Министерство высшего образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет» (ПНИПУ)**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

ОТЧЁТ

по лабораторной работе №12

Тема: «Ассоциативные контейнеры

Библиотеки STL.»

Выполнил

Студент группы РИС-22-2б

Хохряков А.С.

Проверил доц. Кафедры ИТАС

Полякова О. А.

Пермь 2023

# Постановка задачи

**Задача 1:**

1. Создать ассоциативный контейнер. Контейнер - multiset

2. Заполнить его элементами типа double.

3. Найти элемент с заданным

ключом и добавить его на

заданную позицию

контейнера

4. Найти элемент с

заданным ключом и

удалить его из

контейнера

5. Найти разницу между

максимальным и минимальным

элементами контейнера и

вычесть ее из каждого элемента

контейнера

6. Выполнение всех заданий оформить в виде глобальных функций.

**Задача 2:**

1. Создать ассоциативный контейнер.

2. Заполнить его элементами пользовательского типа Time (класс Time для работы с временными интервалами. Интервал должен быть

представлен в виде двух полей: минуты типа int и секунды типа int. при выводе минуты

отделяются от секунд двоеточием. Реализовать:

добавление секунд (учесть, что в минуте не может быть более 60 секунд)

вычитание секунд). Для пользовательского типа перегрузить необходимые операции.. Для

пользовательского типа перегрузить необходимые операции.

3. Найти элемент с заданным

ключом и добавить его на

заданную позицию

контейнера

4. Найти элемент с

заданным ключом и

удалить его из

контейнера

5. Найти разницу между

максимальным и минимальным

элементами контейнера и

вычесть ее из каждого элемента

контейнера

6. Выполнение всех заданий оформить в виде глобальных функций.

**Задача 3:**

1. Создать параметризированный класс- контейнер ВЕКТОР с элементами типа int.

Реализовать операции:

- доступа по индексу;

+ вектор - сложение элементов векторов а[i]+b[i];

+ число - добавляет константу ко всем элементам вектора;

используя в качестве контейнера ассоциативный контейнер.

2. Заполнить его элементами.

3. Найти элемент с заданным

ключом и добавить его на

заданную позицию

контейнера

4. Найти элемент с

заданным ключом и

удалить его из

контейнера

5. Найти разницу между

максимальным и минимальным

элементами контейнера и

вычесть ее из каждого элемента

контейнера

6. Выполнение всех заданий оформить в виде методов параметризированного класса.

**Контрольные вопросы**

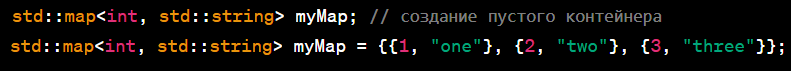
1. Ассоциативный контейнер - это коллекция элементов, которые хранятся в отсортированном порядке и доступны по ключу.

2. Ассоциативные контейнеры библиотеки STL: map, multimap, set, multiset.

3. Доступ к элементам ассоциативного контейнера осуществляется по ключу, используя операцию [] или метод at(). Также можно использовать итераторы для прохода по контейнеру.

4. Примеры методов, используемых в ассоциативных контейнерах: insert(), erase(), find(), size(), empty(), clear(), count().

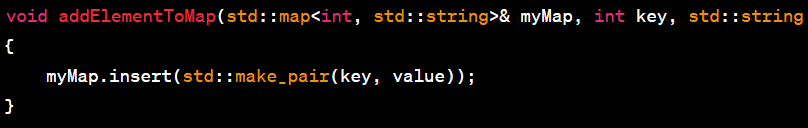
5. Контейнер map можно создать с помощью конструктора по умолчанию или конструктора с параметрами, передавая пары ключ-значение. Пример создания:



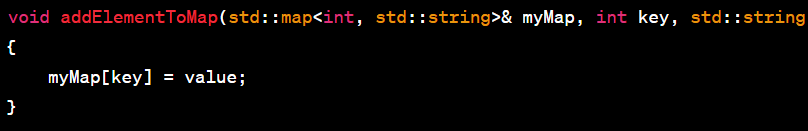
6. Элементы в контейнере map упорядочены по ключу в порядке возрастания. Чтобы изменить порядок на обратный, можно использовать функцию std::greater в качестве компаратора при создании контейнера:



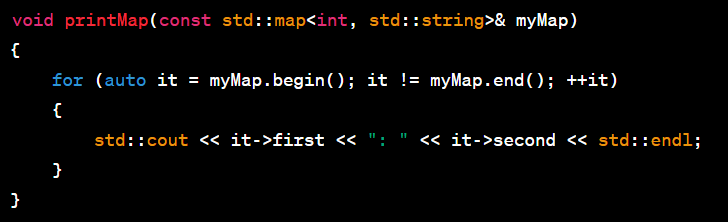
7. Операции, определенные для контейнера map: insert(), erase(), find(), size(), empty(), clear(), count(), operator[], at().

