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**

**ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**

**«Структура хранения данных: стек на списках»**

**Выполнил:** студент группы 381706-1

Голубева А. С.

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

аспирант каф. МОСТ ИИТММ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лебедев И.Г.

Нижний Новгород

2018.

Содержание

[**1.** **Введение** 3](#_Toc560660)

[**2.** **Постановка задачи** 4](#_Toc560661)

[**3.** **Руководство пользователя** 5](#_Toc560662)

[**4.** **Руководство программиста** 6](#_Toc560663)

[**4.1** **Описание структуры программы** 6](#_Toc560664)

[**4.2** **Описание структур данных** 6](#_Toc560665)

[**4.3** **Описание алгоритмов** 7](#_Toc560666)

[**5.** **Заключение** 8](#_Toc560667)

[**6.** **Литература** 9](#_Toc560668)

# **Введение**

Стек – это последовательный список с переменной длиной, в котором включение и исключение элементов происходит только с одной стороны – с его вершины. Стек функционирует по принципу: последним пришел – первым ушел, Last In – First Out (LIFO).

Программная реализация стека возможна на основе различных структур данных, например, с использованием статических или динамических одномерных массивов (векторов) и линейных списков.

Стек на списках — структура данных, представляющая собой упорядоченный набор элементов одного типа, фиксированного количества, связанных между собой последовательно посредством указателей, организованных по принципу LIFO. Каждый элемент стека на списках имеет указатель на следующий элемент. Последний элемент стека на списках указывает на NULL. На первый элемент списка указателя нет.

Целью данной лабораторной работы является реализация стека на списках.

# **Постановка задачи**

Выполнение работы предполагает решение следующих задач:

1. Реализация класса стека на списках TStackList.
2. Реализация класса TException для обработки исключений.
3. Реализация тестов и обеспечение их работоспособности.
4. Пример использования и обеспечение его работоспособности.

# **Руководство пользователя**

Рассмотрим пример работы программы.

При запуске программы требуется ввести размер стека и положить в него элементы. Затем стек выводится на экран. Далее демонстрируется работа метода, позволяющего получить элемент из стека.

