**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-30 Лобачев Дмитрий Сергеевич

Ссылка на репозиторий: https://github.com/MUCTR-IKT-CPP/DSLobachev\_30/blob/main/Algorithms/Laba3/Laba3.cpp

Приняли: Пысин Максим Дмитриевич

Краснов Дмитрий Олегович

Лобанов Алексей Владимирович

Крашенинников Роман Сергеевич

Дата сдачи: 12.03.2023

**Оглавление**

[Описание задачи 3](#_Toc129527897)

[Описание метода/модели 4](#_Toc129527898)

[Выполнение задачи. 5](#_Toc129527899)

[Заключение 11](#_Toc129527900)

# Описание задачи

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

* Использовать шаблонный подход, обеспечивая работу контейнера с произвольными данными.
* Реализовывать свой итератор предоставляющий стандартный для языка механизм работы с ним (для С++ это операции ++ и операция !=)
* Обеспечивать работу стандартных библиотек и конструкции for-each если она есть в языке, если их нет, то реализовать собственную функцию, использующую итератор.
* Проверку на пустоту и подсчет количества элементов.
* Операцию сортировки с использованием стандартной библиотеки.

Стек операции:

* добавление в начало
* взятие из начала

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.
* Заполнить структуру 1000 элементов и отсортировать ее, проверить правильность использую структуру из стандартной библиотеки и сравнив результат.
* Инвертировать содержимое контейнера, заполненного отсортированными по возрастанию элементами не используя операцию перемещения при помощи итератора, а только операторы изъятия и вставки.

# Описание метода/модели

Стек – абстрактный тип данных, представляющий собой список элементов, работающий по принципу LIFO (последним пришел – первым ушел), поэтому, часто его реализация является модификацией над существующим в библиотеке типом списка.

Каждый элемент стека состоит из двух полей: из самого элемента стека и из указателя на следующий элемент.

Операции со стеком:

* Вставка

Алгоритм выглядит следующим образом:

1. Получаем нужный элемент
2. Создаем новый элемент стека, хранящий нужные поля
3. Присваиваем полю со значением в стеке нужный нам элемент
4. Присваиваем к указателю на последующий элемент указатель на текущий первый элемент стека
5. Присваиваем к указателю на текущий первый элемент, указатель на новосозданный элемент

* Удаление

Алгоритм выглядит следующим образом:

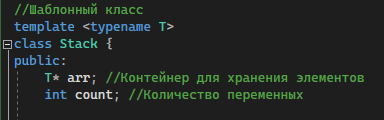
1. Берем верхний элемент стека
2. В указатель на верхний элемент стека записываем значение поля указателя на следующий элемент взятое из удаляемого первого элемента
3. Возвращаем значение удаленного элемента

# Выполнение задачи.

Для реализации данного метода сортировки использовался язык программирования C++.

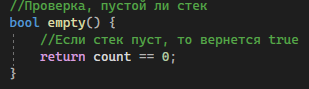
**Стек**

1. Класс стека имеет два поля: массив всех элементов arr и количество переменных в этом стеке count

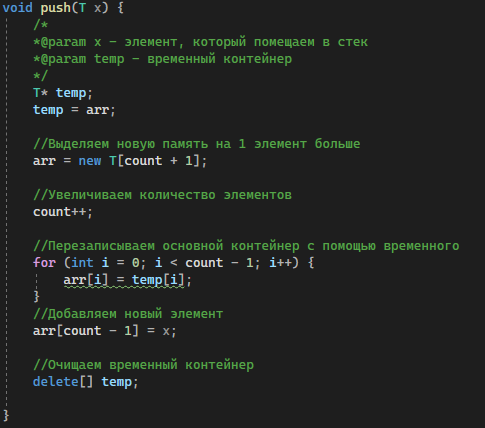


1. Класс стека имеет несколько необходимых нам функций:

