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 хранения очереди»**

**Выполнил:** студент группы 381706-1

Остапович Денис Евгеньевич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

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

Нижний Новгород

2018.

Содержание

[1. Введение 3](#_Toc533083407)

[2. Постановка задачи 4](#_Toc533083408)

[3. Руководство пользователя 5](#_Toc533083409)

[4. Руководство программиста 6](#_Toc533083410)

[4.1. Описание структуры программы 6](#_Toc533083411)

[4.2. Описание структур данных 6](#_Toc533083412)

[4.3. Описание алгоритмов 6](#_Toc533083413)

[5. Заключение 8](#_Toc533083414)

[6. Литература 9](#_Toc533083415)

# Введение

**Очередь** - структура данных, представляющая собой упорядоченный набор элементов, с дисциплиной доступа к элементам «первый пришёл — первый вышел» (англ. *first-in, first-out — FIFO*). 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.

Очередь применяется в программировании, когда действия необходимо выполнять последовательно.

**Цель лабораторной работы** – разработать структуру хранения очереди

# Постановка задачи

Выполнение работы предполагает решение следующих задач:

1. Описать класс очереди – TQueue.
2. Реализовать пример использования программного решения, демонстрирующее работу класса TQueue.
3. Создать набор автоматических тестов с использованием Google C++ Testing Framework.

# Руководство пользователя

Рассмотрим пример использования класса TQueue.

После запуска программы пользователю предлагается ввести размер очереди . Затем создается очередь и полностью заполняется натуральными числами от 0 до . Каждое новое добавленное в очередь число выводится на экран.

После этого, пока очередь не пуста, из нее извлекаются все элементы и каждый следующий извлечённый элемент выводится на экран. Программа будет завершена, когда в очереди не останется элементов для извлечения.

# Руководство программиста

## Описание структуры программы

Программа состоит из модулей:

* Модуль Queue. Содержит пример использования очереди. Реализация в файле *main.cpp.*
* Модуль QueueLib – статическая библиотека. Содержит файл queue.h, в котором реализован шаблонный класс TQueue*.*
* Модуль QueueTest. Содержит 10 тестов на Google C++ Testing Framework в файле *queuetest.cpp.*

## Описание структур данных

#### 4.2.1 Класс TQueue

Класс *TQueue* является шаблонным классом, причем он наследуется от класса TStack. В нем определены два поля со спецификатором доступа protected:

* *int start* –начало очереди.
* *int count* – количество элементов, очереди.

Далее в публичной зоне (public):

1. Прописаны 2 конструктора и деструктор:

* *TQueue(int n = 0)* – конструктор по умолчанию.
* *TQueue(TQueue<T> &q)* – конструктор копирования.
* *virtual ~TQueue()* – деструктор.
* *void Push(T A)* – добавить новый элемент A в конец очереди.
* *T Pop()* – изъять с удалением элемент находящийся вначале очереди.
* *bool IsFull()* – проверка очереди на полноту.
* *bool IsEmpty()* – проверка очереди на пустоту.

## Описание алгоритмов

**Добавление элемента в очередь.**

При добавлении размещаем элемент в свободную ячейку массива на которую указывает *top* – вершина стека. Значение *top* рассчитывается по следующей формуле:

|  |  |
| --- | --- |
|  |  |

**Удаление элемента из очереди.**

При изъятии элемента из очереди, возвращается элемент с индексом начала очереди *start*. S*tart* рассчитывается по формуле:

|  |  |
| --- | --- |
|  |  |

Значения изъятого элемента не удаляется, т.к. оно будет перезаписано при следующем добавлении другого элемента.

# Заключение

В ходе лабораторной работы была разработана библиотека QueueLib, реализован шаблонный класс стека TQueue. В ней реализованы методы работы с очередью описанные в разделе «Структуры данных». Написаны тесты для данного программного проекта с использованием Google C++ Testing Framework.

# Литература

* Книги

1. A.O. Грудзинский. Методы программирования, Издательство Нижегородского госуниверситета, 2006.
2. Bjarne Stroustrup The C++ programming language

* Ссылки в Internet

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2»: [http://www.itmm.unn.ru/files/2018/10/Primer-1.4.-Struktury-hraneniya-ocheredi.pdf], 2015.

Википедия: [https://ru.wikipedia.org/wiki/Очередь\_(программирование)]