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**

**«Очередь»**

**Выполнил:**студент группы 381706-2

Гущин Александр Владимирович

\_\_\_\_\_\_\_\_\_\_\_  Подпись

**Руководитель:**

Ассистент кафедры МОСТ

Лебедев Илья Геннадьевич

\_\_\_\_\_\_\_\_\_\_\_  Подпись

Нижний Новгород

2018

Оглавление

[1.Введение 3](#_Toc534395506)

[2.Постановка задачи 4](#_Toc534395507)

[3.Руководство пользователя 5](#_Toc534395508)

[4.Руководство программиста 6](#_Toc534395509)

[4.1.Описание структуры программы 6](#_Toc534395510)

[4.2.Описание структур данных 6](#_Toc534395511)

[4.3.Описание алгоритмов 6](#_Toc534395512)

[5.Эксперименты 7](#_Toc534395513)

[6.Заключение 8](#_Toc534395514)

[7.Литература 9](#_Toc534395515)

1.Введение

Очередь - абстрактный тип данных с дисциплиной доступа к элементам «первый пришёл — первый вышел» (FIFO, англ. first in, first out). Добавление элемента возможно лишь в конец очереди, выборка — только из начала очереди , при этом выбранный элемент удаляется. Очередь на массиве имеет фиксированный размер.

2.Постановка задачи

Необходимо написать реализацию очереди со всеми сопутствующими конструкторами, методами(положить элемент, взять элемент, проверка на пустоту/полноту). Данная реализация представляет собой закольцованною очередь от стека на массиве.

3.Руководство пользователя

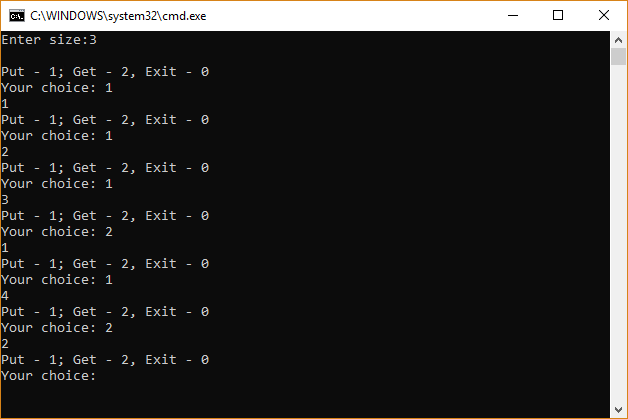


Рис.1 Пример использования.

Сначала пользователю предлагают выбрать размер очереди. После это пользователь может класть значения в очередь или забирать их.

4.Руководство программиста

4.1.Описание структуры программы

Программа состоит из таких модулей как:

* Queue.h – очередь.

4.2.Описание структур данных

Очередь наследует 3 поля и 4 неизменённых метода.

1. int st – начало очереди.
2. TQueue(int n) – конструктор с параметром. Принимает размер.
3. TQueue(TQueue<T>& A) – конструктор копирования.
4. void Put(const T s) – метод. Кладет элемент в конец очереди.
5. T Get() – метод. Берет начальный элемент из очереди.
6. T Check() – метод. Просматривает начальный элемент из очереди.

4.3.Описание алгоритмов

Void Put(const T s)

Если очередь не является полной, то если сумма st + pos строго меньше size, то в элемент массива с индексом [st + pos] кладется s, после чего увеличиваем pos на 1. Иначе в элемент массива с индексом [st + pos - size] кладется s.

T Get()

Если очередь не пуста, то уменьшаем pos на 1. После чего возвращаем элемент массива с индексом st. st увеличивается на 1. Если st равен size, то st обнуляется.

5.Эксперименты

Все методы и конструкторы имеют сложность не более O(1).

Характеристики компьютера: Amd fx-8350, 8gb оперативной памяти

6.Заключение

Благодаря этой работе у меня появился опыт взаимодействия с очередью, улучшил понимание основ программирования базовых структур данных, осознал важность очереди в жизни человека. Очередь значительно упрощает работу с информацией и помогает решать многие практические задания.

7.Литература

1. Википедия https://ru.wikipedia.org/wiki/Очередь\_(программирование) (Дата обращения 30.12.2018)
2. Сайт cpp.com.ru http://cpp.com.ru/shildt\_spr\_po\_c/22/2202.html (Дата обращения 30.12.2018)