

STL. Стандартные контейнеры, Алгоритмы и Классы функциональных объектов в C++

Задание 1. Многокритериальная сортировка

`std::vector<std::pair>`

Создайте вектор `std::vector<std::pair<std::string, int>>`, где:

- первый элемент пары — название города;
- второй элемент — его население.

Заполните вектор минимум 8 элементами.

Отсортируйте его сначала по населению (по убыванию), а при равном населении — по названию города (по возрастанию).

Для сортировки используйте `std::sort` и напишите свой компаратор.

Задание 2. Использование контейнера `std::list` и алгоритмов

Создайте список `std::list<std::string>`, содержащий несколько строк (например, названия месяцев).

Удалите из списка все строки, длина которых меньше 5 символов, используя алгоритм `std::remove_if` и лямбда-выражение.

Выведите содержимое списка до и после удаления.

Задание 3. Поиск элементов в контейнере `std::map`

Создайте отображение `std::map<std::string, int>`, где ключ — название товара, а значение — его цена.

Реализуйте функцию поиска товара по названию: если товар найден — вывести его цену, иначе вывести сообщение "Товар не найден".

Задание 4. Собственный функциональный объект для фильтрации вектора

Создайте вектор `std::vector<std::pair<std::string, int>>`, где строка — название книги, а число — количество страниц.

Реализуйте функциональный объект, принимающий параметры `N` и `M` в конструкторе, и проверяющий, что название книги длиннее `N` символов, а страниц больше `M`.

С помощью `std::copy_if` и вашего функционального объекта отфильтруйте книги в новый вектор и выведите результат.

Задание 5. Комбинирование алгоритмов и контейнеров

Создайте два вектора `std::vector<int>`.

Один вектор заполните числами от 0 до 9, второй — числами от 5 до 14.

Используя алгоритм `std::set_intersection`, найдите пересечение этих двух множеств и сохраните результат в третий вектор.

Выведите содержимое всех трех векторов.