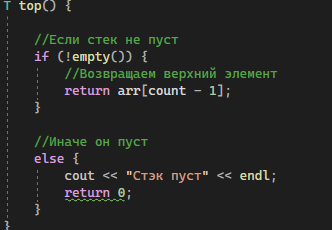
* Функция проверки на пустоту



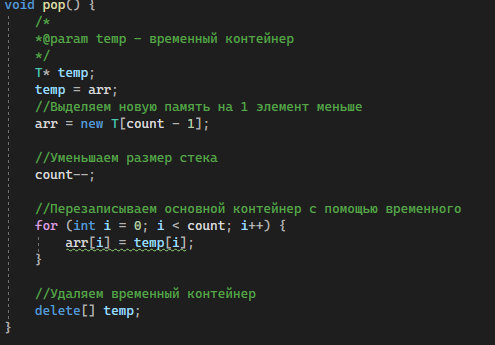
* Функция вставки элемента



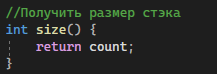
* Функция получения элемента



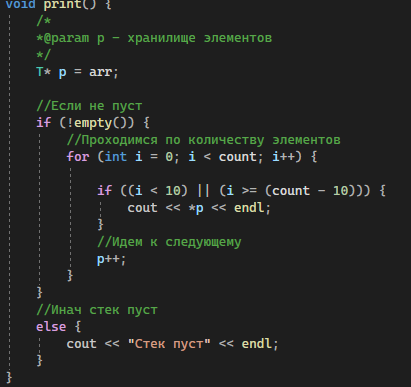
* Функция удаления элемента



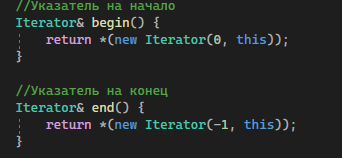
* Функция получения размера стека



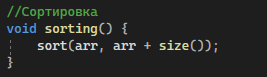
* Функция вывода 10 первых и 10 последний элементов



* Указатели на начало и конец для итератора

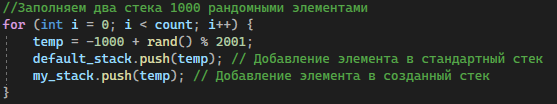


* Функция сортировки стека



**Тестирование структуры**

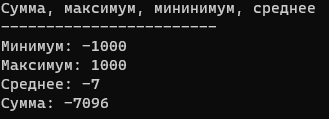
1. Функция oneKFillAndSorting() заполняет стек случайными числами в [-1000; 1000]



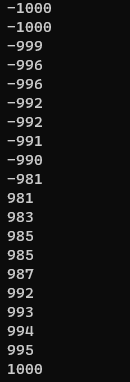
1. Функция minMaxSum() находит сумму всех элементов, максимальным и минимальный элементы и среднее значение стека



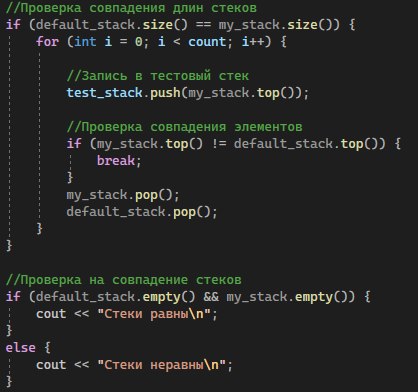
Результат функции:



1. Функция sorting() сортирует данный стек



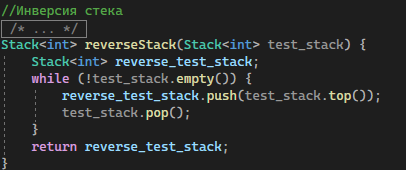
1. По заданию было произведено сравнение стеков, а именно стеков, созданный из шаблона и из стандартной библиотеки.



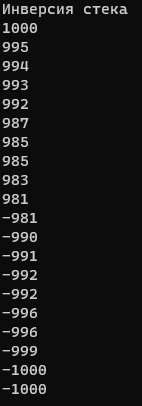
Результат работы:



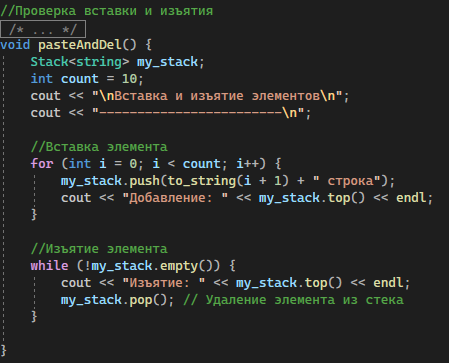
1. Функция reverseStack() инвертирует отсортированный стек



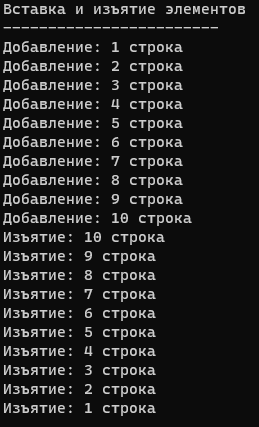
Результат работы:



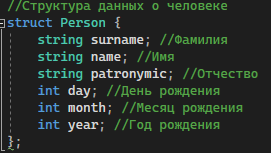
1. Функция pasteAndDel() проверяет вставка и изъятие 10 строковых элементов

