

**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 данных: стек»**

Выполнил:  
студент группы 361706-1  
Резанцев Сергей Алексеевич  
\_\_\_\_\_ Подпись

Научный руководитель:  
ассистент каф. МОСТ ИИТММ  
\_\_\_\_\_ Лебедев И. Г.

Нижний Новгород  
2018 г.

<b>Содержание</b>	
<b>Введение</b>	<b>3</b>
<b>Постановка задачи</b>	<b>4</b>
<b>Руководство пользователя</b>	<b>5</b>
<b>Руководство программиста</b>	<b>6</b>
Описание структуры программы	6
Описание структур данных	6
Описание алгоритмов	7
<b>Заключение</b>	<b>8</b>
<b>Литература</b>	<b>9</b>

# 1. Введение

Целью данной лабораторной работы является реализовать стек, и для создания этой программы понадобится создать класс с шаблоном и различными функциями.

Стек — абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO (last in — first out, «последним пришёл — первым вышел»).

Область применения:

Программный вид стека используется для обхода структур данных, например, дерево или граф. При использовании рекурсивных функций также будет применяться стек, но аппаратный его вид. Кроме этих назначений, стек используется для организации стековой машины, реализующей вычисления в обратной инверсной записи.

Для отслеживания точек возврата из подпрограмм используется стек вызовов.

Арифметические сопроцессоры, программируемые микрокалькуляторы и язык Forth используют стековую модель вычислений.

Идея стека используется в стековой машине среди стековых языков программирования.

Применение стека упрощает и ускоряет работу программы, так как идет обращение к нескольким данным по одному адресу.

Цель данной лабораторной работы - реализовать класс TStack.

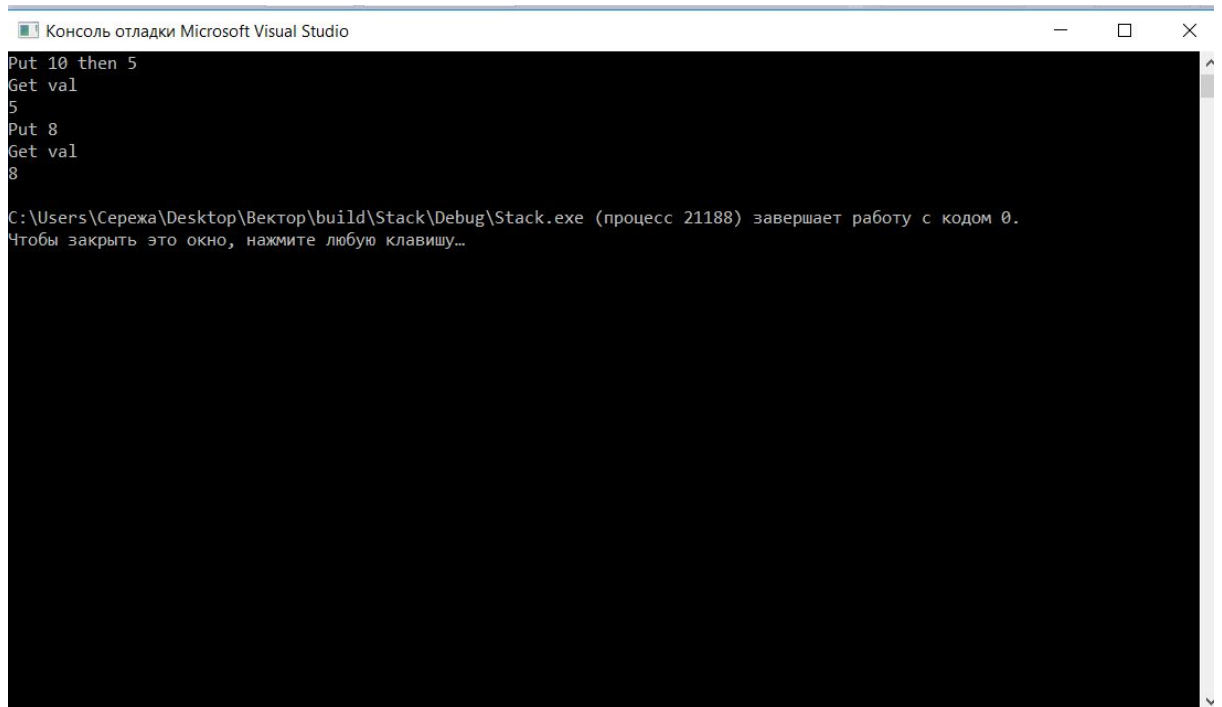
## **2. Постановка задачи**

В рамках лабораторной работы ставится задача разработки программы, в которой реализованы методы шаблонного класса TStack. Предоставить пример использования и обеспечить работоспособность тестов, покрывающих все методы класса TStack.

