## Лабораторная работа №7

## РАБОТА СО СПИСКАМИ

**Цели работы**

Приобретение навыков работы со списками.

**Постановка задачи**

Составить универсальную программу (псевдокласс) работы с очередями. Вызов функций работы с очередями (формирование очереди вводом с клавиатуры, вывод очереди на экран монитора, добавление элементов в очередь, удаление элементов из очереди, удаление (стирание) очереди, завершение работы с программой) оформить с помощью консольного меню. Очередь реализовать на базе двусвязного списка.

**Описание программы**

Программа объявляет переменные: размер списка, добавляемые при инициализации элементы, новый элемент, удаляемый элемент. Пользователь вводит с клавиатуры размер списка, это значение проходит проверку на тип данных при помощи cin.fail(). Проверяется и принадлежность к натуральным числам. В при некорректном вводе он повторяется.

Объявляется пустой список и инициализируется с клавиатуры пользователем. Исходный список выводится на экран. Далее пользователю предлагается добавить новый элемент в конец (push\_back) и начало (push\_front) списка. Далее требуется удалить (remove()) любой элемент на выбор. Измененный список выводится на экран. Далее список сортируется (sort()) по возрастанию и отображается. После всех операций список удаляется, программа завершается.

**Описание алгоритма**

Начало

Инициализация size, element, newelement, deleted

Ввод size;

size<1

true

cout<<”Введите корректное size”;

Объявление list<int> myList;

false

3/1

Рис. 1. Схема алгоритма программы

2/1

true

i <=size;

i = 1;

false

cin >> element;

myList.push\_back(element);

i++;

cout << "Исходный список: ";

copy(myList.begin(), myList.end(), ostream\_iterator<int>(cout, " "));

cin >> newelement;

cout << "Добавить новый элемент в конец: " << endl;

myList.push\_back(newelement);

4/2

Рис. 1. Схема алгоритма программы

3/2

cout << "Добавить новый элемент в начало: " << endl;

cin >> newelement;

myList.push\_front(newelement);

cout << "Удалить элемент из списка. Введите значение элемента: ";

cin >> deleted;

myList.remove(deleted);

cout << "Обновленный список:" << endl;

copy(myList.begin(), myList.end(), ostream\_iterator<int>(cout, " "));

5/3

Рис. 1. Схема алгоритма программы

4/3

cout << "Отсортированный список:" << endl;

copy(myList.begin(), myList.end(), ostream\_iterator<int>(cout, " "));

myList.sort();

cout << "Список был очищен. Текущий список: ";

Конец

copy(myList.begin(), myList.end(), ostream\_iterator<int>(cout, " "));

myList.clear();

Рис. 1. Схема алгоритма программы

**Текст исходного кода программы с комментариями**

#include <iostream>

#include <list>

#include <iterator>

#include <cstdlib>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int size, element, newelement, deleted;

cout << "Введите длину списка: ";

cin >> size;

if (size < 1 || cin.fail())

cout << "Неверно. Введите число от 1." << endl;

cin.clear();

cin.ignore(cin.rdbuf()->in\_avail());

while (size<1) {

cout << "Введите длину списка: ";

cin >> size;

if (size<1 || cin.fail()) cout << "Неверно. Введите число от 1." << endl;

cin.clear();

cin.ignore(cin.rdbuf()->in\_avail());

}

list<int> myList; // объявляем пустой список

for (int i = 1; i <= size; i++) {

cout << "Введите элемент списка " << i << ": ";

cin >> element;

myList.push\_back(element); // добавляем в список новые элементы

}

cout << "Исходный список: ";

copy(myList.begin(), myList.end(), ostream\_iterator<int>(cout, " ")); // вывод на экран элементов списка

cout << endl;

cout << "Добавить новый элемент в конец: " << endl;

cin >> newelement;

myList.push\_back(newelement);

cout << "Добавить новый элемент в начало: " << endl;

cin >> newelement;

myList.push\_front(newelement);

cout << "Удалить элемент из списка. Введите значение элемента: ";

cin >> deleted;

myList.remove(deleted);

cout << "Обновленный список:" << endl;

copy(myList.begin(), myList.end(), ostream\_iterator<int>(cout, " "));

cout << endl;

cout << "Отсортированный список:" << endl;

myList.sort();

copy(myList.begin(), myList.end(), ostream\_iterator<int>(cout, " "));

cout << endl;

cout << "Список был очищен. Текущий список: ";

myList.clear();

copy(myList.begin(), myList.end(), ostream\_iterator<int>(cout, " "));

cout << endl;

system("Pause");

return 0;

}

**Контрольный прогон программы**

Тест №1 – ввод списка и работа с ним.

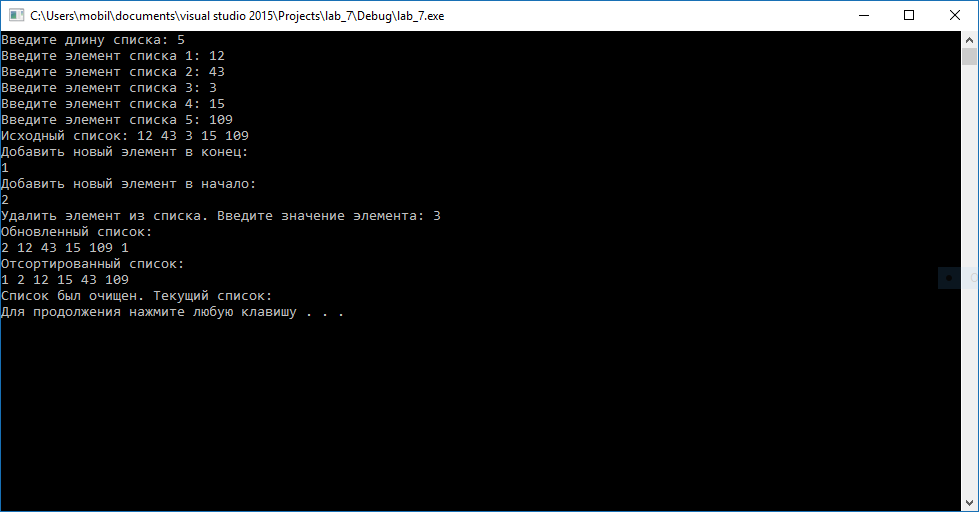


Рис. 2. Результат теста №1

**Выводы**

Данная программа позволила мне освоить работу со списками в С++. Данный раздел для меня абсолютно новый, оттого выше его цена.

**Список используемых информационных источников**

1. Павловская Т.А. С/С++. Программирование на языке высокого уровня- Санкт-Петербург. Питер, 2006.

2. Интернет-ресурс: http://cppstudio.com/post/8482/

3. Интернет-ресурс: http://www.cplusplus.com/reference/list/list/clear/

4. Единая система программной документации (ЕСПД) ГОСТ 19.701-90

5. Сыромятников В.П. «Программирование», лекции, МТУ МИРЭА, 2016/2017.