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**

**ЛАБОРАТОРНАЯ РАБОТА**

**«Структура хранения данных: Очередь»**

**Выполнил:** студент группы 381706-2

Паузин Леонид Павлович

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Руководитель:**

Ассистент кафедры МОСТ

Лебедев Илья Геннадьевич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

Нижний Новгород

2018

Содержание

[1. Введение 3](#_Toc8268085)

[2. Постановка задачи 4](#_Toc8268086)

[3.Руководство пользователя 5](#_Toc8268087)

[4.Руководство программиста 6](#_Toc8268088)

[4.1 Описание структуры программы 6](#_Toc8268089)

[4.2 Описание структур данных 6](#_Toc8268090)

[4.3 Описание алгоритмов 7](#_Toc8268091)

[5. Заключение 8](#_Toc8268092)

[6.Литература 9](#_Toc8268093)

# 1. Введение

Лабораторная работа направлена на практическое освоение динамической структуры данных очередь

Очередью называется структура данных, из которой удаляется первым тот элемент, который был первым в очередь добавлен. То есть очередь в программировании соответствует «бытовому» понятию очереди. Очередь также называют структурой типа FIFO (first in, first out — первым пришел, первым ушел).

Очередь поддерживает те же операции, что и стек, за исключением того, что некоторые операции работают с другим концом очереди. А именно «Извлечь из очереди последний элемент» и «узнать значение последнего элемента»

Очередь в программировании используется, как и в реальной жизни, когда нужно совершить какие-то действия в порядке их поступления, выполнив их последовательно

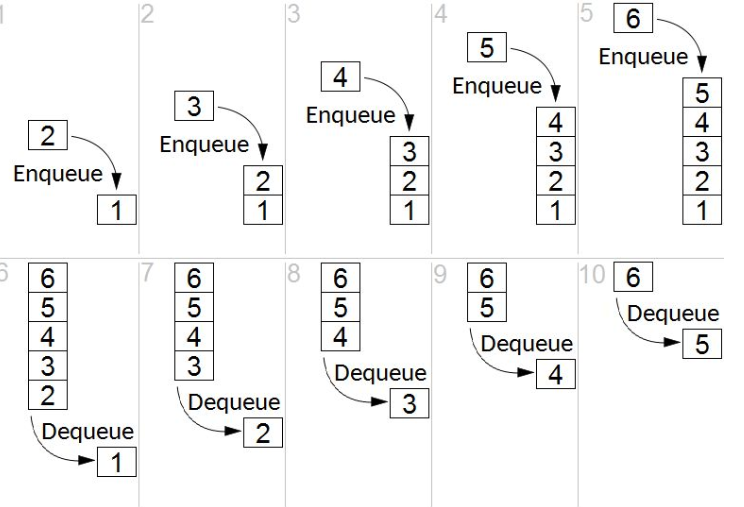


Рисунок 1. Представление очереди

# 2. Постановка задачи

Реализация класса структуры данных ‒ очередь и выполнение следующих операций:

* добавить элемент
* извлечь элемент
* проверка полноты
* проверка пустоты

# 3.Руководство пользователя

Данная программа предназначена для тестирования динамической структуры очередь. В список поочерёдно загружаются 12 элементов, а затем они извлекаются из очереди.

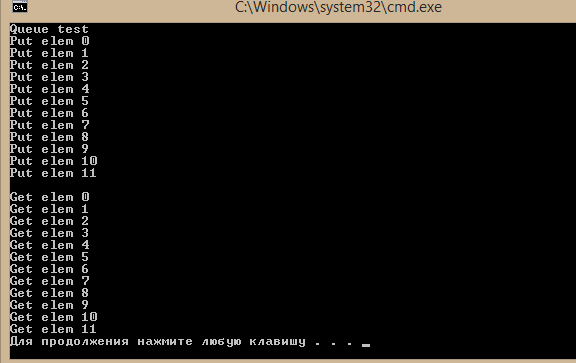


Рисунок 2. Демонстрация работы класса очередь

# 4.Руководство программиста

## 4.1 Описание структуры программы

Модульная структура программы:

* queuelib (TQueue.h, TQueue.cpp) ‒ модуль с классом, реализующий структуру очередь
* queue(main.cpp) ‒ модуль реализующий тестирование структуры очередь
* Модуль queuetest ‒ тестирование при помощи Google C++ Testing Framework

## 4.2 Описание структур данных

Структура очередь унаследована от структуры стек, а следовательно имеет поля и методы родительского класса:

Поля:

Унаследованные от стека:

mem ‒ память для элементов структуры

top ‒ индекс последнего элемента

memSize ‒ размер памяти, выделяемой под структуру

Ненаследуемые:

li ‒индекс первого элемента структуры

Методы:

int IsFull() ‒ проверка переполнения

int IsEmpty() ‒ проверка пустоты

void Put(const int Val) ‒ добавить значение

ValType Get() ‒ извлечь значение

int GetCount() ‒ число элементов в очереди

## 4.3 Описание алгоритмов

**Добавление элемента в очередь**

Проверяем не заполнена ли очередь

если очередь заполнена – выбрасываем ошибку

если она не заполнена, то размещаем добавляемый элемент в свободную ячейку на которую указывает top. Значение топ рассчитывается по следующей формуле:

top = (top+1) %memSize

Такая реализация поиска последнего элемента в очереди необходима, чтобы можно было заполнить все свободные ячейки в массиве, в котором хранится очередь.

**Удаление элемента из очереди**

проверяем не пустая ли очередь

если очередь пустая ‒ выбрасываем ошибку

если есть элементы ‒ положим в переменную tmp mem[li]

уменьшим top на 1

вернуть tmp

# 5. Заключение

В результате лабораторной работы был реализован класс TQueue. Была разработана библиотека queuelib, реализующая шаблонный класс очереди TQueue.

# 6.Литература

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.
2. <https://foxford.ru/wiki/informatika/ochered> (дата обращения 3.01.2019)
3. <https://ru.wikipedia.org/wiki/Очередь_(программирование)> (дата обращения 3.01.2019)
4. <http://www.8bytes.net/2017/10/24/struktura-dannyh-ochered-fifo/> (дата обращения 3.01.2019)