Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы» направление подготовки: 09.03.04 – «Программная инженерия»

Лабораторная работа №12.

" Ассоциативные контейнеры библиотеки STL. "

15 вариант

Выполнил студент гр. РИС-24-2б

Иванова Елена Олеговна

Проверил:

Доц. Каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

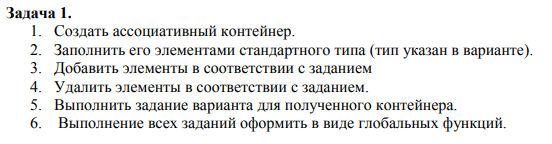
г. Пермь, 2025

Цель:

1) Создание консольного приложения, состоящего из нескольких файлов в системе программирования Visual Studio.

2) Использование ассоциативных контейнеров библиотеки STL в ОО программе.

Постановка задачи:



Задача:



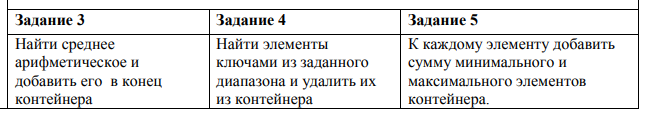
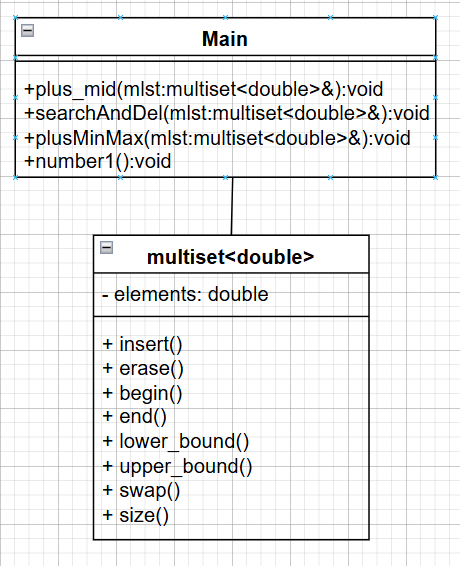
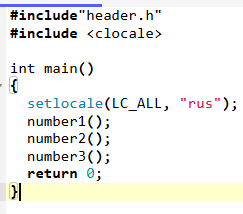
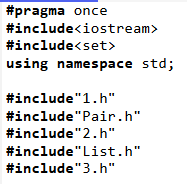


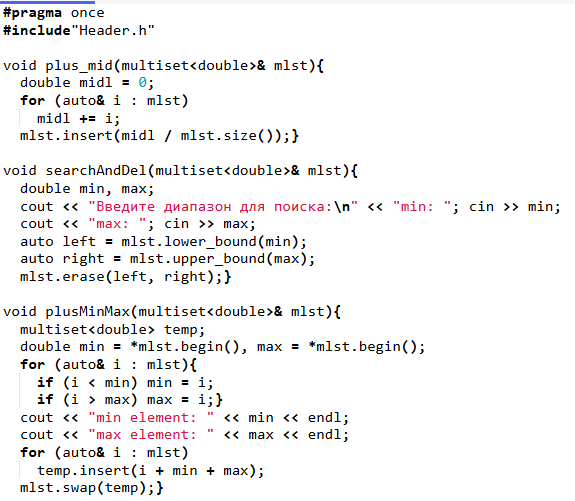
Диаграмма:

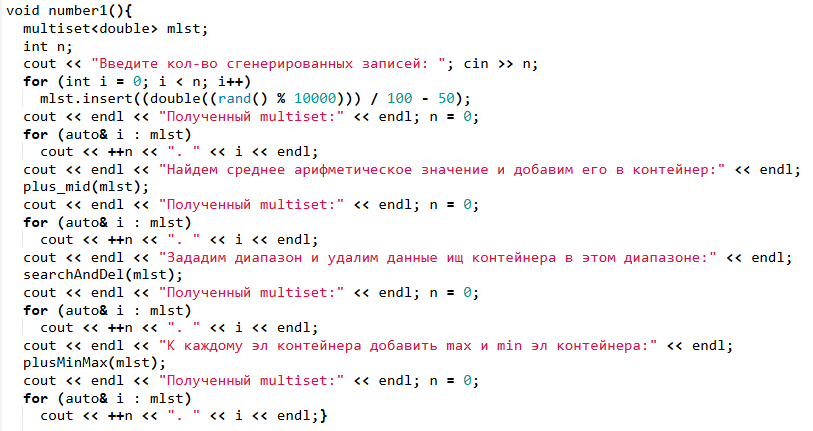


Код:

