Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский химико-технологический университет имени Д.И. Менделеева»

Кафедра информационных компьютерных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

Выполнил студент группы КС-36 Алёшин Михаил Алексеевич

Ссылка на репозиторий: https://github.com/MUCTR-IKT-CPP/MAAleshin\_36\_algo

Приняли: Пысин Максим Дмитриевич

Краснов Дмитрий Олегович

Дата сдачи: 14.02.2022

Оглавление

[Описание задачи. 2](#_Toc63548272)

[Описание метода/модели. 2](#_Toc63548273)

[Выполнение задачи. 2](#_Toc63548274)

[Заключение. 2](#_Toc63548275)

# Описание задачи.

В рамках лабораторной работы необходимо изучить и реализовать одну из трёх структур(двухсвязный список, стек, очередь), в соответствии со своим вариантом, при этом, все структуры должны:

* Использовать шаблонный подход, обеспечивая работу контейнера с произвольными данными.
* Реализовывать свой итератор с реализацией операторов ++ и !=
* Обеспечивать работу стандартных библиотек и конструкции for each если она есть в языке.
* Проверку на пустоту и подсчет количества элементов.
* Операцию сортировки с использованием стандартной библиотеки.

Список должен реализовывать операции:

* добавления элемента после произвольного элемента
* удаление произвольного элемента из списка

Для демонстрации работы структуры необходимо создать набор тестов(под тестом понимается функция, которая создаёт структуру, проводит операцию или операции над структурой и удаляет структуру):

* заполнение контейнера 1000 целыми числами в диапазоне от -1000 до 1000 и подсчет их суммы, среднего, минимального и максимального.
* Провести проверку работы операций вставки и изъятия элементов на коллекции из 10 строковых элементов.
* заполнение контейнера 100 структур, содержащих фамилию, имя, отчество и дату рождения (от 01.01.1980 до 01.01.2020) значения каждого поля генерируются случайно из набора заранее заданных. После заполнение необходимо найти всех людей младше 20 лет и старше 30 и создать новые структуры, содержащие результат фильтрации, проверить выполнение на правильность подсчётом кол-ва элементов, не подходящих под условие в новых структурах.
* Заполнить структуру 1000 элементов и отсортировать ее, проверить правильность использую структуру из стандартной библиотеки и сравнив результат.
* Перемешать все элементы отсортированного списка в случайном порядке.

# Описание метода/модели.

Список — это структура данных, в которой каждый элемент связан с одним из своих соседних элементов, для однонаправленного списка это следующий элемент. Структура одного элемента списка:   
Двунаправленный – поле указателя на предыдущий, поле значения, поле указателя на последующий  
Однонаправленный – поле значения элемента, поле указателя на последующий  
Имея такую структуру список позволяет легко проводить операции вставки и удаления, так же в зависимости от реализации списки могут поддерживать обращения по индексу, хотя в базовой модели такого не предполагается. Вне зависимости о того, является ли список двунаправленным или однонаправленным, для вставки элемента алгоритм будет выглядеть так:

1. Получить номер (или, чаще всего, итератор) элемента после которого производиться вставка.
2. Получить вставляемое значение
3. Создать новый объект элемента списка с полем значения нужного типа и полем указателя на следующий элемент.
4. Записать в поле значения нового объекта вставляемое значение
5. Записать в поле указателя на следующий элемент поставить значение аналогичного поля после которого производиться вставка
6. У элемента, после которого производиться вставка, поменять значения поля указателя на следующий элемент на указатель на ново созданный элемент списка

Для двунаправленного списка добавляются аналогичные действия с указателем на предыдущий элемент.

Вне зависимости о того, является ли список двунаправленным или однонаправленным, для удаления элемента алгоритм будет выглядеть так:

1. Получить номер (или, чаще всего, итератор) элемента который подлежит удалению из списка.
2. Найти элемент, предшествующий удаляемому элементу
3. Сохранить значение поля указателя на последующий элемент удаляемого элемента
4. Вставить в поле указателя на последующий элемент предшествующего удаляемому элементу сохраненное значение
5. Удалить элемент

Для двунаправленного списка добавляются аналогичные действия с указателем на предыдущий элемент.

Преимущества:

* Невозможность переполнения такой структуры
* Быстрота и простота операций удаления и вставки по сравнению с массивом (и простота реализации по сравнению с другими структурами)
* Быстрота манипуляции списком при работе с комплексными большими объектами, так как минимизируются операции копирования самих элементов, работа видеться с указателями.

Недостатки:

* Использование дополнительного места для хранения указателей.
* Отсутствие эффективного способа обращения к произвольному элементу.
* Слабая локализация данных в памяти, что делает невозможным применение процессорных векторных оптимизаций, т.е. не дает процессору сразу манипулировать группами элементов.

