы. Написать функции организации, добавления элемента в список, исключения элемента из списка, просмотра списка, а также одну из функций в соответствии с вариантом, приведенным ниже.

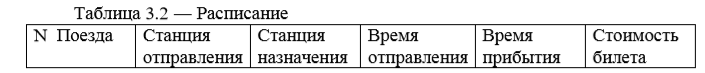


Таблица 3.2. Функцию, которая вставляет в конец стека новый элемент.

3 СТРУКТУРНАЯ СХЕМА АЛГОРИТМА

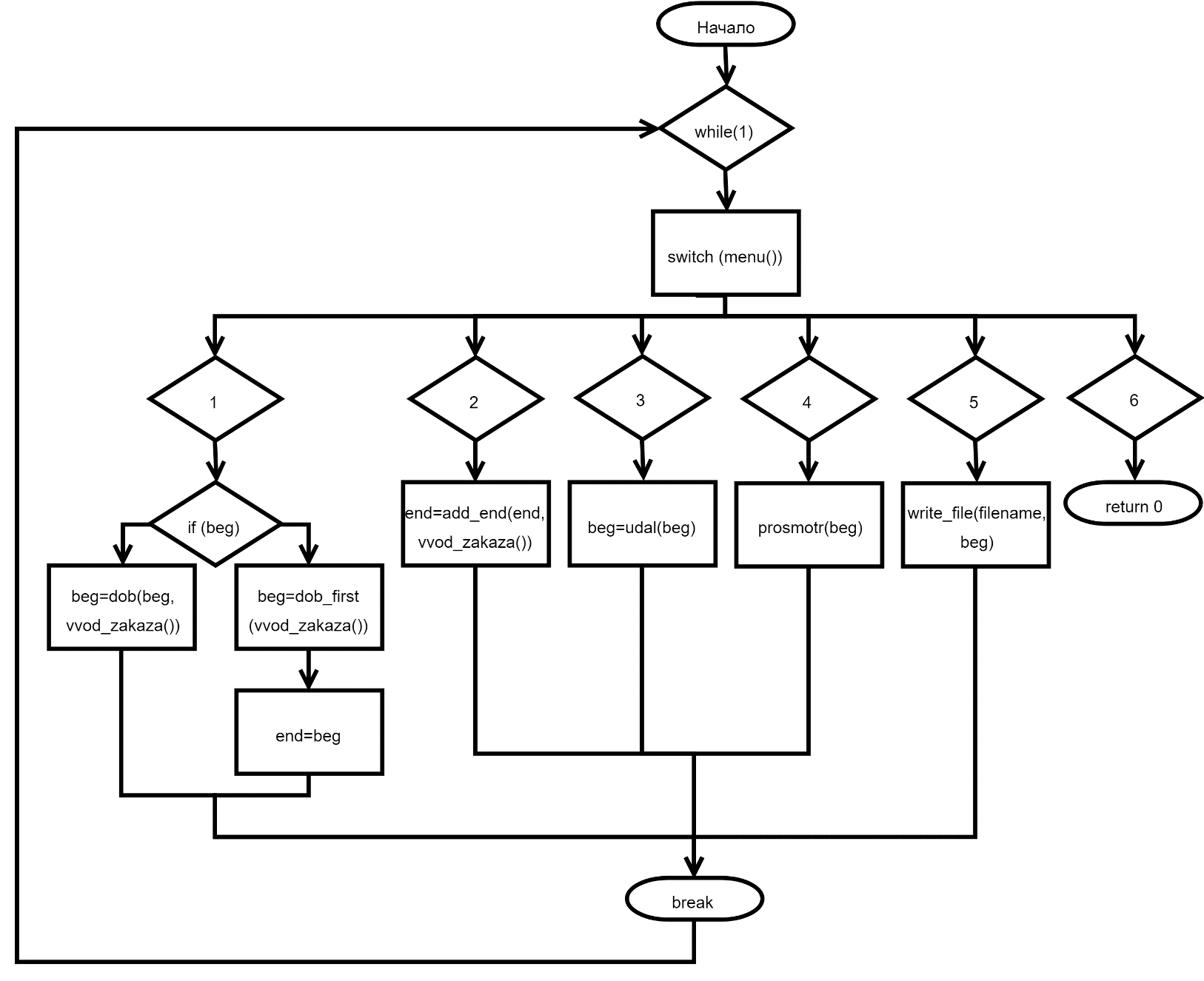


Рисунок 1 – Основная программа

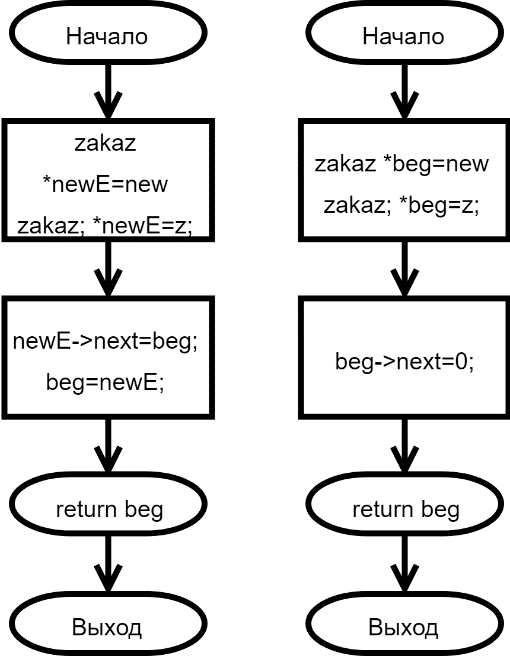