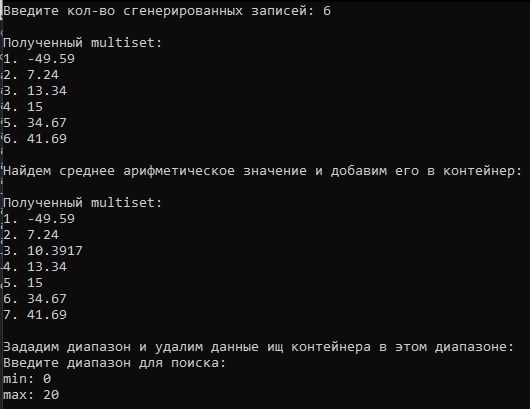


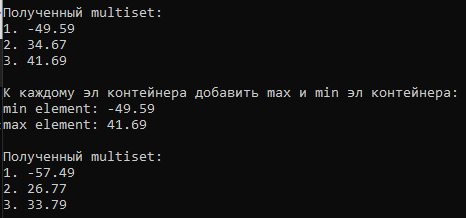




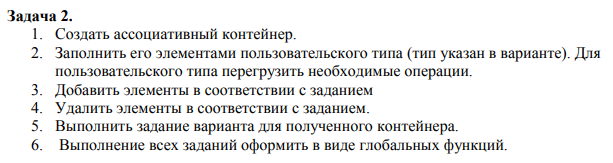


Вывод:





Постановка задачи:



Задача:



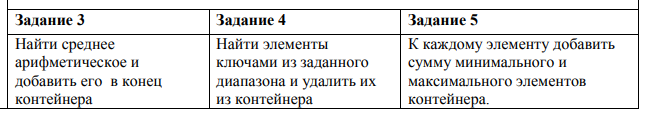
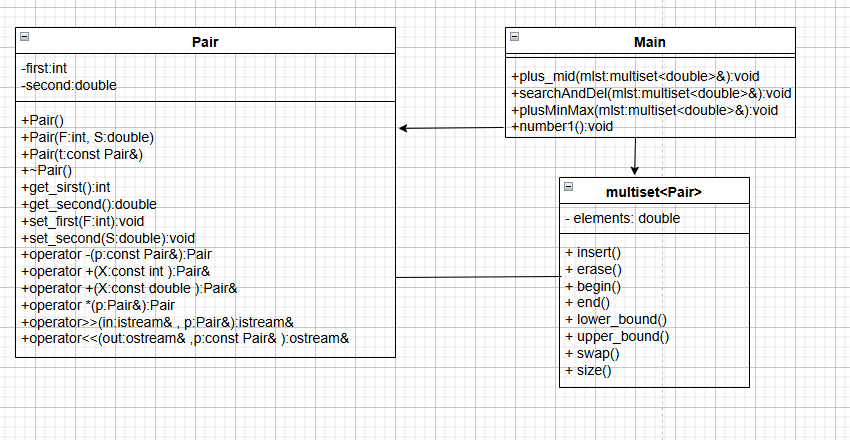
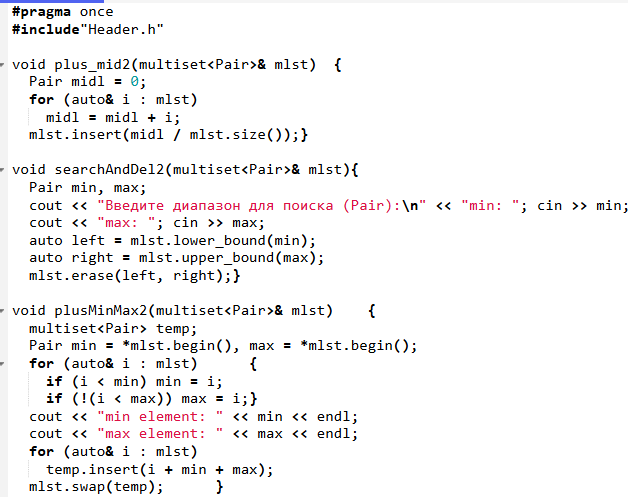
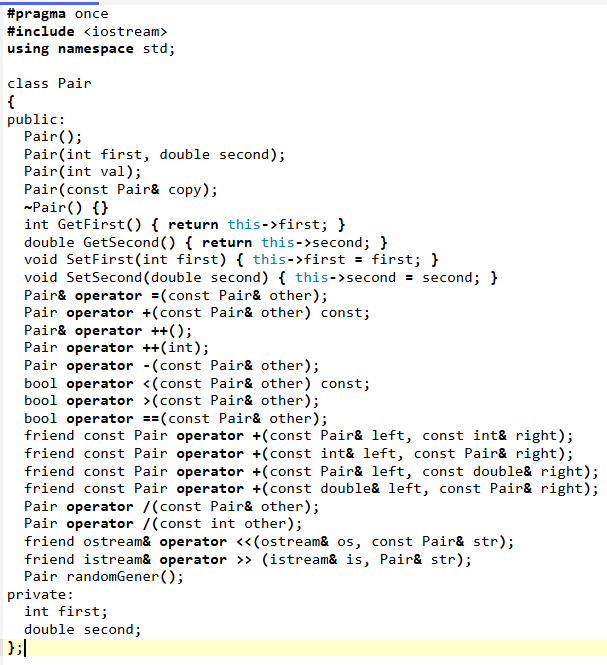
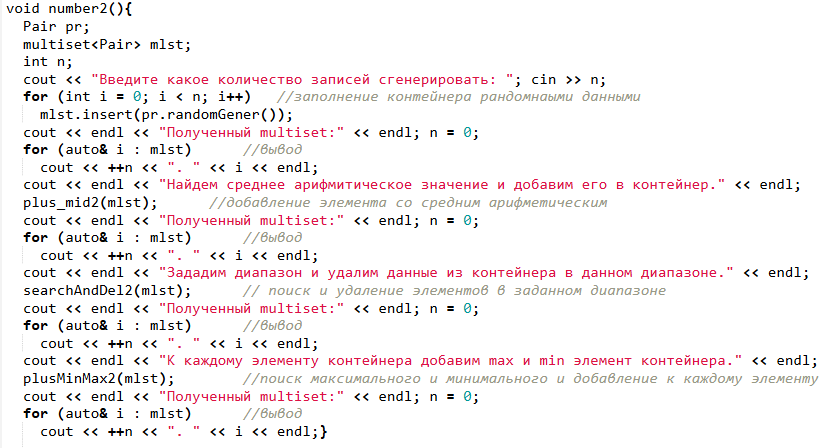