### 3. Руководство пользователя

Чтобы начать работу с программой запустите приложение Stack.

На экране появится следующее:

A screenshot of the 'Консоль отладки Microsoft Visual Studio' (Microsoft Visual Studio Debug Console) window. The window has a title bar with standard Windows controls (minimize, maximize, close). The console area is black with white text. The output shows: 'Put 10 then 5', 'Get val', '5', 'Put 8', 'Get val', '8'. Below this, a message states: 'C:\Users\Сергея\Desktop\Вектор\build\Stack\Debug\Stack.exe (процесс 21188) завершает работу с кодом 0. Чтобы закрыть это окно, нажмите любую клавишу...' (C:\Users\Сергея\Desktop\Вектор\build\Stack\Debug\Stack.exe (process 21188) finishes work with code 0. To close this window, press any key...).

```
Консоль отладки Microsoft Visual Studio
Put 10 then 5
Get val
5
Put 8
Get val
8

C:\Users\Сергея\Desktop\Вектор\build\Stack\Debug\Stack.exe (процесс 21188) завершает работу с кодом 0.
Чтобы закрыть это окно, нажмите любую клавишу...
```

Результат программы, выведенный на консоль.

Затем программа завершится.

## 4. Руководство программиста

### 4.1. Описание структуры программы

Для реализации алгоритмов будет использован класс TStack.

Лабораторная работа состоит из следующих модулей:

StackLib

Библиотека, содержащая заголовочный файл TStack.h, в котором содержится класс TStack и реализация его методов, и файл TStack.cpp

Stack

Пример использования программы.

test

В файле test\_stack.cpp прописаны тесты, покрывающие каждый метод класса TStack.

### 4.2. Описание структур данных

В программе описан шаблонный класс TStack:

В нем 3 поля (protected):

int size; //размер стека

int r; //индекс элемента, расположенного на вершине стека

T\* m; //стек

И реализованы следующие функции:

TStack(int n); // конструктор инициализации

TStack(TStack<T> &A); // конструктор копирования

void Put(T A); // добавляет элемент на вершину стека

T Get(); // возвращает элемент, расположенный на вершине стека

bool IsFull(); // проверка на полноту

bool IsEmpty(); // проверка на пустоту

### 4.3. Описание алгоритмов

Описание алгоритмов класса TStack:

- 1)Get() - проверяет на пустоту, если стек не пуст, возвращает значение по индексу вершины стека, перемещает указатель вершины стека и уменьшает кол-во элементов.
- 2)Put(T A) - проверяет на полноту, если стек не полон, при добавлении элемента в стек перемещает указатель его вершины, добавляет элемент в соответствующую позицию массива и увеличивает кол-во элементов.
- 3)IsFull() - проверяет на полноту, если полон, то - true, иначе false.
- 4)IsEmpty() - проверяет на пустоту, если пуст, то - true, иначе false.

## **5. Заключение**

В данном курсовом проекте при разработке программы был реализован и рассмотрен стек. Классы оказались очень полезными и удобными и упростили работу с программой. Также была закреплена техника составления тестов на базе GoogleTest.



## 6. Литература

1. Васильев А.Н. Самоучитель С++ с примерами и задачами. -СПб.: Наука и Техника, 2016. -480с.
2. Т. А. Павловская С/С++ Программирование на языке высокого уровня. - СПб.:Питер, 2011. - 461 с.
- 3.Крапенко С. Н. и др. Методы объектно-ориентированного программирования.  
<http://e-learning.unn.ru/course/view.php?id=251>.
- 4.Страуструп. Б. Курс «Язык программирование С++ для профессионалов»  
<http://www.intuit.ru/studies/courses/98/98/info>
- 5.Гергель В.П. Методические материалы по курсу “Методы программирования 2”:  
[<http://www.itmm.unn.ru/files/2018/10/Primer-1.1.-Struktury-hraneniya-mnozhestva.pdf>],  
2015.