Рисунок 2 – Добавление элемента и создание первого элемента

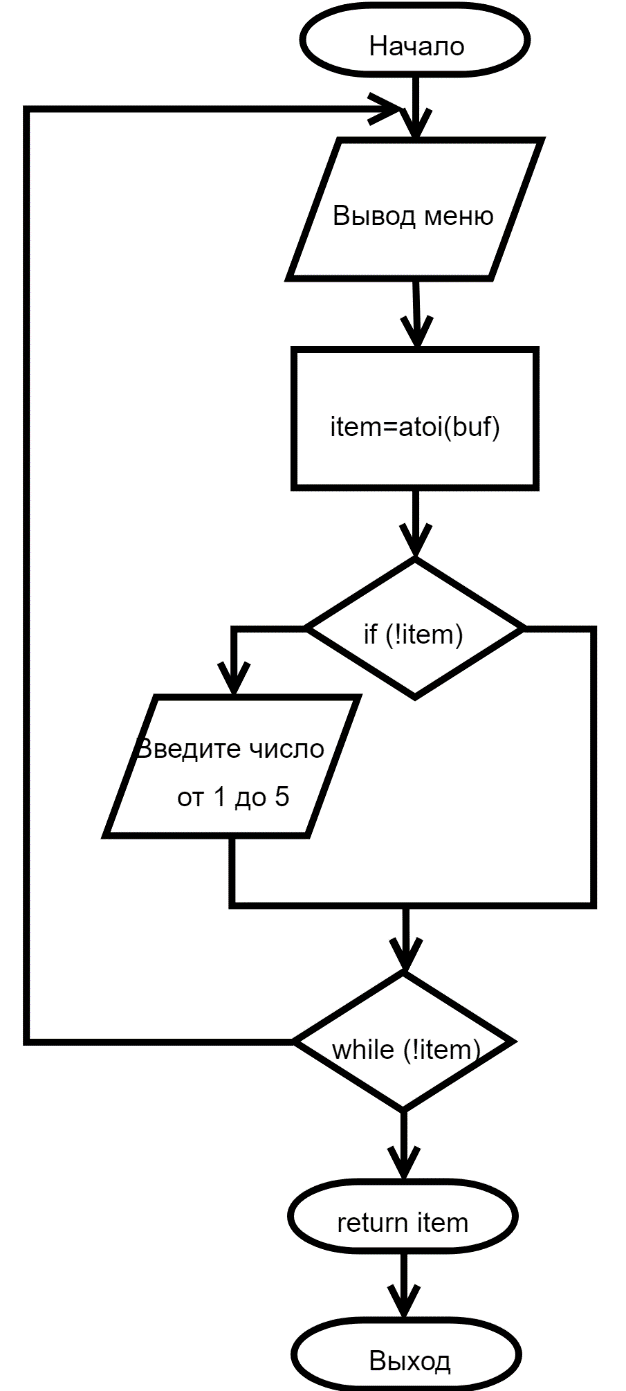
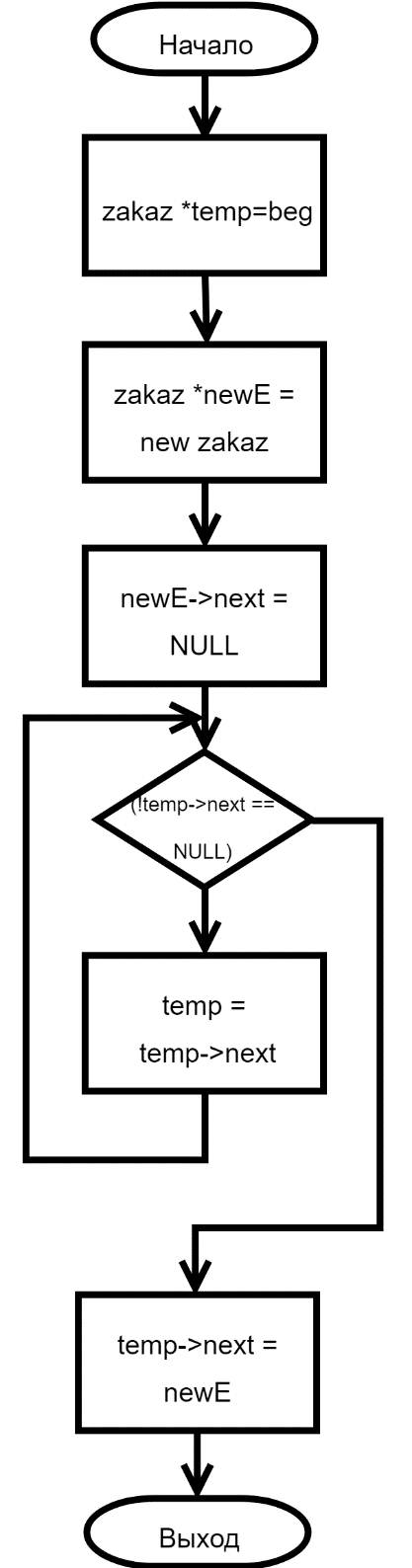


Рисунок 3 – Отображение меню и ввод пункта меню

Рисунок 4 – Добавление элемента в конец

4 ТЕКСТ ПРОГРАММЫ

#include <fstream>

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

#include <time.h>

#include <iomanip>

#include <conio.h>

#include <iostream>

using namespace std;

const int d\_o=20, d\_n=20;

struct zakaz

{

int nomer;

char otpravlenie[d\_o], naznachenie[d\_n];

float price, time\_o, time\_p;

zakaz\* next;

};

zakaz\* dob(zakaz\* beg, const zakaz& z);

zakaz\* add\_end(zakaz \*beg,const zakaz& z);

zakaz\* dob\_first(const zakaz& z);

zakaz\* udal(zakaz\* beg);

zakaz vvod\_zakaza();

int menu();

void print(const zakaz& z);

void prosmotr(zakaz\* beg);

int read\_file(char\* filename, zakaz\*\* beg, zakaz\*\* end);

int write\_file(char\* filename, zakaz\* temp);