# Выполнение задачи.

Данная лабораторная работа была реализована на языке python. Реализован класс Dlist, в котором реализован класс Node, в котором хранятся значение самого элемента, и ссылки на предыдущий и следующий элемент. Далее в классе Dlist реализованы операторы ++ и !=. Далее было добавлено в класс Dlist несколько методов для работы со списком: is\_emty(index), add(index), del(), sort(). Продемонстрируем работу списка и заполним список 1000 целыми числами в диапазоне от -1000 до 1000 и подсчитаем их суммы, среднего, минимального и максимального. Результат приведен на рис. 1.

  
Рис. 1. Результаты анализа 1000 элементов.

Далее проведем результаты изъятия и вставка строк в список:

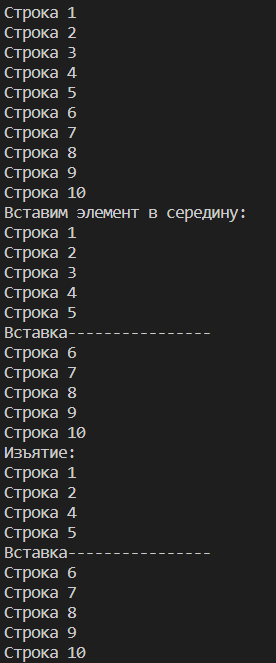


Рис. 2. Изъятие и вставка.

Заполним список из 100 структур содержащих фамилию, имя, отчество и дату рождения (от 01.01.1980 до 01.01.2020) значения каждого поля генерируются случайно из набора заранее заданных. После заполнение найдем всех людей младше 20 лет и старше 30 и создадим новые структуры, содержащие результат фильтрации:

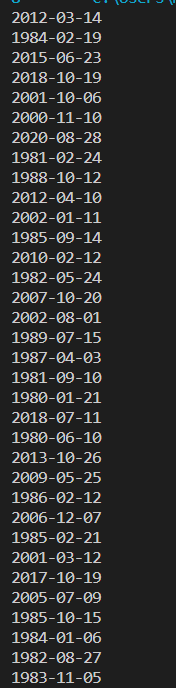


Рис. 3. Список людей больше 30 и меньше 20 лет.

Заполним структуру 1000 элементов и отсортируем ее, проверим правильность использую структуру из стандартной библиотеки и сравним результат:

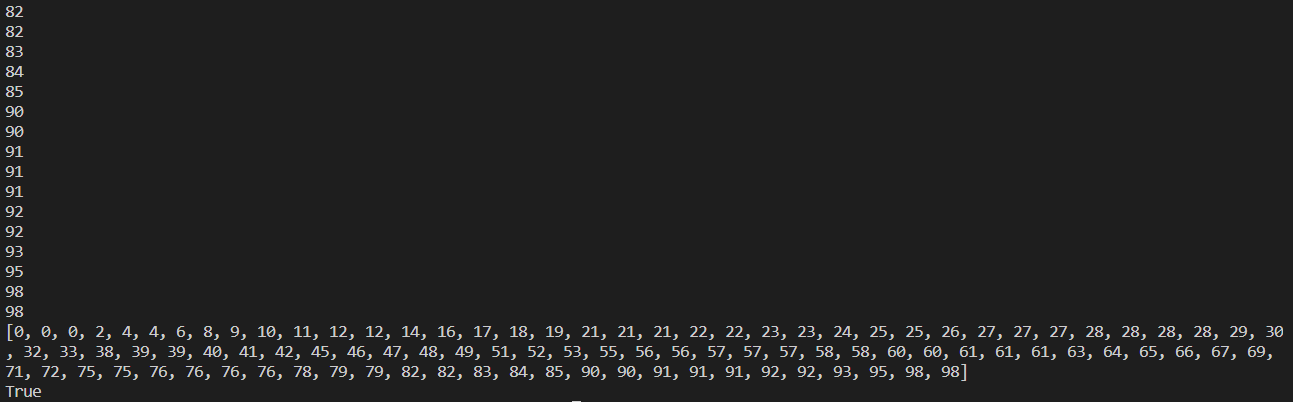


Рис. 4. Результат работы сортировки.

Перемешаем отсортированный список в случайном порядке:

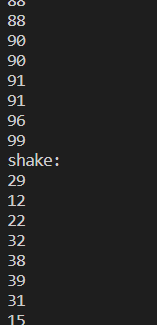


Рис. 5. Начало перемешивания

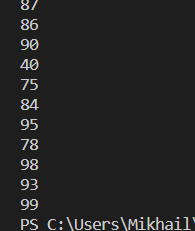


Рис. 6. Конец перемешивания

# Заключение.

Из полученных нами результатов можно сделать вывод, что со списками можно производить большое количество операций, так же, как и с массивом. Список нельзя переполнить, так как он хранит только ссылки на значения. Самым главным преимуществом списка является то, что он работает с ссылками, что минимизирует операции копирования элементов. Однако у списка есть и недостатки, одним из которых является неэффективный доступ произвольного элемента, так как приходится идти от начала списка. Вторым главным недостатком является отсутствие локализации элементов списка в памяти, что не дает обращаться сразу к группе элементов.