8. Функция для добавления элементов в контейнер map с помощью функции make\_pair():

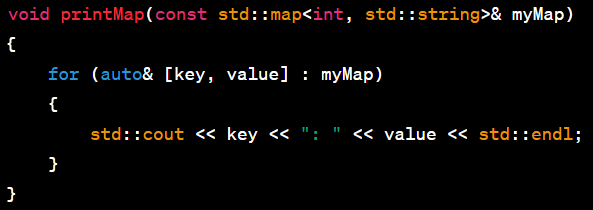
9. Функция для добавления элементов в контейнер map с помощью функции операции прямого доступа []:



10. Функция для печати контейнера map с помощью итератора:



11. Функция для печати контейнера map с помощью функции операции прямого доступа []:

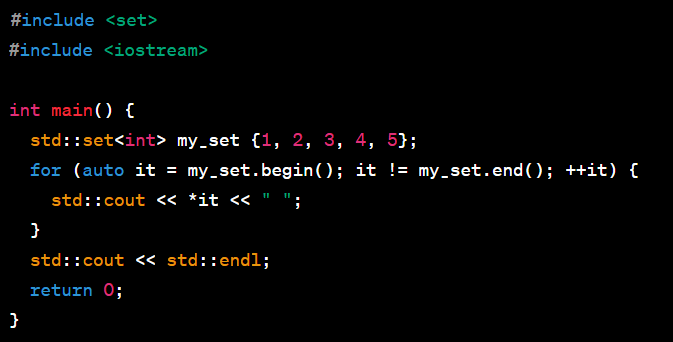


12. Контейнеры map и multimap отличаются тем, что в map каждому ключу соответствует только одно значение, тогда как в multimap одному ключу может соответствовать несколько значений.

13. Контейнер set представляет собой упорядоченный набор уникальных элементов, к которым можно обращаться по их значению.

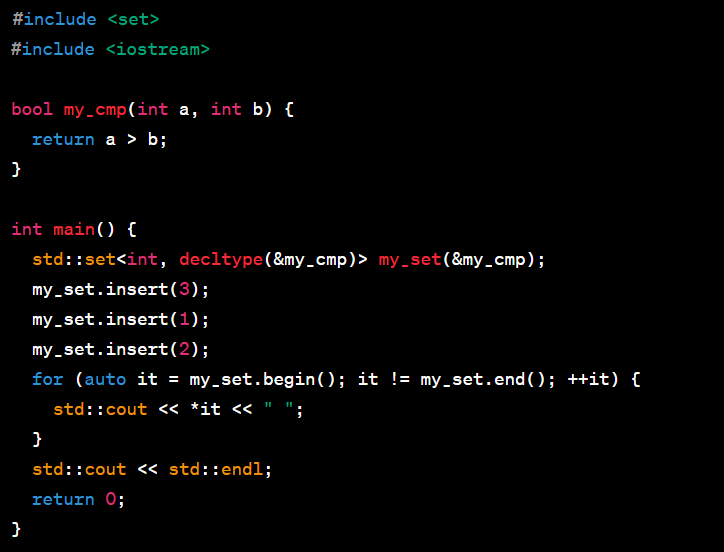
14. Основное отличие между контейнерами map и set заключается в том, что в map каждому ключу соответствует значение, тогда как в set элементы сами являются ключами.

15. Контейнер set можно создать следующим образом:



Этот пример создает контейнер set, содержащий целые числа от 1 до 5. Затем элементы контейнера печатаются на экран.

16. Элементы в контейнере set упорядочены по возрастанию значений по умолчанию. Чтобы изменить порядок на обратный, можно создать контейнер set с компаратором, который будет возвращать обратное значение:

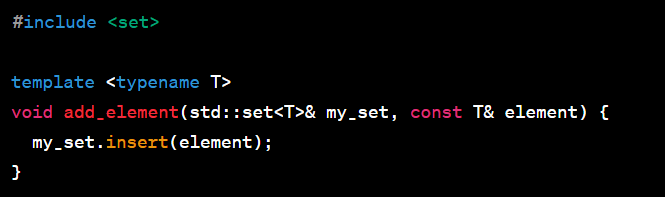


Этот пример создает контейнер set с компаратором my\_cmp, который сравнивает элементы в обратном порядке. Затем элементы контейнера печатаются на экран.

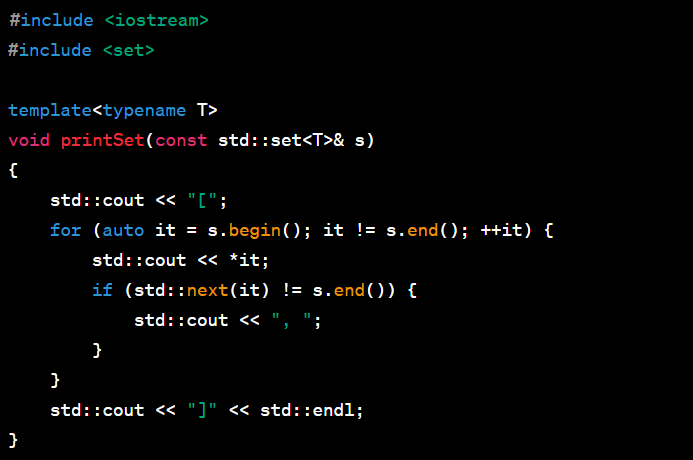
17. Для контейнера set определены следующие операции:

* Вставка элемента (insert)
* Удаление элемента (erase)
* Поиск элемента (find)
* Проверка наличия элемента (count)
* Получение итератора на первый элемент (begin)
* Получение итератора на последний элемент (end)
* Получение размера контейнера (size)

18. Функция для добавления элементов в контейнер set может выглядеть следующим образом:



19. Функция для печати контейнера std::set может быть написана следующим образом:



Данная функция принимает ссылку на std::set и выводит ее элементы в квадратных скобках, разделенных запятыми. Для перебора элементов контейнера используется итератор it, который проходит от начала контейнера до его конца. Каждый элемент контейнера выводится с помощью оператора <<, а если это не последний элемент, то после него добавляется запятая и пробел.

20. std::multiset отличается от std::set тем, что он может хранить дубликаты элементов. То есть, в std::multiset могут быть несколько элементов с одним и тем же значением. В std::set каждый элемент уникален и может быть только один раз в контейнере. Кроме того, в std::multiset элементы не хранятся в отсортированном порядке, а в порядке добавления. В std::set элементы автоматически сортируются по возрастанию при добавлении в контейнер.