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**

**«Реализация стека на основе одномерного**

**динамического массива»**

**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](#_Toc533027637)

[**2.** **Постановка задачи** 4](#_Toc533027638)

[**3.** **Руководство пользователя** 4](#_Toc533027639)

[**4.** **Руководство программиста** 5](#_Toc533027640)

[**1.** **Описание структуры программы** 5](#_Toc533027641)

[**2.** **Описание структур данных** 6](#_Toc533027642)

[**3.** **Описание алгоритмов** 7](#_Toc533027643)

[**5.** **Заключение** 8](#_Toc533027644)

[**6.** **Литература** 9](#_Toc533027645)

# **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.

Целью данной лабораторной работы является реализация стека на основе динамического одномерного массива.

# **Постановка задачи**

Выполнение работы предполагает решение следующих задач:

1. Реализация класса TStack.
2. Реализация класса TException для обработки исключений
3. Реализация тестов и обеспечение их работоспособности.
4. Пример использования и обеспечение его работоспособности.

**Стек**

Для работы со стеком предлагается реализовать следующие операции:

* Метод Put – добавить элемент;
* Метод Get – удалить элемент;
* Метод IsEmpty – проверить стек на пустоту;
* Метод IsFull – проверить стек на полноту;
* Метод PrintStack – вывести стек на экран.

# **Руководство пользователя**

Пример использования стека:

* При запуске программы пользователю предлагается ввести число элементов в стеке, затем создается стек заданного размера с элементами типа int;
* Полученный стек выводится на экран;
* Далее требуется ввести элементы стека;
* Извлекается один элемент и стек снова выводится на экран.

