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**

**«Стек»**

**Выполнил:**студент группы 381706-2

Гущин Александр Владимирович

\_\_\_\_\_\_\_\_\_\_\_  Подпись

**Руководитель:**

Ассистент кафедры МОСТ

Лебедев Илья Геннадьевич

\_\_\_\_\_\_\_\_\_\_\_  Подпись

Нижний Новгород

2018

Оглавление

[1.Введение 3](#_Toc534393212)

[2.Постановка задачи 4](#_Toc534393213)

[3.Руководство пользователя 5](#_Toc534393214)

[4.Руководство программиста 6](#_Toc534393215)

[4.1.Описание структуры программы 6](#_Toc534393216)

[4.2.Описание структур данных 6](#_Toc534393217)

[4.3.Описание алгоритмов 6](#_Toc534393218)

[5.Эксперименты 7](#_Toc534393219)

[6.Заключение 8](#_Toc534393220)

[7.Литература 9](#_Toc534393221)

1.Введение

Стек (англ. stack — стопка) — абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO (англ. last in — first out, «последним пришёл — первым вышел»). Стек представляет возможность реализации некоторых алгоритмов без использования рекурсии.

2.Постановка задачи

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

3.Руководство пользователя

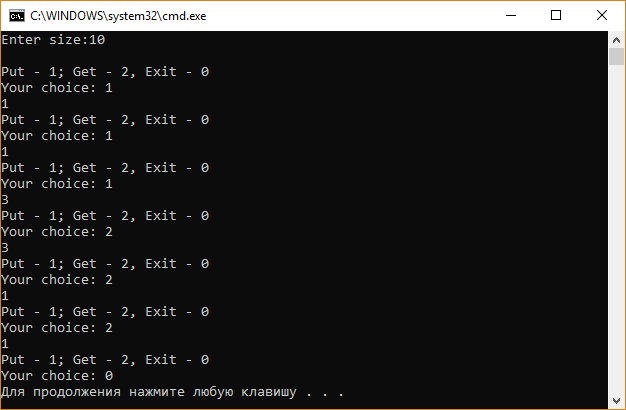


Рис.1 Пример использования.

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

4.Руководство программиста

4.1.Описание структуры программы

Программа состоит из таких модулей как:

* Stack.h – стек.

4.2.Описание структур данных

1. T\* p – указатель на стек.
2. int pos – кол-во элементов.
3. int size – размер стека.
4. TStack(int n) – конструктор с параметром. Принимает размер.
5. TStack(TStack<T>& A) – конструктор копирования.
6. ~TStack() – деструктор.
7. bool IsEmpty() – метод. Возвращает единицу если стек пуст.
8. bool IsFull() - метод. Возвращает единицу если стек полон.
9. int GetSize() – метод. Возвращает размер стека.
10. int GetCount() – метод. Возвращает кол-во элементов в стеке.
11. void Put(const T s) – метод. Кладет элемент в стек.
12. T Get() – метод. Берет верхний элемент из стека.
13. T Check() – метод. Просматривает верхний элемент из стека.

4.3.Описание алгоритмов

Void Put(const T s)

Если стек не является полным, то в элемент массива с индексом pos кладется s, после чего увеличиваем pos на 1.

T Get()

Если стек не пуст, то уменьшаем pos на 1. После чего возвращаем элемент массива с индексом pos.

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 со стеком, улучшил понимание основ программирования базовых структур данных, осознал важность стека в жизни человека. Стек значительно упрощает работу с информацией и помогают решать многие практические задания.

7.Литература

1. Википедия https://ru.wikipedia.org/wiki/Стек (Дата обращения 29.12.2018)
2. Сайт prog-cpp.ru https://prog-cpp.ru/data-stack/ (Дата обращения 29.12.2018)