//-------------------------основная функция------------------------------

int main()

{

setlocale(0,"RUS");

zakaz \*beg=0, \*end=0;

char \*filename="123.txt";

read\_file (filename,&beg, &end);

while (1)

{

switch (menu())

{

case 1: if (beg) beg=dob(beg,vvod\_zakaza());

else {

beg=dob\_first(vvod\_zakaza());

end=beg;

}

break;

case 2: end=add\_end(end,vvod\_zakaza());

break;

case 3: beg=udal(beg);

cout<<"Нажмите любую клавишу"<<endl;

cin.get();

break;

case 4: prosmotr(beg);

break;

case 5: write\_file(filename,beg);

break;

case 6: return 0;

default:

cout<<"Вам следует ввести номер от 1 до 5"<< endl;

cin.get();

break;

}

}

}

//-----------------------добавление элемента-----------------------------

zakaz\* dob(zakaz \*beg,const zakaz& z)

{

zakaz \*newE=new zakaz;

\*newE=z;

newE->next=beg;

beg=newE;

return beg;

}

//----------------------создание первого элемента------------------------

zakaz\* dob\_first(const zakaz& z)

{

zakaz \*beg=new zakaz; //выделение памяти под элемент

\*beg=z; //присвоить значения элементу

beg->next=0; //ссылка на следующий эл-т нулевая

return beg; //вернуть указатель на элемент

}

// ---------------- печать и удаление элемента---------------------------

zakaz\* udal(zakaz \*beg)

{

zakaz \*temp;

if (!beg) { cout<<"Стек пустой"<<endl; return 0; }

cout<<"=============================="<<endl;

print(\*beg); //печать значений головного эл-та

cout<<"=============================="<<endl;

temp=beg; //запомнить указатель на голову

beg=beg->next; //перевод beg на следующий эл-т

delete temp; //удаление элемента

return beg; //вернуть измененное значение beg

}

//-------------------ввод значений элемента------------------------------

zakaz vvod\_zakaza()

{

zakaz z;

cout<<"Введите станцию отправления:"<<endl;

cin.getline(z.otpravlenie,d\_o);

cout<<"Введите станцию назначения:"<<endl;

cin.getline(z.naznachenie,d\_n);

cout<<"Введите время отправления:"<<endl;

cin>>z.time\_o;

cout<<"Введите время прибытия:"<<endl;

cin>>z.time\_p;

cout<<"Введите стоимость билета:"<<endl;

cin>>z.price;

cout<<"Введите номер поезда:"<<endl;

cin>>z.nomer;

return z;

}

//-----------------отображение меню и ввод пункта меню-----------------------

int menu()

{

char buf[10];

int item;

do

{

system("CLS");

cout<<"1- Добавление элемента в стек"<<endl;

cout<<"2- Добавление элемента в конец"<<endl;

cout<<"3- Удаление элемента"<<endl;

cout<<"4- Просмотр стека"<<endl;

cout<<"5- Запись данных в файл"<<endl;

cout<<"6- Выход"<<endl;

cout<<"============================"<<endl;

cout<<"Введите номер пункта меню:"<<endl;

cin>>buf;

cin.get();

item=atoi(buf); //преобразовать его в целое

if (!item)

{

cout<<"Вам следует вводить число от 1 до 5"<<endl;

cin.get();

}

}

while (!item);

return item;

}

// -----------------------------печать заказа-----------------------------

void print(const zakaz& z)

{

cout<<"Станция отправления: " <<z.otpravlenie <<endl;

cout<<"Станция назначения: "<<z.naznachenie <<endl;

cout<<"Время отправления: "<<z.time\_o <<endl;

cout<<"Время прибытия: "<<z.time\_p <<endl;

cout<<"Стоимость билета: "<<z.price <<endl;

cout<<"Номер поезда: " <<z.nomer <<endl;

