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-38 (Казанцев Леонард Антонович)

Ссылка на репозиторий: (https://github.com/SimpleMaking/Algorithms-6-sem-/tree/main/Lab\_2(сортировка\_part2))

Приняли: Пысин Максим Дмитриевич

Краснов Дмитрий Олегович

Дата сдачи: (17.02.2023)

Оглавление

[Описание задачи. 2](#_Toc127270410)

[Описание метода/модели. 3](#_Toc127270411)

[Выполнение задачи. 4](#_Toc127270412)

[Заключение. 4](#_Toc127270413)

# Описание задачи.

В данной лабораторной работе необходимо было реализовать коллекцию очередь. Для этого использовал python.

Основные требования к коллекции:

* 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.
* добавление в конец
* взятие с начала

Помимо реализации еще нужно было провести тесты ее работы:

* 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.
* (Стек и Очередь) Инверсировать содержимое контейнера заполненного отсортированными по возрастанию элементами не используя операцию перемещения при помощи итератора, а только операторы изъятия и вставки.

# Описание метода/модели.

Очередь

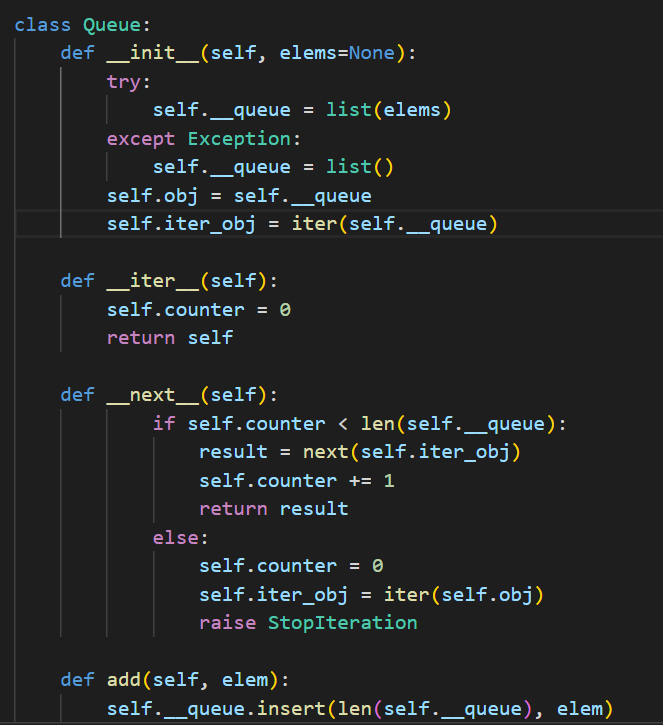
**Очередь** (англ. *queue*) — способ хранения данных. Данные хранятся последовательно, основная особенность очередь заключается в том, что кто первый зашел, тот первый вышел (FIFO – first-in, first-out).

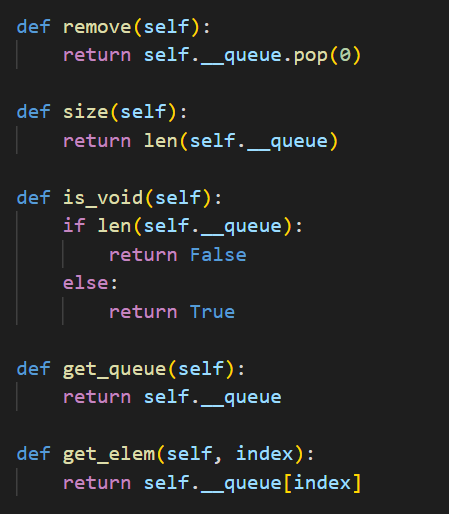
# Выполнение задачи.

Для решения поставленной задачи, как я уже упоминал, был выбран язык программирования python.

Что из себя представляет очередь в моем исполнении? Это класс, который использует механизм встроенной в python коллекции список, но с особенностями, присущими коллекции очередь.

Код класса:





