|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_Информатика и системы управления\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_Программное обеспечение ЭВМ и информационные технологии\_\_\_\_\_\_\_\_

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ ПО ПРЕДМЕТУ "ТИПЫ И СТРУКТУРЫ ДАННЫХ" №4**

Студент \_Золотухин Алексей Вячеславович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*фамилия, имя, отчество*

Группа \_\_\_\_\_\_ИУ7-34Б\_\_\_\_\_\_\_\_\_

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_Золотухин А. В.\_\_\_\_

*подпись, дата фамилия, и.о.*

Принял \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_Барышникова М. Ю. \_

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Москва, 2021**

**Цель работы**

Реализовать операции работы со стеком, который представлен в виде динамического массива и в виде односвязного линейного списка; оценить преимущества и недостатки каждой реализации: получить представление о механизмах выделения и освобождения памяти при работе со стеком.

**Условие задачи**

Создать программу работы со стеком, выполняющую операции добавление, удаления элементов и вывод текущего состояния стека. Реализовать стек: а) массивом; б) списком. Все стандартные операции со стеком должны быть оформлены подпрограммами. При реализации стека списком в вывод текущего состояния стека добавить просмотр адресов элементов стека и создать свой список или массив свободных областей (адресов освобождаемых элементов) с выводом его на экран.

**Описание функционала системы**

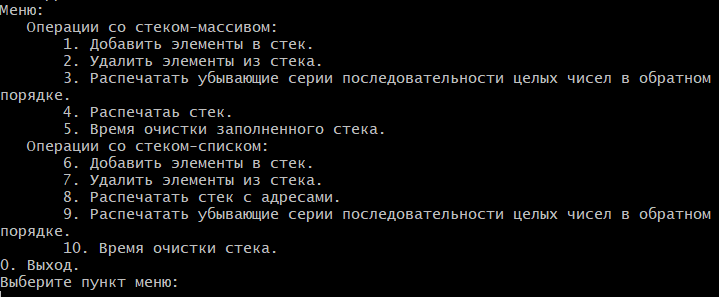
Ввод и вывод производится пользователем вручную через консоль с помощью меню.

В меню пользователь может добавить элемент в стек, удалить элемент из стека. При работе с программой стеки выводятся автоматически.

Добавление и удаление элементов происходит по одному элементу.

Изначально стеки пустые. Максимальный размер стека – 10 элементов. Максимальный размер стека изменяется вручную в файлах.

***Главное меню программы***



**Требования к программе**

Программа написана на языке Си.

При неправильном вводе пользователем какого-либо значения программа завершает свою работу, выдавая сообщение об ошибке.

**Входные данные:** на вход программа получает целое число в зависимости от требуемой операции. Далее, в случае выбора операций добавления элементов – целые числа (элементы стека).

**Выходные данные:** при выборе пункта меню – информация о последнем совершённом действии;

**Аварийные ситуации**

1. Удаление элемента из пустого стека
2. Добавление элемента при полностью заполненном стеке
3. Неправильный ввод пункта меню

**Структуры данных**

Стек-массив представляет собой структуру:

typedef struct

{

    size\_t n;

    size\_t mn;

    int \*stack;

} stack\_a\_t;

Стек-список представляет собой структуру:

typedef struct node

{

    int d;

    struct node \*next;

} stack\_l\_t;

**Функциональные тесты**

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Вывод** | **Класс эквивалентности** |
| Меню: y | Сообщение об ошибке, завершение программы | Некорректный ввод в меню |
| Меню: 3 | Успешный ввод | Корректный ввод в меню |
| Ввод элемента стека при добавлении: а | Сообщение об ошибке, завершение программы | Некорректный ввод при добавлении элемента в стек |
| Ввод элемента стека при добавлении: 3 | Успешный ввод | Корректный ввод при добавлении элемента в стек |

**Алгоритм**

Ввод: choice

Вывод: –

Считать choice

Пока choice != 0

Если choice = 1

Добавить элементы стек-массив

Если choice = 2

Удалить элементы из стек-массива

Если choice = 3

Распечатать элементы из стека массива по заданию

Если choice = 4

«Посмотреть» на содержание стек-массива

Если choice = 5

Замер времени очистки стек-массива

Если choice = 6

Добавить элементы стек-список

Если choice = 7

Удалить элементы из стек-список

Если choice = 8

Распечатать элементы из стека-списка по заданию

Если choice = 9

«Посмотреть» на содержание стек-списка с адресами

Если choice = 10

Замер времени очистки стек-списка

Все если

Считать choice

Все пока

**Выводы**

В ходе выполнения данной лабораторной работы я научился реализовывать стек двумя способами – массивом и списком. Также была сравнена эффективность этих способов по памяти и времени, затрачиваемому на добавление и удаление элементов.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Количество элементов | Массив, очистка | Массив, память | Список, очистка | Список, память |
| 10 | 0us | 56B | 13us | 160B |
| 100 | 2us | 416B | 231us | 1600B |
| 1000 | 12us | 4016B | 18762us | 16000B |

Также стоит отметить, что работать со стеком-массивом эффективнее, чем работать со стеком-списком. По результатам тестирования видно, что при добавлении и удалении элемента массивы значительно быстрее списков из-за отсутствия необходимости выделения/освобождения памяти под каждый элемент, а также не тратится время на то чтобы добраться до последнего элемента структуры.

При реализации стека списком элементы располагаются в определенных участках памяти. Также стоит отметить, что фрагментации памяти не наблюдается.

**Контрольные вопросы**

1. Что такое стек?

Стек – это последовательный список с переменной длиной, в котором включение и исключение элементов происходит только с одной стороны – с его вершины. Стек функционирует по принципу: последним пришёл – первым ушёл, Last In – First Out (LIFO).

2. Каким образом и сколько памяти выделяется под хранение стека при различной его реализации?

При хранении стека в виде массива под него выделяется определенный объем памяти. При хранении стека в виде списка объем памяти ограничен объемом доступной оперативной памяти, но может быть ограничен искусственно.

Память под массив выделяется len\*sizeof(элемент). Память для списка выделяется не только под элементы самого списка, но также и под указатели.

3. Каким образом освобождается память при удалении элемента стека при различной реализации стека?

При реализации стека массивом память освобождается в конце работы программы при уничтожении соответствующей переменной. При реализации стека списком память освобождается при каждом удалении элемента.

4. Что происходит с элементами стека при его просмотре?

При просмотре стека он очищается.

5. Каким образом эффективнее реализовывать стек? От чего это зависит?

Эффективнее реализовывать стек массивом, поскольку добавление/удаление осуществляются просто по индексу последнего элемента, а память выделяется один раз (несколько раз, если стек расширяемый).