Лабораторная работа №4

Шаблоны. Контейнеры STL

Цель лабораторной работы: *изучение видов контейнеров библиотеки STL, способов их применения и особенностей работы; изучение способов использования шаблонов функций; получение навыков программирования на языке C++.*

Задание на программирование: *разработать программу и провести анализ быстродействия различных контейнеров.*

Порядок выполнения работы:

1) Выбрать вариант по номеру в списке группы (циклически, если группа больше 23 человек)
2) Выполнить индивидуальное задание, пользуясь для выполнения средствами библиотеки STL (методами классов и алгоритмами)[1]. Задание выполняется для каждого из контейнеров: vector, list, multiset, map.
3) Оформить отчет о лабораторной работе в составе:

- титульный лист;
- основная часть.

4) Основная часть должна содержать:

- номер варианта и индивидуальное задание;
- описание выбранной модели решения, сопоставление и объяснение основных частей алгоритма и выбранных для их реализации синтаксических конструкций языка, в том числе модель решения с указанием подходящих методов и алгоритмов из библиотеки STL. Сравнение времени выполнения задачи в целом для каждого контейнера, быстрые и медленные места каждого контейнера, сравнение с теоретической информацией о скорости. Вывод о наиболее подходящем контейнере для задачи;
- исходный код;
- снимки экрана с результатом работы программы;

5) Обязательные элементы кода:

- Шаблон функции, заполняющей произвольными элементами любой контейнер.
- Шаблон функции, выводящей на экран содержимое любого контейнера.
- В качестве функциональных объектов и предикатов использовать лямбда-функции.
- Проверку быстродействия и замеры времени производить на коллекции с большим количеством элементов. От момента заполнения значениями до завершения выполнения задачи (вывод на консоль содержимого контейнеров не

---

[1] https://docs.microsoft.com/ru-ru/cpp/cpp/?view=msvc-170
https://docs.microsoft.com/ru-ru/cpp/standard-library/cpp-standard-library-reference
https://en.cppreference.com/w/cpp

требуется и не учитывается при замерах). Так же проверку можно осуществлять средствами профилирования Visual Studio.

- Продемонстрировать корректность выполнения индивидуального задания программой на малом количестве элементов (в некоторых задачах возможна ручная инициализация, если заполнение шаблоном не даст наглядного результата). Для демонстрации работы требуется вывести на экран начальное состояние контейнеров и конечное.

б) Загрузить отчёт в личный кабинет в формате PDF.

Примечание: особое внимание уделить поиску стандартных алгоритмов и методов контейнеров для решения задачи. Сомописные циклы и злоупотребление std::for_each не приветствуются.

Варианты индивидуальных заданий

1. По коллекции L построить две новых коллекции L1 и L2: первая из элементов с положительными значениями, а вторая из остальных элементов исходной коллекции.
2. Вставить в коллекцию L новый элемент со значением E1 за каждым элементом с заданным значением E, если элемент со значением E входит в L.
3. Вставить в коллекцию L, элементы которой изначально упорядочены по не убыванию их значений, новый элемент со значением E так, чтобы сохранить упорядоченность элементов коллекции.
4. Удвоить каждое вхождение элемента со значением E в коллекции L.
5. Удалить из коллекции L все вхождения элемента со значением E.
6. Удалить из коллекции L все элементы с отрицательными значениями.
7. Удалить из коллекции L за каждым вхождением элемента со значением E один элемент, если он есть и его значение отличено от E.
8. Оставить в коллекции L только первые вхождения одинаковых элементов.
9. В коллекции L из каждой группы подряд идущих элементов с равными значениями оставить только один.
10. Перевернуть коллекцию L.
11. Найти элемент коллекции с максимальным значением.
12. Проверить, есть ли в коллекции L хотя бы два элемента с одинаковыми значениями.
13. Проверить на равенство две коллекции L1 и L2.
14. Вставить в коллекцию L за последним вхождением элемента со значением E все элементы коллекции L1, если элемент со значением E входит в L.
15. Сформировать коллекцию L, включив в неё по одному разу элементы, которые входят хотя бы в одну из коллекций L1 и L2.
16. Сформировать коллекцию L, включив в него по одному разу элементы, которые входят одновременно в обе коллекции L1 и L2.
17. Сформировать коллекцию L, включив в неё по одному разу элементы, которые входят в коллекцию L1, но не входят в коллекцию L2.
18. Сформировать коллекцию L, включив в неё по одному разу элементы, которые входят в одну из коллекций L1 и L2, но в то же время не входят в другую из них.
19. Объединить две упорядоченных коллекции L1 и L2 в одну упорядоченную коллекцию, построив новую коллекцию L.
20. Удалить из коллекции последний отрицательный элемент, если такой есть.
21. Заменить в коллекции L все вхождения элемента со значением E1 на E2.
22. Подсчитать число вхождений элемента со значением E в коллекцию L.
23. Подсчитать количество элементов коллекции, которые совпадают с последним элементом.