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

Полетуева Анастасия Николаевна

\_\_\_\_\_\_\_\_\_\_\_ 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** *(англ. queue)* – это структура данных, представляющая собой упорядоченный набор элементов, с дисциплиной доступа к элементам «первый пришёл — первый вышел» (англ. *first-in, first-out — FIFO*). У очереди имеется голова и хвост. Когда элемент ставится в очередь, он занимает место в её хвосте. Из очереди всегда выводится элемент, который находится в ее голове.

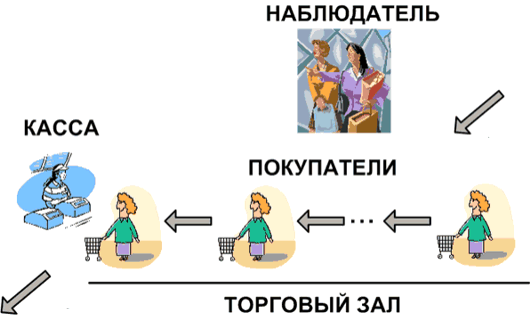
Примером очереди в реальной жизни может являться очередь в магазине. 

Рисунок 1. Пример очереди в жизни.

# Постановка задачи

В рамках лабораторной работы ставится задача разработки очереди (класс TQueue). Построение данной структуры данных рационально производить на основе стека(TStack), который был сделан ранее. Для работы с этой структурой данных будут реализованы методы:

* добавления элемента в очередь,
* извлечения элемента из очереди (с удалением),
* проверка очереди на полноту/пустоту.

Выполнение работы предполагает решение следующих задач:

1. Разработка и реализация класса стек – TQueue.
2. Разработка программы, использующую класс TQueue.
3. Реализация набора тестов, написанных с использованием GoogleTesting Framework.

# Руководство пользователя

При запуске программы c пользователя спрашивается размер очереди . Затем создается очередь и полностью заполняется натуральными числами от 0 до , и выводится на консоль. Далее, пока очередь не пуста, из нее извлекаются все элементы. И в порядке извлечения выводятся на экран.

# Руководство программиста

## Описание структуры программы

Программа состоит из следующих модулей:

* Модуль Queue. Содержит пример использования очереди. Реализация в файле main\_Queue.cpp.
* Модуль QueueLib – статическая библиотека. Содержит файл Queue.h, в котором описан интерфейс и реализация шаблонного класса TQueue.
* Модуль StackTest. Содержит 10 тестов, описанных в файле StackTest.cppи разработанных с помощью использования Google C++ Testing Framework.
* Модуль ExceptionLib – библиотека, содержащая класс исключений.

## Описание структур данных

#### 4.2.1 Класс TQueue

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

* int start – позиция вершины стека.
* int count – количество, заполненных элементов очереди.

Далее в публичной зоне 2 конструктора, 4 метода для работы со стеком (public):

* TQueue(int n = 0) – конструктор по умолчанию.
* TQueue(TQueue<T> &Q) – конструктор копирования.
* void Put(T el) – добавить новый элемент в конец очереди.
* T Get() – 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 элемента в очередь, размещаем его в свободную ячейку массива на которую указывает индекс *start* – вершина очереди. Значение *start* рассчитывается по следующей формуле:

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

Обусловлено это тем, что наша очередь - это кольцевой буфер. Увеличиваем число элементов count на один.

**Удаление элемента из очереди.**

При удалении элемента из очереди, возвращается элемент с индексом начала очереди *start*, а сам индекс *start* рассчитывается по формуле:

|  |  |
| --- | --- |
| Уменьшаем число элементов count. |  |

# Заключение

В ходе выполнения лабораторной было установлено понятие очереди. Была разработана программа, реализующая структуру хранения данных - очередь. Реализованы методы, описанные в «Описание структур данных». Программная реализация была продемонстрирована на примере. Разработаны тесты на основе Google C++ Testing Framework.

# Литература

1. Гергель В.П. Методические материалы по курсу «Методы программирования» (часть 1), 2015, -96с.
2. Викиконспекты: <http://neerc.ifmo.ru/wiki/index.php?title=Стек>
3. Википедия: свободная электронная энциклопедия: на русском языке:

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