}

//----------------------------просмотр стека---------------------------

void prosmotr(zakaz \*beg)

{

if (!beg)

{

cout<<"Стек пустой"<<endl;

cout<<"Нажмите любую клавишу"<<endl;

cin.get();

return;

}

zakaz \*temp=beg;

cout<<"=============================="<<endl;

while (temp)

{

print(\*temp);

cout<<"=============================="<<endl;

cout<<"Нажмите любую клавишу"<<endl;

cin.get();

temp=temp->next;

}

}

// ------------------------------чтение из файла-----------------------------

int read\_file(char\* filename,zakaz\*\* beg, zakaz\*\* end)

{

ifstream fin(filename,ios::in);

if (!fin) { cout<<"Нет файла" <<filename<<endl; return 1;}

zakaz z;

\*beg = 0;

while (fin.getline(z.otpravlenie,d\_o))

{

fin.getline(z.naznachenie,d\_n);

fin>>z.time\_o;

fin>>z.time\_p;

fin>>z.price;

fin>>z.nomer;

fin.get();

if (\*beg) \*beg=dob(\*beg,z);

else

{ \*beg=dob\_first(z); \*end=\*beg; }

}

return 0;

}

// ----------------------------запись в файл ----------------------------

int write\_file(char\* filename, zakaz\* temp)

{

ofstream fout(filename);

if (!fout) {cout<<"Не могу открыть файл для записи"<<endl; return 1;}

while (temp)

{

fout<<temp->otpravlenie <<endl;

fout<<temp->naznachenie <<endl;

fout<<temp->time\_o <<endl;

fout<<temp->time\_p <<endl;

fout<<temp->price <<endl;

fout<<temp->nomer <<endl;

temp=temp->next;

}

cout<<"Данные сохранены в файле: "<<filename<<endl;

cin.get();

return 0;

}

// -------------------добавление элемента в конец------------------------

zakaz\* add\_end(zakaz \*beg,const zakaz& z)

{

zakaz \*temp=beg;

zakaz \*newE = new zakaz;

newE->next = NULL;

while (!temp->next == NULL)

{

temp = temp->next;

}

temp->next = newE;

}

5 ТЕСТИРОВАНИЕ

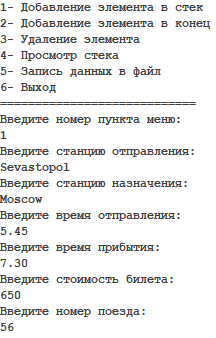


Рисунок 5 – Добавление элемента в стек

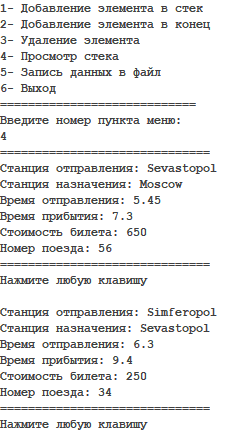


Рисунок 6 – Просмотр стека до удаления элемента

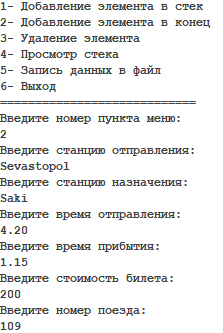


Рисунок 7 – Добавление элемента в конец стека

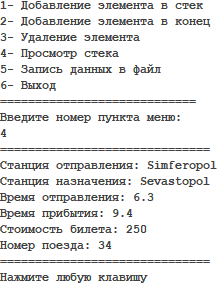


Рисунок 8 – Просмотр стека после удаления элемента

ВЫВОД

В ходе выполнения лабораторной работы были получены навыки работы с линейными списками. Изучены виды списков и их организация. Была написана программа с функциями организации стека, добавления элемента в стек, исключение элемента из стека, просмотр списка и функция добавления элемента в конец стека.