

**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: очередь»**

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>9</b>
<b>Литература</b>	<b>10</b>

# 1. Введение

Целью данной лабораторной работы является реализация очередей с помощью стека, и для реализации этой программы понадобится создать классы с шаблонами, различными функциями.

Очередь — это структура данных, которая построена по принципу LIFO (last in — last out: последним пришел — последним вышел).

Очередь в программировании используется, как и в реальной жизни, когда нужно совершить какие-то действия в порядке их поступления, выполнив их последовательно. Примером может служить организация событий в Windows. Когда пользователь оказывает какое-то действие на приложение, то в приложении не вызывается соответствующая процедура (ведь в этот момент приложение может совершать другие действия), а ему присылается сообщение, содержащее информацию о совершенном действии, это сообщение ставится в очередь, и только когда будут обработаны сообщения, пришедшие ранее, приложение выполнит необходимое действие.

Цель данной лабораторной работы - реализовать класс TQueue.

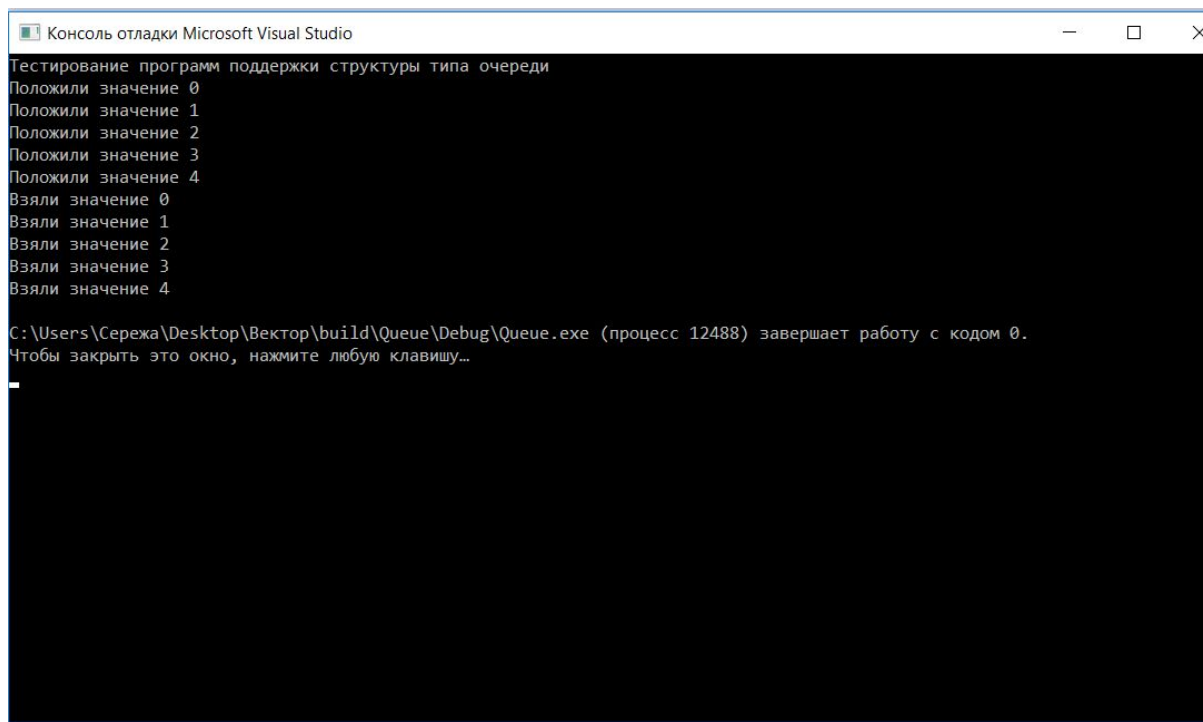
## **2. Постановка задачи**

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

### 3. Руководство пользователя

Чтобы начать работу с программой запустите приложение Queue.

На экране появится следующее:



```
Консоль отладки Microsoft Visual Studio
Тестирование программ поддержки структуры типа очереди
Положили значение 0
Положили значение 1
Положили значение 2
Положили значение 3
Положили значение 4
Взяли значение 0
Взяли значение 1
Взяли значение 2
Взяли значение 3
Взяли значение 4
C:\Users\Сергея\Desktop\Вектор\build\Queue\Debug\Queue.exe (процесс 12488) завершает работу с кодом 0.
Чтобы закрыть это окно, нажмите любую клавишу...
-
```

*Рисунок 1. Результат программы, выведенный на консоль.*

Затем программа завершится.

## 4. Руководство программиста

### 4.1. Описание структуры программы

Для реализации алгоритмов будут использованы классы TStack и TQueue.

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

QueueLib

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

Queue

Пример использования программы.

test

В файле test\_queue.cpp прописаны тесты, покрывающие каждый метод класса TQueue.

### 4.2. Описание структур данных

В программе описаны классы:

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

Шаблонный класс TQueue:

В нем 2 поля:

int start; // начало очереди

int count; // кол-во элементов в очереди

И реализованы следующие функции:

TQueue(int n); // конструктор инициализации

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

void Put(T A); // положить элемент в конец очереди

T Get(); // взять элемент из начала очереди

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.

Описание алгоритмов класса TQueue:

- 1)Get() - проверяет на пустоту, если очередь не пуста, возвращает элемент из начала очереди и уменьшает кол-во элементов.
- 2)Put(T A) - проверяет на полноту, если очередь не заполнена, добавляет элемент в конец очереди 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 были рассмотрены очереди и их реализация с помощью стеков. Классы и наследование 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 C++ с примерами и задачами. -СПб.: Наука и Техника, 2016. -480с.
2. Т. А. Павловская C/C++ 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 C++ 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.