**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 Перминова П.А.

Приняли: Пысин Максим Дмитриевич

Краснов Дмитрий Олегович

Дата сдачи:

**Оглавление**

[**Описание задачи**](#_j6utndkz3a28) **3**

[**Описание метода/модели**](#_k3f55hfgbc2p) **4**

[**Выполнение задачи**](#_ixk8vo7b9gik) **5**

[**Заключение**](#_61ena4qjart6) **6**

# Описание задачи

В рамках лабораторной работы необходимо изучить и реализовать одну из трех структур (двусвязный список), 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 создает 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 подсчетом кол-ва элементов не подходящих под условие в новых структурах.
* (Список) Перемешать все элементы отсортированного списка в случайном порядке.

# **Описание метода/модели**

Список - абстрактный тип данных, реализующий упорядоченный набор значений. Списки отличаются от массивов тем, что доступ к их элементам осуществляется последовательно, в то время как массивы - структура данных произвольного доступа. Данный абстрактный тип имеет несколько реализаций в виде структур данных. Некоторые из них будут рассмотрены далее.

Связный список- это структура данных, представляющая собой конечное множество упорядоченных элементов, связанных друг с другом посредством указателей. Каждый элемент этой структуры содержит поле с какими-либо данными, а также указатель (ссылку) на следующий и/или предыдущий элемент. По структуре связности выделяют односвязные, двусвязные, кольцевые и некоторые другие списки.

Односвязный список не слишком удобен, так как из одной точки есть возможность попасть лишь в следующую точку, двигаясь тем самым в конец. Когда кроме указателя на следующий элемент есть указатель и на предыдущий, то такой список называется двусвязным.  
Возможность двигаться как вперед, так и назад полезна для выполнения некоторых операций, но дополнительные указатели требуют задействования большего количества памяти, чем таковой необходимо в односвязном списке.

# Выполнение задачи

Программа алгоритма написана на языке Python.

Функции и классы, использованные в программе:

* Класс DoublyLinkedList
* Класс Node, в котором хранятся значение самого элемента, и ссылки на предыдущий и следующий элемент

Также добавлены методы работы со списком:

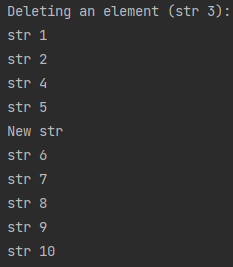
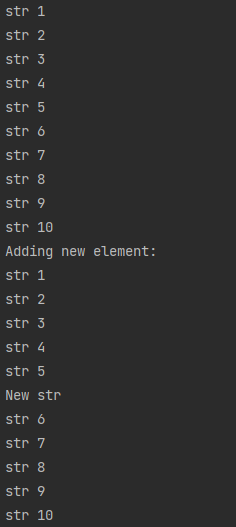
* + add - добавляет новый элемент
  + del - удаляет элемент
  + sort - сортировка
  + is\_empty - проверка на пустоту

Демонстрация работы двусвязного списка:

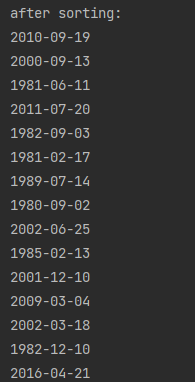
1. Заполним список 1000 целыми числами в диапазоне от -1000 до 1000 и подсчитаем их суммы, среднего, минимального и максимального (analysis\_1000)



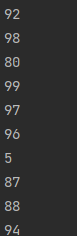
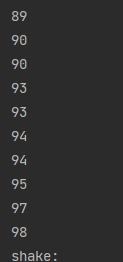
1. Проведем проверку изъятия и вставка строк в список (функция analysis\_str)



1. Заполним контейнер 100 структурами, содержащими фамилию, имя, отчество и дату рождения(от 01.01.1980 до 01.01.2020). После найдем всех людей младше 20 лет и старше 30 и создадим новые структуры, содержащие результат фильтрации. (функция analysis\_struct)



1. Перемешиваем все элементы в случайном порядке (функция shake - на примере последних 10 элементов)



# Заключение

В ходе выполнения лабораторной работы мы видим, что с двусвязным списком можно выполнять множество операций - например, как с массивом. Длина списка не фиксирована, и, так как он хранит только ссылки на значения, его нельзя переполнить. Благодаря работе с ссылками появляется структурная гибкость: порядок элементов связного списка может не совпадать с порядком расположения элементов данных в памяти компьютера, а порядок обхода списка всегда явно задаётся его внутренними связями, что дает данному типу преимущество над массивами. У списков есть и недостатки: каждый узел имеет дополнительный указатель, который требует дополнительного места; операции с двусвязным списком требуют обработки большего количества указателей, следовательно, больше времени.