

Отчёт по лабораторной работе 11

Корнеев Егор, СКБ-231

Вступление:

В данной лабораторной работе я научился работать с различными итераторами и контейнерами библиотеки STL.

Фоновая музыка:

Список треков, которые помогли мне выполнить эту лабораторную работу.

- [VOGUE](#)
- [BOO](#)
- [SUPERAGENT](#)
- [цунами](#)
- [ТАГА](#)
- [If I Could Teach the World](#)
- [что для тебя красота](#)

Условия:

Лабораторная работа 11.

Контейнеры и итераторы

1. Контейнерные алгоритмы:

Напишите программу, которая считывает список целых чисел от пользователя, а затем выполняет несколько контейнерных алгоритмов в списке с помощью итераторов.

2. Итераторы-адаптеры:

Используйте адаптеры итераторов, такие как `back_inserter`, `front_inserter` и `inserter`, для изменения контейнеров с помощью итераторов. Напишите программу, которая создает вектор целых чисел, а затем использует итераторы и несколько функций адаптера для добавления или удаления элементов из вектора.

3. Файловый ввод-вывод с итераторами:

Используйте стандартные итераторы для чтения и записи файлов. Напишите программу, которая демонстрирует работу с текстовым файлом с помощью итераторов.

4. Взаимодействие итераторов:

Продemonстрируйте совместимость итераторов между различными типами контейнеров, такими как вектор и двухсторонняя очередь. Напишите программу, которая создает вектор целых чисел, а затем использует итераторы для копирования элементов вектора в очередь.

5. Обратные итераторы:

Используйте обратные итераторы, чтобы изменить порядок элементов в контейнере. Напишите программу, которая считывает список целых чисел от пользователя, а затем использует обратные итераторы для вывода элементов в обратном порядке.

Процедура:

1. Пункт первый. Воздействовал на список чисел с помощью алгоритмов STL, параметрами которых задавал итераторы `l.begin()` и `l.end()`. Вывод списка в консоль также делал с помощью этих итераторов.
2. Пункт второй. Использовал различные итераторы, такие как `back_inserter`, `front_inserter` для добавления элементов в вектор, и копирования элементов вектора в список.
3. Пункт третий. С помощью итераторов `istream_iterator` и `ostream_iterator` записал вектор в файл, а затем считал вектор из файла.
4. С помощью итератора скопировал элементы вектора в очередь.
5. Записал введенные пользователем числа в вектор, а затем с помощью обратных итераторов `rbegin()` и `rend()` вывел их в обратном порядке.

Код:

<https://github.com/egoridze74/cpp-lab-11>

Заключение:

В ходе лабораторной работы я познакомился с различными итераторами, научился ими пользоваться, а также научился взаимодействовать с помощью итераторов с разными контейнерами.