МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Нижегородский государственный университет им. Н.И. Лобачевского»**

**Национальный исследовательский университет**

**Институт информационных технологий, математики и механики**

**Кафедра математического обеспечения и суперкомпьютерных технологий**

**ЛАБОРАТОРНАЯ РАБОТА**

**«Связный список»**

**Выполнил:**студент группы 381706-2

Гущин Александр Владимирович

\_\_\_\_\_\_\_\_\_\_\_  Подпись

**Руководитель:**

Ассистент кафедры МОСТ

Лебедев Илья Геннадьевич

\_\_\_\_\_\_\_\_\_\_\_  Подпись

Нижний Новгород

2018

Оглавление

[1.Введение 3](#_Toc4704470)

[2.Постановка задачи 4](#_Toc4704471)

[3.Руководство пользователя 5](#_Toc4704472)

[4.Руководство программиста 6](#_Toc4704473)

[4.1.Описание структуры программы 6](#_Toc4704474)

[4.2.Описание структур данных 6](#_Toc4704475)

[4.3.Описание алгоритмов 7](#_Toc4704476)

[5.Эксперименты 8](#_Toc4704477)

[6.Заключение 9](#_Toc4704478)

[7.Литература 10](#_Toc4704479)

1.Введение

Связный список — базовая динамическая структура данных в информатике, состоящая из узлов, каждый из которых содержит как собственно данные, так и одну ссылку на следующий узел списка. Принципиальным преимуществом перед массивом является структурная гибкость: порядок элементов связного списка может не совпадать с порядком расположения элементов данных в памяти компьютера, а порядок обхода списка всегда явно задаётся его внутренними связями.

2.Постановка задачи

Необходимо написать реализацию списка со всеми сопутствующими конструкторами, методами (положить элемент в начало, в конец, на указанную позицию, взять элемент из указанной позиции). Данная реализация представляет собой связный список.

3.Руководство пользователя

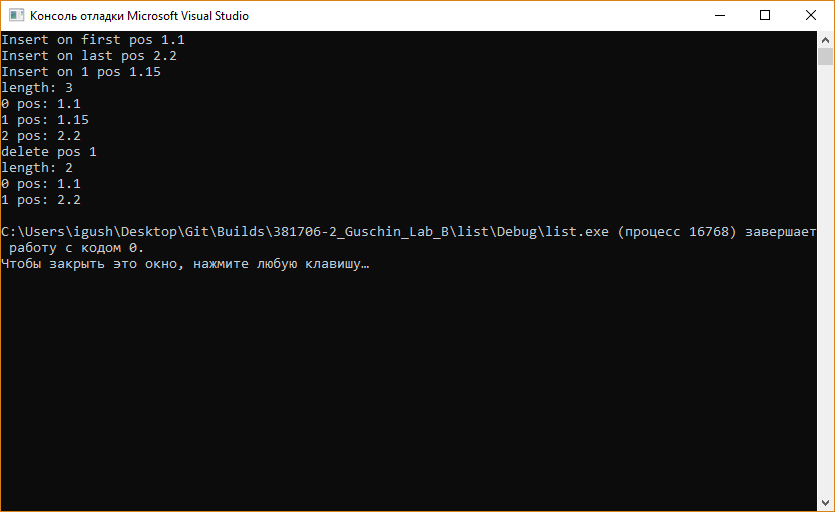


Рис.1 Пример использования.

Демонстрация методов класса(Создание, вставка в начало, в конец, на указанную позицию, удаление элемента).

4.Руководство программиста

4.1.Описание структуры программы

Программа состоит из таких модулей как:

List.h – список.

TLink – элемент списка.

4.2.Описание структур данных

1. T value – значение элемента.
2. TLink<T>\* p – указатель на следующий элемент.
3. TLink() – конструктор по умолчанию.
4. TLink(const T& a, TLink<T>\* \_p) – конструктор с параметрами. Принимает значение элемента и указатель на следующий элемент.
5. TLink(const TLink<T>& A) – конструктор копирования.
6. void SetValue(const T& a) – ставит новое значение элемента.
7. T GetValue() – возвращает значение элемента.
8. TLink<T>\* GetNextLink() – возвращает указатель на следующий элемент списка.
9. void setP(TLink<T>\* \_p) – ставит новый указатель.
10. TLink<T>\* firstLink – первый элемент в списке.
11. TLink<T>\* lastLink – последний элемент в списке.
12. int len – длинна.
13. List() - конструктор по умолчанию.
14. ~List() – деструктор.
15. T GetValue(int pos) const – возвращает значение с позиции pos.
16. bool IsEmpty() const – возвращает 1 если список пуст.
17. int GetLen() const – возвращает длину списка.
18. void InsFirst(const T& a) – вставляет элемент а в начало списка.
19. void InsCustom(const T& a, int pos) - вставляет элемент а на позицию pos.
20. void InsLast(const T& a) – вставляет элемент в конец.
21. void DelFirst() – удаляет первый элемент в списке.
22. void DelCustom(int pos) – удаляет элемент на позиции pos.

4.3.Описание алгоритмов

DelCustom(int pos)

Проверка на попадание в промежуток.

Если pos == 0, то удаляем первый элемент.

Если pos != последнему элементу, то ищем предшествующий элемент, после чего меняем ему указатель и удаляем нужный элемент, уменьшаем длину на 1.

Если pos == последнему элементу, то ищем предшествующий элемент, после чего меняем ему указатель на NULL, после чего удаляем элемент и уменьшаем длину на 1.

5.Эксперименты

Все методы, связанные с указыванием позиции, имеют сложность О(n).



Рис.1 Пример удаления из списка.

Характеристики компьютера: Amd fx-8350, 8gb оперативной памяти

6.Заключение

Благодаря этой работе у меня появился опыт взаимодействия со списком, улучшил понимание основ программирования базовых структур данных, осознал важность списка в жизни человека. Список значительно упрощает работу с информацией и помогают решать многие практические задания.

7.Литература

1. Википедия https://ru.wikipedia.org/wiki/Список\_(информатика) (Дата обращения 30.12.2018)
2. Википедия <https://ru.wikipedia.org/wiki/Связный_список> (Дата обращения 30.12.2018)