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**

**«Структура хранения данных: Стек»**

**Выполнил:** студент группы 381706-2

Паузин Леонид Павлович

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Руководитель:**

Ассистент кафедры МОСТ

Лебедев Илья Геннадьевич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

Нижний Новгород

2018

Содержание

[1. Введение 3](#_Toc534406586)

[2. Постановка задачи 4](#_Toc534406587)

[3.Руководство пользователя 5](#_Toc534406588)

[4.Руководство программиста 6](#_Toc534406589)

[4.1 Описание структуры программы 6](#_Toc534406590)

[4.2 Описание структур данных 6](#_Toc534406591)

[4.3 Описание алгоритмов 6](#_Toc534406592)

[5. Заключение 8](#_Toc534406593)

[6.Литература 9](#_Toc534406594)

# 1. Введение

Лабораторная работа направлена на практическое освоение динамической структуры данных стек.

Стек (англ. stack ‒ стопка) ‒ абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO (англ. last in ‒ first out, «последним пришёл ‒ первым вышел»). Это значит, что мы будем иметь доступ только к последнему добавленному элементу.

В стеке мы не можем получить доступ к произвольному элементу. Мы можем только добавлять или удалять элементы с помощью специальных методов



Рисунок 1. Организация стека в виде одномерного упорядоченного по адресам массива

# 2. Постановка задачи

Реализация класса структуры данных ‒ стек и выполнение следующих операций:

* добавить элемент
* извлечь элемент
* проверка полноты
* проверка пустоты

# 3.Руководство пользователя

Данная программа предназначена для тестирования динамической структуры стек. В список поочерёдно загружаются 12 элементов, а затем они извлекаются из стека.

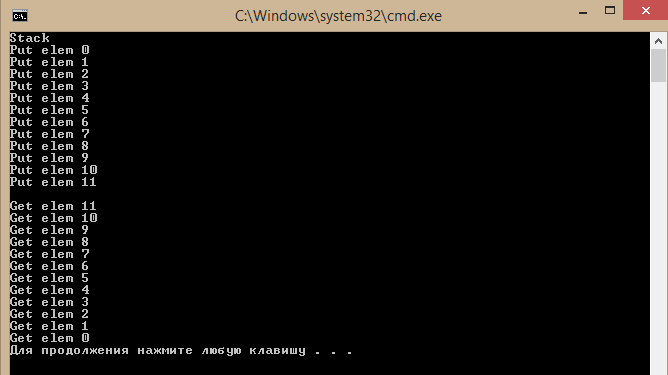


Рисунок 2. Демонстрация работы класса стек

# 4.Руководство программиста

## 4.1 Описание структуры программы

Модульная структура программы:

* Модуль stacklib (Stack.h, Stack.cpp) ‒ реализация класса стек
* Модуль stack(main.cpp) ‒ реализация программы для тестирования динамической структуры стек
* Модуль stacktest ‒ тестирование

## 4.2 Описание структур данных

Поля:

mem ‒ память для элементов стека

top ‒ индекс последнего элемента

memSize ‒ размер памяти, выделяемой под стек

Методы:

int IsFull() ‒ проверка переполнения

int IsEmpty() ‒ проверка пустоты

void Put(const int Val) ‒ добавить значение

ValType Get() ‒ извлечь значение

int GetCount() ‒ число элементов в стеке

## 4.3 Описание алгоритмов

**Добавление элемента в стек**

При добавлении элемента в стек, размещаем его в первую свободную ячейку массива, выделенного под хранение данных (если она имеется). На эту ячейку указывает top ‒ вершина стека. Затем значение top увеличиваем на 1.

**Удаление элемента из стека**

При удалении элемента из стека уменьшаем значение top на 1 ‒ спускаемся к последнему существующему элементу и возвращаем его значение (если стек не пуст). Само значение не затирается, т.к. оно будет перезаписано при следующем добавлении другого элемента

# 5. Заключение

В результате лабораторной работы был реализован класс TStack. Была разработана библиотека stacklib, реализующая шаблонный класс стека TStack.

# 6.Литература

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.
2. Википедия: свободная электронная энциклопедия ‒ Стек: <https://ru.wikipedia.org/wiki/Стек> (дата обращения 2.01.2019)
3. Статья «Стек» на сайте: <https://prog-cpp.ru/data-stack/> (дата обращения 2.01.2019)