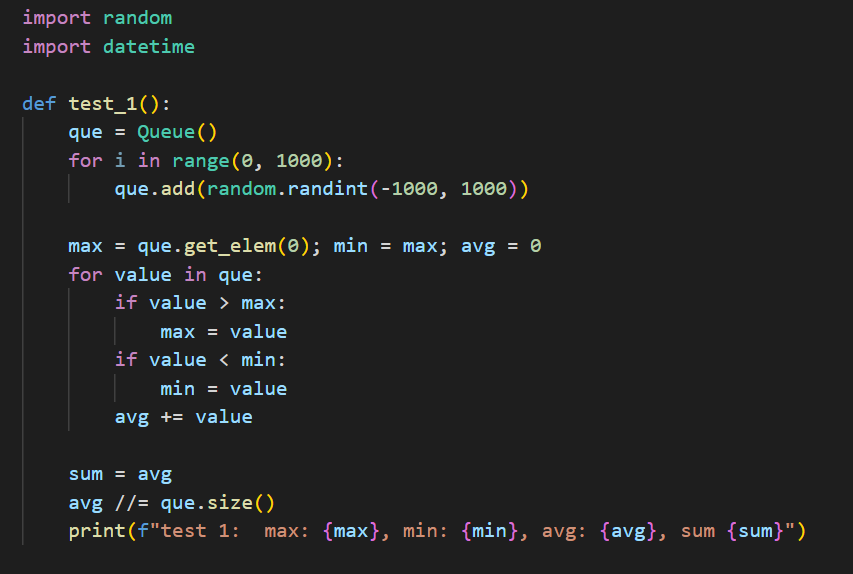
Данный класс реализует механизм итераторов, то есть мы можем работать с данной коллекцией в цикле. Также есть методы изъятия из начала, вставки в конец, проверки на пустоту, получения элемента по индексу и получения размера очереди. Шаблонизации используемый язык не предоставляет, так как является динамически типизированным и принимает на вход любые данные изначально.

Рассмотрим проведенные тесты.

Первый тест

Заполнение контейнера 1000 целыми числами в диапазоне от -1000 до 1000 и подсчет их суммы, среднего, минимального и максимального.

1. Создаем объект очереди и заполняем его случайными числами.
2. Далее с помощью перебора получившейся очереди находим нужные значения.



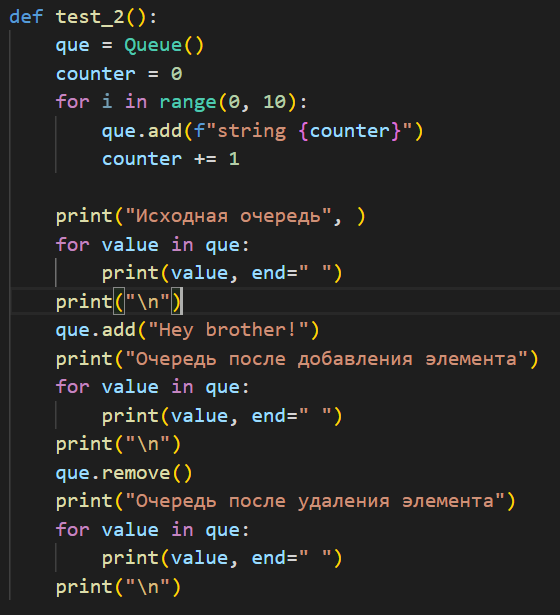
Результат данного теста:



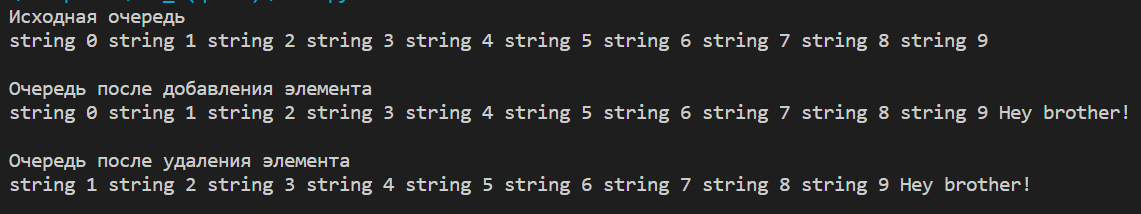
Второй тест

Провести проверку работы операций вставки и изъятия элементов на коллекции из 10 строковых элементов.

1. Создаем очередь и заполняем ее значениями строк, выводим ее.
2. Далее с помощью метода add() добавляем в нее еще один элемент, выводим очередь.
3. Удаляем элемент из начала и выводим получившуюся очередь.



Результат данного теста:



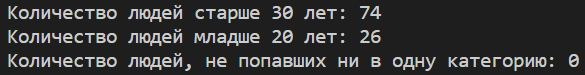
Третий тест

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.

1. Создаем списки для 4 характеристик человека.
2. Создаем 100 структур людей и заполняем их случайными значениями из списков, созданных ранее.
3. Далее смотрим каждую структуру и проверяем, если человек удовлетворяет условиям из задания, то добавляем его в результирующую структуру.
4. Выводим результирующие структуры.



