МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра защиты информации

*Изображение выглядит как текст, Шрифт, снимок экрана, логотип

Автоматически созданное описание*

**ЛАБОРАТОРНАЯ РАБОТА №1**

**«**Исследования угроз информационной безопасности**»**

**по дисциплине: «*Основы информационной безопасности*»**

|  |  |
| --- | --- |
| Выполнил:  Студент гр. «АБс-324», «АВТФ»  *Петров Максим Игоревич*  «» октября 2024г  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | Проверил:  *Ассистент кафедры ЗИ*  *Исаев Глеб Андреевич*  «\_\_\_» \_\_\_\_\_\_ 2024г  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Новосибирск 2024

**Цели и задачи работы:** изучение алгоритмов формирования и работы с абстрактными структурами данных.

**Задание к работе:** Самостоятельно решить задачи в соответствии с индивидуальным вариантом.

**Задания:**

1. Реализовать структуры данных с базовым набором операций.
2. Реализовать интерфейс работы со структурами данных. Предусмотреть считывание из файла и запись в файл при внесении изменений в состав элементов.
3. Обосновать сложность выполнения каждой операции с позиции BigO нотации. Составить таблицу BigO по операциям и структурам в отчете.
4. В отчете в контрольном примере представить визуальные формы, подтверждающие соответствие результатам работы программы.

**Таблица BigO по операциям структур**

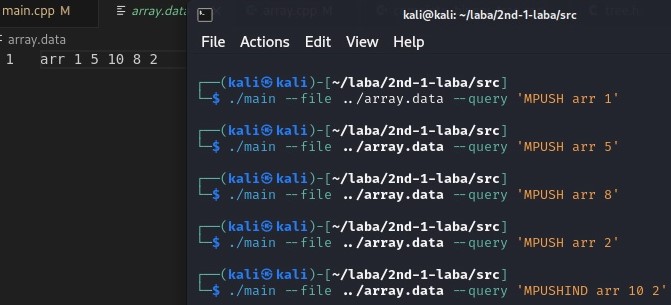
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Структура** | **Вставка** | **Удаление** | **Доступ** | **Поиск** | **Замена** |
| Массив | O(n) | O(n) | O(1) | O(n) | O(1) |
| Список | O(1) | O(n)(конец/начало - O(1)) | O(n) | O(n) | O(n) |
| Стек | O(1) | O(1) | O(n) | O(n) | - |
| Очередь | O(1) | O(1) | O(n) | O(n) | - |
| Хеш-таблица | O(1) | O(1) | O(1) | O(1)(В случае коллизий O(n)) | O(1) |
| Дерево | O(log(n)) | O(log(n)) | O(log(n)) | O(log(n)) | O(log(n)) |

Контрольные примеры

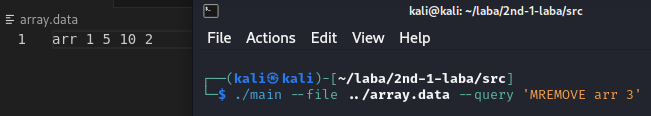
Открываем терминал в директории где находится наш проект и используем команду: ./названиеФайла –file названиеФайла –query “Команда”.

Массив:

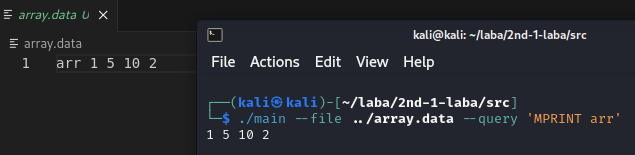
Добавление элементов:

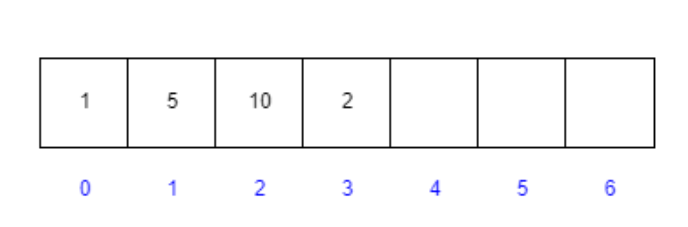


Удаление элементов по индексу 3:



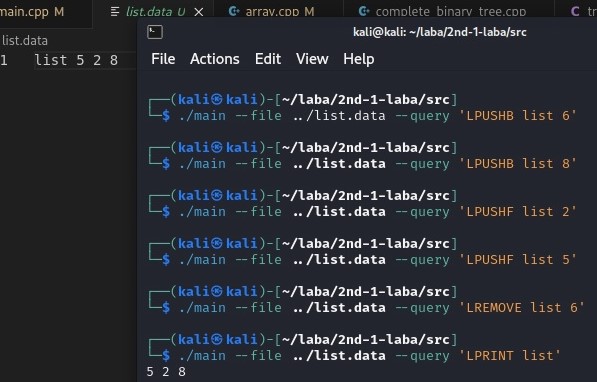
Вывод массива на экран:

