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 класса очередь»**

**Выполнил:** студент группы 381706-2

Банденков Даниил Викторович

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

ассистент каф. МОСТ ИИТММ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лебедев И.Г

Нижний Новгород

2018.

# Содержание

[Содержание 2](#_Toc534328769)

[1 Введение 3](#_Toc534328770)

[2 Постановка задачи 4](#_Toc534328771)

[3 Руководство пользователя 5](#_Toc534328772)

[4 Руководство программиста 6](#_Toc534328773)

[4.1 Описание структуры программы 6](#_Toc534328774)

[4.2 Описание структур данных 6](#_Toc534328775)

[4.3 Описание алгоритмов 7](#_Toc534328776)

[5 Заключение 8](#_Toc534328777)

[6 Литература 9](#_Toc534328778)

# Введение

Для решения задач многие программисты используют структуру данных — очередь. Очередь — это структура данных, которая построена по принципу «первый пришёл — первый вышел» (FIFO, англ. first in, first out)

В очереди, если вы добавите элемент, который вошел самый первый, то он выйдет тоже самым первым. Получается, если вы добавите 4 элемента, то первый добавленный элемент выйдет первым. Добавление элемента (принято обозначать словом enqueue — поставить в очередь) возможно лишь в конец очереди, выборка — только из начала очереди (что принято называть словом dequeue — убрать из очереди), при этом выбранный элемент из очереди удаляется.

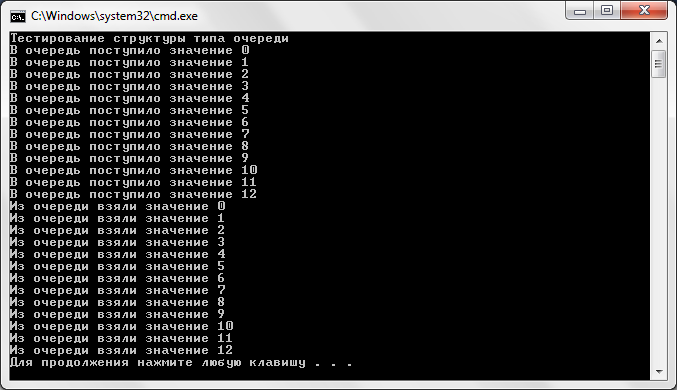
Чтобы понять принцип работы очереди вы можете представить себе магазинную очередь. И вы стоите посреди нее, чтобы вы оказались напротив кассы, сначала понадобится всех впереди стоящих людей обслужить. А вот для последнего человека в очереди нужно, чтобы кассир обслужил всех людей кроме него самого.

# Постановка задачи

В данной работе требуется создать класс очередь, в нем реализовать функции контроля пустоты и переполнения, а так же добавления и извлечения элементов.

# Руководство пользователя

Данная программа тестирует динамическую структуру-очередь. В список поочередно загружаются 13 элементов, а затем они извлекаются из очереди и выводятся в консоли (см Рис 1).

Рис 1.Пример использования программы.

# Руководство программиста

## Описание структуры программы

*В решении содержатся следующие модули:*

1. queuelib (TQueue.h, TQueue.cpp) – модуль с классом, реализующий структуру очередь.
2. queue(main.cpp)– модуль реализующий тестирование динамической структуры очередь.
3. queuetest(queue\_test.cpp)– модуль тестирование класса очереди при помощи Google C++ Testing Framework. .

Структура очередь унаследована от структуры стек, следовательно, имеет все поля и методы родительского класса.

## Описание структур данных

*Поля:*

*Наследуемые:*

mem - память для элементов структуры.

top - индекс последнего занятого в mem.

memSize — размер памяти под структуру.

*Ненаследуемые:*

li - индекс первого элемента структуры.

*Методы:*

int IsEmpty() - контроль пустоты.

int GetCount() - число элементов в очереди.

int IsFull() - контроль переполнения.

void Put(const int Val) - добавить значение.

ValType Get() - извлечь значение.

## Описание алгоритмов

*Извлечение из очереди элемента (псевдокод)*

Если:

Очередь пуста

Ошибка

Иначе:

Положить в переменную tmp mem[li]

li = ++li % memSize

Уменьшить top на 1

Вернуть tmp

*Наглядный алгоритм*

На рисунке 2 находятся 7 чисел: 2, 4, 7, 1, 4, 9, 10. Если нам понадобится их извлечь, то извлекать мы будем в таком же порядке как они находятся на рисунке.

Рисунок 2.

Так например чтобы извлечь число 4 нам понадобится сначала обслужить число 2, а потом уже и само число 4.

# Заключение

В ходе работы был создан класс очередь, в нем реализованы функции контроля пустоты и переполнения, а так же добавления и извлечения элементов.

# Литература

1. Ге ргель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.
2. <https://codelessons.ru/cplusplus/ochered-queue-v-c-realizaciya-i-chto-eto-voobshhe-takoe.html> (Дата обращения 02.01. 2019)
3. <https://prog-cpp.ru/data-queue/> (Дата обращения 02.01. 2019)