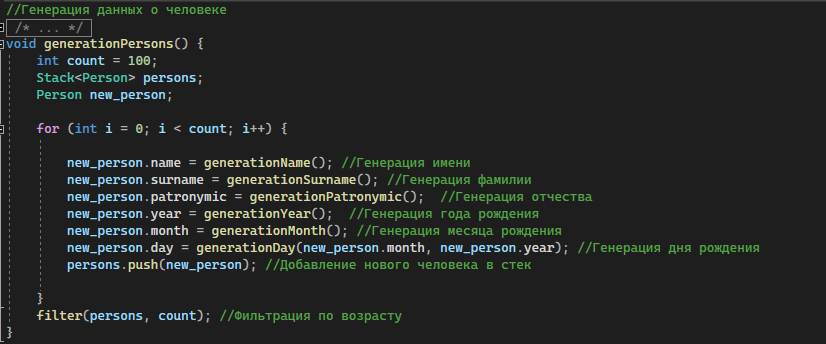


Результат работы:

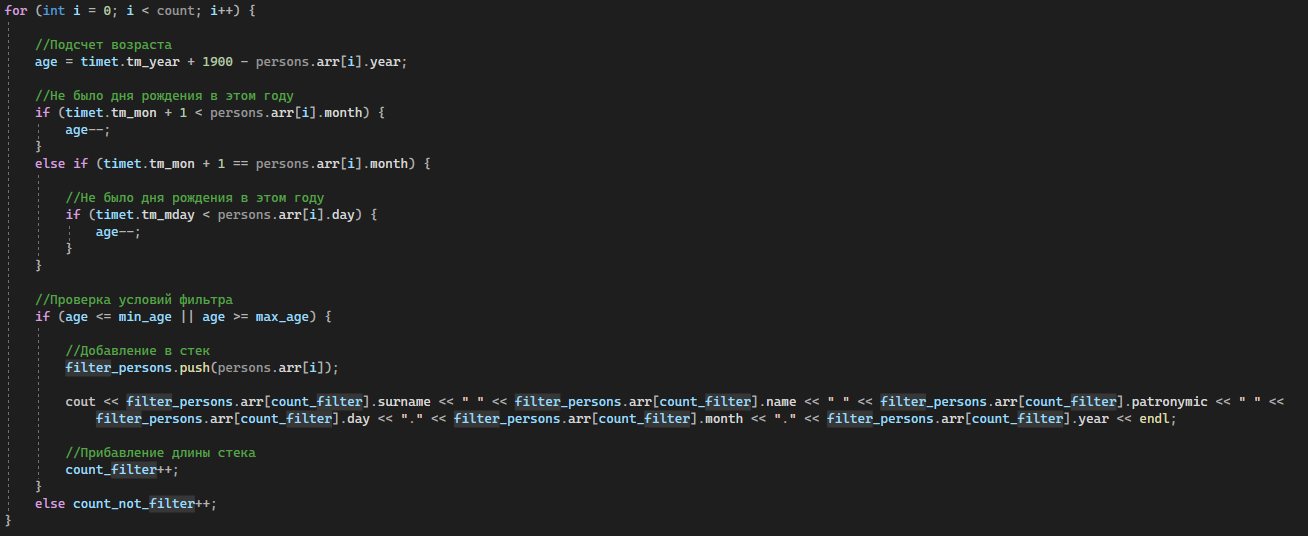


1. По заданию был создан стек структур Person с полями в виде ФИО и даты рождения и заполнен 100 случайными людьми

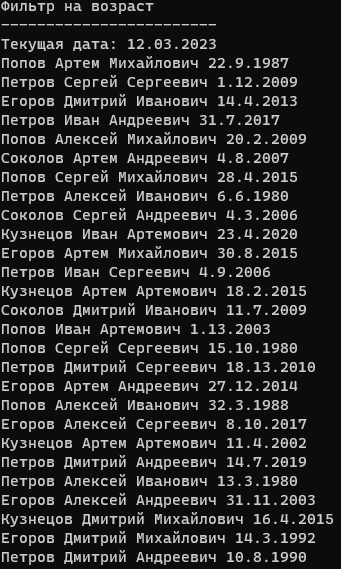




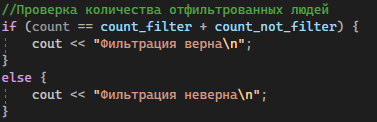
1. Функция filter() отфильтровывает людей, чей возраст меньше 20 или больше 30 лет, и записывает их в отдельный стек



Результат работы:



В данной функции также была реализована проверка на количество вошедших по условию в новый стек и количество не попавших в него.



Результат работы:



# Заключение

По результатам проделанной работы, можно сделать вывод, что реализованный стек имеет практически те же функции, что и стек из стандартной библиотеки, и данный стек также успешно сортируется. Также были успешно написаны функция инверсии стека и функция, позволяющая создать стек структур из полей, содержащие ФИО и дату рождения. Данный стек позже был успешно отфильтрован и проходящие по фильтру люди были занесены в новый стек. Была проведена проверка правильности работы фильтра, показавшая, что фильтрация проходит правильно.