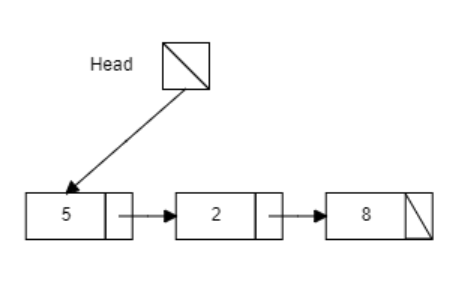




Список:

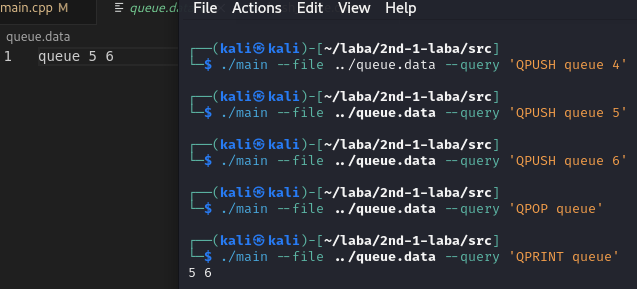
Добавление элементов 6 и 8 в конец списка, и добавление 2 и 5 в начало списка. Далее происходит удаление элемента 6, после происходит вывод списка.

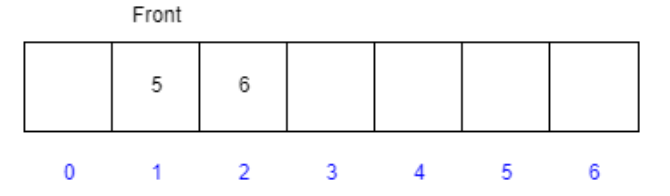




Очередь:

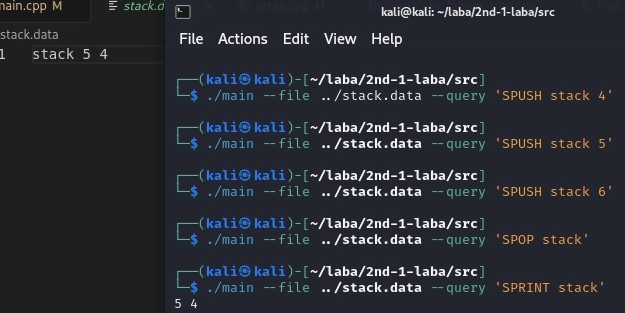
Добавление элементов в очередь 4,5,6. Далее происходит удаление элемента из очереди и вывод.

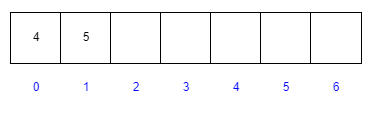




Стек:

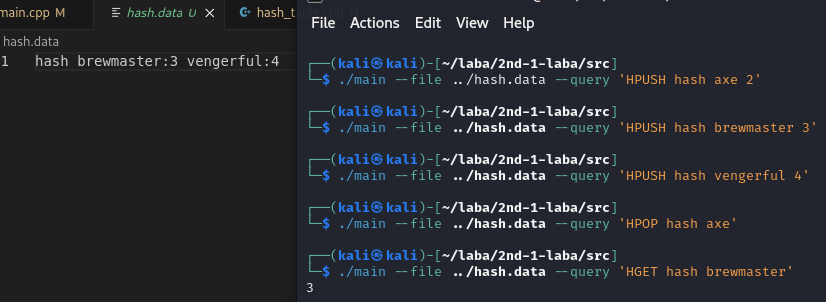
Добавление элементов в стек 4,5,6. Далее происходит удаление элемента 6 из стека и вывод.

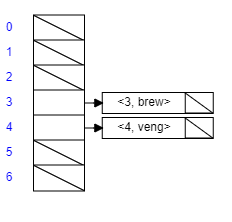




Хеш таблица:

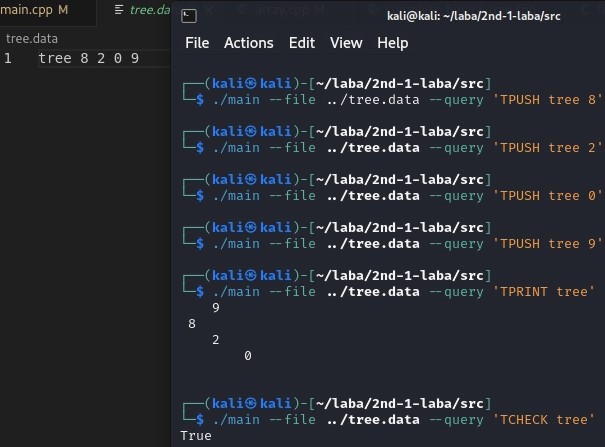
Добавление элементов: axe:2; brewmaster:3; vengerful:4. Удаление элемента по ключу axe, и получаем значение по ключу brewmaster.

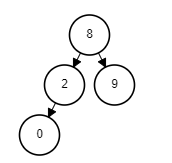




Дерево:

Добавление элементов 8, 2, 0, 9. Далее вывод дерева и проверка на complete.





**Вывод**

В ходе выполнения работы я научился реализовывать свои структуры данных, производить их обработку и обновлять. Также реализовывать своё меню по работе со структурами.

Ссылка на гит: <https://github.com/Mixassss/2nd-1-laba>