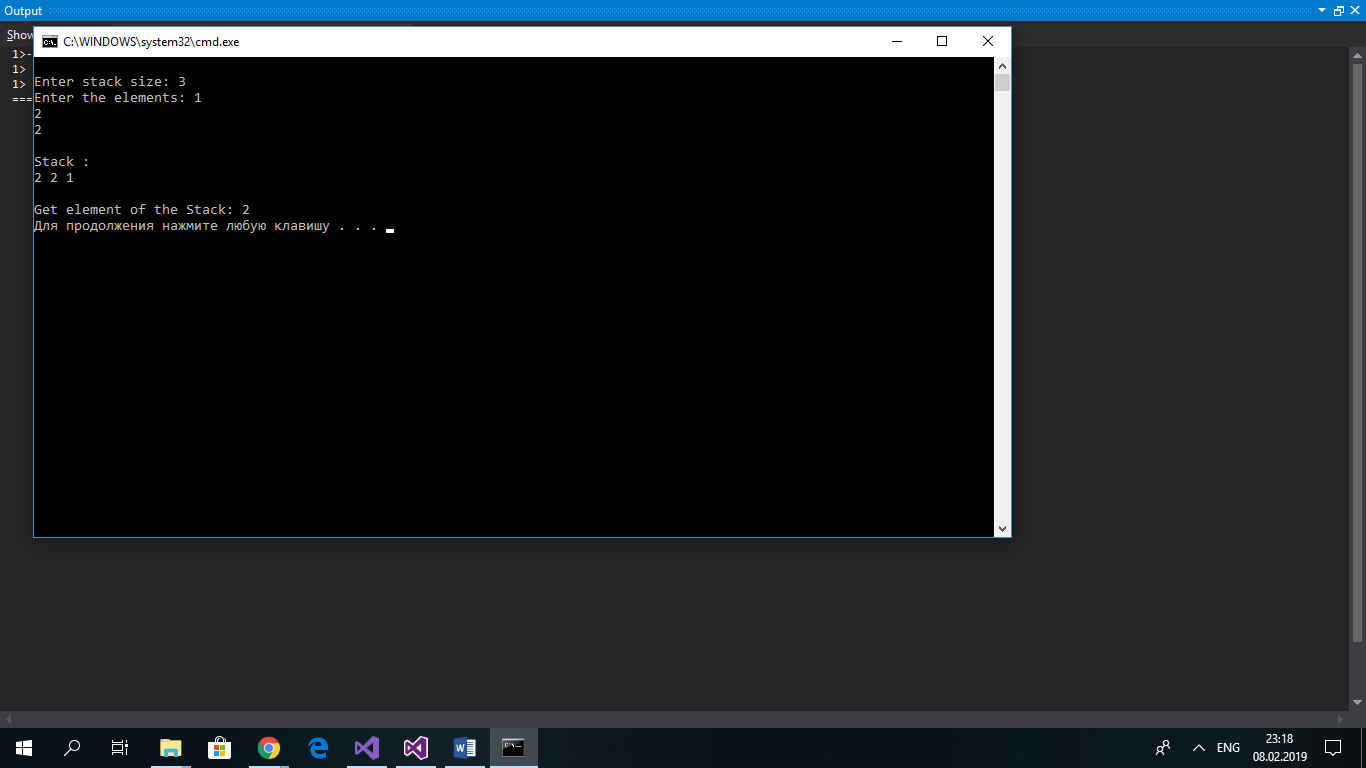


Рисунок 1. Пример работы программы

# **Руководство программиста**

## **Описание структуры программы**

Программа состоит из модулей:

* Модуль StackListLib – содержит заголовочный файл StackList.h, в котором определен интерфейс класса TStackList и реализованы его методы, и файл Stack.cpp;
* Модуль StackList – содержит файл StackList\_main.cpp, в котором реализован пример использования класса TStackList;
* Модуль StackListTest – содержит файл StackList\_Test.cpp, в котором реализованы тесты для класса TStackList;
* Модуль Exception – содержит заголовочный файл Exception.h, в котором реализован класс исключений, и файл Exception.cpp.

## **Описание структур данных**

**Класс TSatackList**

*TStackList* - шаблонный класс, является наследником класса TList.

Protected:

int size\_st – максимальный размер стека.

Public:

TStackList(int \_size = 10) – конструктор.

TStackList(TStackList<T> &A) – конструктор копирования.

~TStackList() – деструктор.

void Print() – вывод стека на списках на экран.

void Put(T A) – добавить новый элемент на вершину стека.

T Get() –получить значение элемента с вершины стека и удалить его.

bool IsFull() – проверка на полноту.

bool IsEmpty() – проверка на пустоту.

int GetSize() – получить количество элементов в стеке.

int GetMaxSize() – получить максимальный размер стека.

## **Описание алгоритмов**

**Добавление элемента в стек**

При добавлении элемента в стек, он размещается на вершине стека. Так как мы используем списки, вершина будет первым элементом списка.

**Изъятие элемента из стека.**

Аналогично, при изъятии элемента из стека берется элемент, находящийся на вершине стека. Следовательно, сложность добавления и извлечения элементов из начала списка будет О(1).

# **Заключение**

В результате лабораторной работы разработан и реализован стек на списках, также были реализованы операции для работы со стеком.

Успешно выполнены тесты, проверяющие работоспособность методов класса TStackList. Приведен пример, демонстрирующий работу со стеком.

# **Литература**

* 1. Лабораторный практикум. Составители: Барышева И.В., Мееров И.Б., Сысоев А.В., Шестакова Н.В. Под редакцией Гергеля В.П. Учебно-методическое пособие. – Нижний Новгород: Нижегородский госуниверситет, 2017. – 105с.

URL: <http://www.unn.ru/books/met_files/Pract_ADS.pdf>

(Дата обращения: 08.02.2019)

* 1. Университет ИТМО: Викиконспекты.

URL: http://neerc.ifmo.ru/wiki/index.php?title=Стек

(Дата обращения: 08.02.2019)