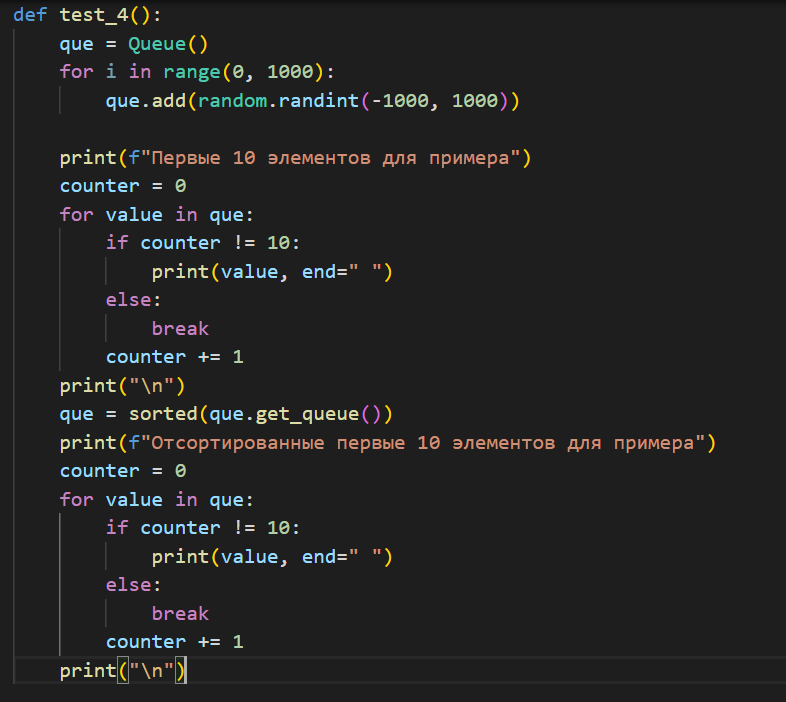
Результат данного теста:



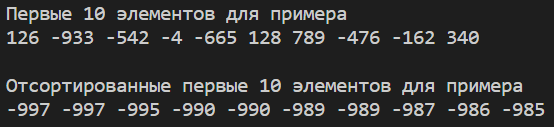
Четвертый тест

Заполнить структуру 1000 элементов и отсортировать ее, проверить правильность использую структуру из стандартной библиотеки и сравнив результат.

1. Заполняем очередь случайными элементами.
2. Выводим первые 10.
3. Сортируем очередь стандартной функцией.
4. Выводим первые 10 элементов из отсортированной очереди.



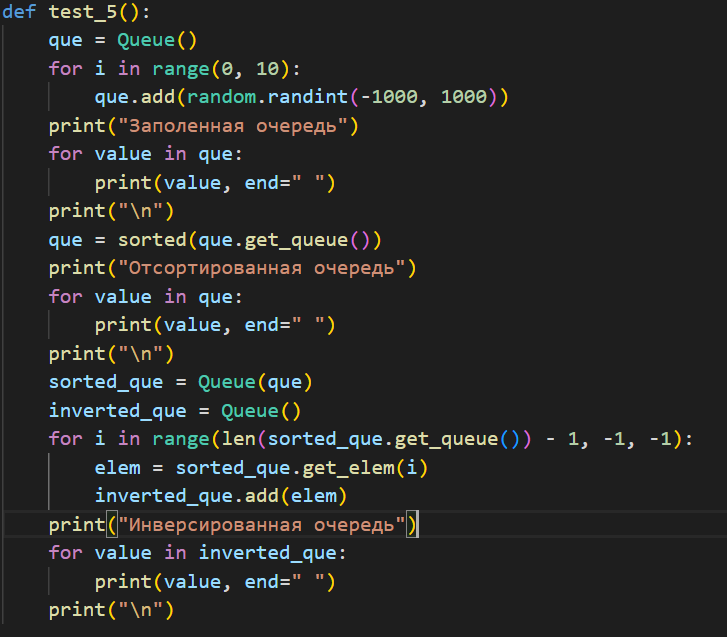
Результат данного теста:



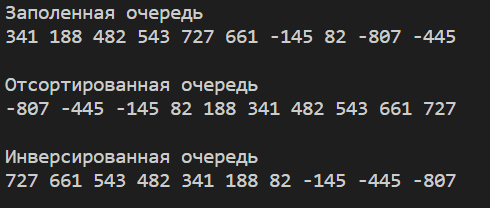
Пятый тест

(Стек и Очередь) Инверсировать содержимое контейнера заполненного отсортированными по возрастанию элементами не используя операцию перемещения при помощи итератора, а только операторы изъятия и вставки.

1. Заполняем очередь случайными числами.
2. Сортируем ее.
3. Инверсируем: идем с конца очереди и заносим каждый ее элемент в новую очередь, выводим результат.



Результат данного теста:



# Заключение.

Реализация очереди на python - простое и приятное занятие, в том числе благодаря динамической типизации и простому устройству итераторов. Все тесты отработали как надо и все функции, указанные в задании, реализованы.