Лабораторная работа № 05

Тема: Итераторы и аллокаторы

Цель:

- Изучение устройства коллекций в стандартной библиотеке
- Получение навыков в использовании концепции «итератор»
- Получение навыков в использовании концепции «аллокатор»

Порядок выполнения работы

1. Ознакомиться с теоретическим материалом.
2. Получить у преподавателя вариант задания.
3. Реализовать задание своего варианта в соответствии с поставленными требованиями.
4. Подготовить тестовые наборы данных.
5. Создать репозиторий на GitHub.
6. Отправить файлы лабораторной работы в репозиторий.
7. Отчитаться по выполненной работе путём демонстрации работающей программы на тестовых наборах данных (как подготовленных самостоятельно, так и предложенных преподавателем) и ответов на вопросы преподавателя (как из числа контрольных, так и по реализации программы).

Требования к программе

1. Аллокатор
    a. Реализовать свой аллокатор памяти. Проверить что он корректно работает для контейнера std::map.
    b. Аллокатор должен параметризоваться количеством выделяемых за раз элементов.
    c. Освобождение конкретного элемента не предполагается - аллокатор должен освобождать всю память самостоятельно.
2. Контейнер
    a. Реализовать свой контейнер (согласно варианта задания), который по аналогии с контейнерами stl, параметрезуя аллокатором.
3. Итератор
    a. Реализовать итераторы (обычный и const)
    b. Итератор должен соответствовать std::forward_iterator_tag

Прикладной код должен содержать следующие вызовы:

- создание экземпляра std::map с созданным аллокатором
- заполнение 10 элементами, где ключ — это число от 0 до 9, а значение - факториал ключа
- вывод на экран всех значений (ключ и значение разделены пробелом) хранящихся в контейнере
- создание экземпляра своего контейнера для хранения int с собственным vаллокатором – заполнение контейнера и печать его элементов

Варианты заданий:

Вариант	Контейнер	Хранилище внутри аллокатора

1.	Динамический массив	std::vector
2.	Стек	std::vector
3.	Однонаправленный список	std::vector
4.	Двунаправленный список	std::vector
5.	Очередь	std::vector
6.	Динамический массив	std::deque
7.	Стек	std::deque
8.	Однонаправленный список	std::deque
9.	Двунаправленный список	std::deque
10.	Очередь	std::deque
11.	Динамический массив	std::list
12.	Стек	std::list
13.	Однонаправленный список	std::list
14.	Двунаправленный список	std::list
15.	Очередь	std::list
16.	Динамический массив	std::array
17.	Стек	std::array
18.	Однонаправленный список	std::array
19.	Двунаправленный список	std::array
20.	Очередь	std::array

Отчет

1. Код программы на языке C++.
2. Ссылка на репозиторий на GitHub.
3. Набор testcases на Google Test.