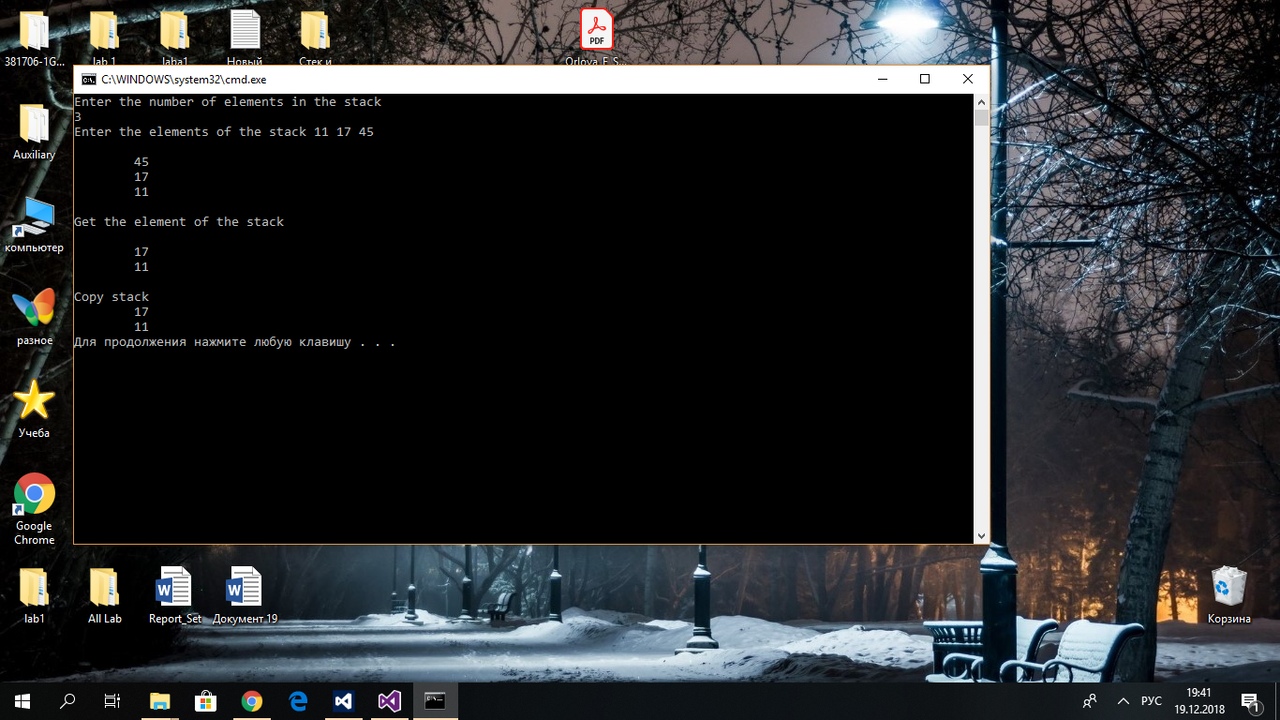


Рисунок 1. Пример работы стека

# **Руководство программиста**

## **Описание структуры программы**

Программа состоит из следующих модулей:

* Модуль StackLib – содержит заголовочный файл Stack.h, в котором определяется интерфейс класса TStack и реализуются его методы. (Класс TStack шаблонный), и файл Stack.cpp;
* Модуль Stack – содержит файл main.cpp, в котором реализован пример использования класса TStack;
* Модуль StackTest – содержит файл test\_tstack.cpp, в котором реализованы тесты для класса TStack;
* Модуль Exception – содержит заголовочный файл Exception.h, в котором реализован класс исключений, и файл Exception.cpp.

## **Описание структур данных**

**Класс TException**

**Private:**

Поле string message – переменная, которая хранит сообщение об ошибке в виде строки.

**Public:**

TException(string \_message)– конструктор;

void Print() – вывод сообщения об ошибке на экран.

**Класс TStack**

Класс является шаблонным.

**Protected:**

int size; - размер стека;

int top; - элемент, расположенный на вершине стека;

T\* mas; - массив элементов стека;

**Public:**

TStack(int n = 0) - конструктор инициализации;

TStack(TStack<T> &st) - конструктор копирования;

T Get() - забирает элемент из стека (если стек не пуст);

void Put(T ф) - добавляет элемент в стек (если стек не полон);

bool IsFull()- проверяет полон ли стек;

bool IsEmpty() - проверяет пуст ли стек;

void PrintStack() - выводит стек на экран.

## **Описание алгоритмов**

Рассмотрим работу методов Put и Get.

1. Put(T a) – добавить элемент в стек.

При добавлении элемента в стек, записываем элемент в ячейку динамического массива, на которую указывает параметр *top*. Затем увеличиваем значение *top* на 1.

1. T Get() – забрать элемент.

При удалении элемента из стека уменьшаем значение *top* на 1. Затем возвращаем значение из динамического массива по индексу top.

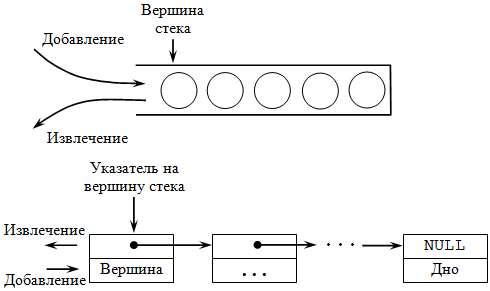


Рисунок 2. Добавление и удаление элемента из стека.

# **Заключение**

В результате лабораторной работы разработан и реализован стек на основе динамического одномерного массива, также реализованы операции для работы со стеком. Успешно выполнены тесты, проверяющие работоспособность методов класса TStack. 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: 19.12.2018)

* 1. Национальный открытый университет «Интуит». Курс «Основы программирования». Лекция 11: Структуры данных: общее понятие, реализация. Простейшие структуры данных: очередь, стек. Использование стека и обратная польская запись.

URL: <https://www.intuit.ru/studies/courses/2193/67/lecture/1980?page=3>

(Дата обращения: 19.12.2018)