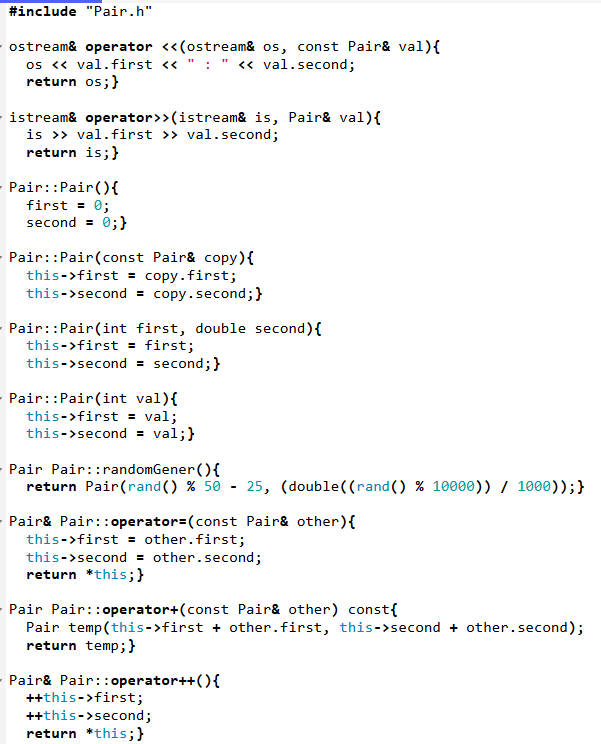
Диаграмма:

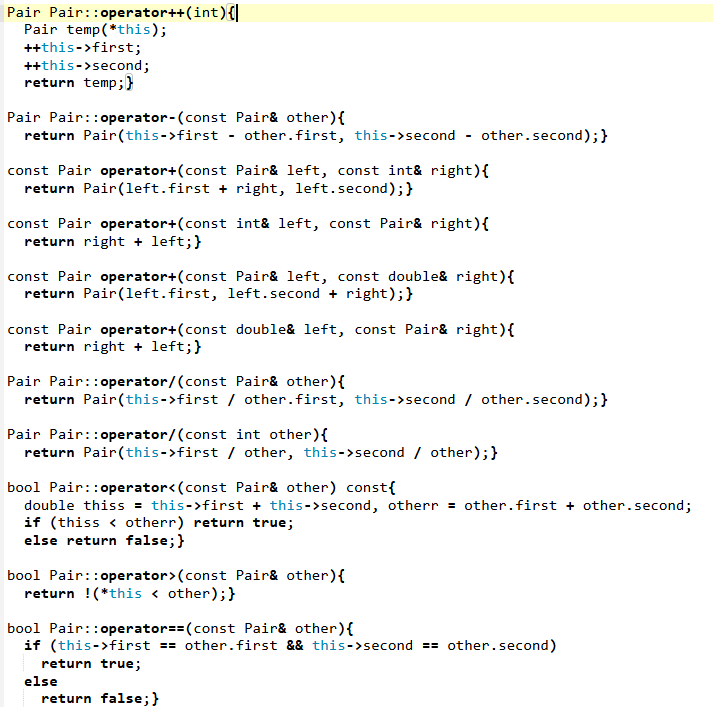


Код:

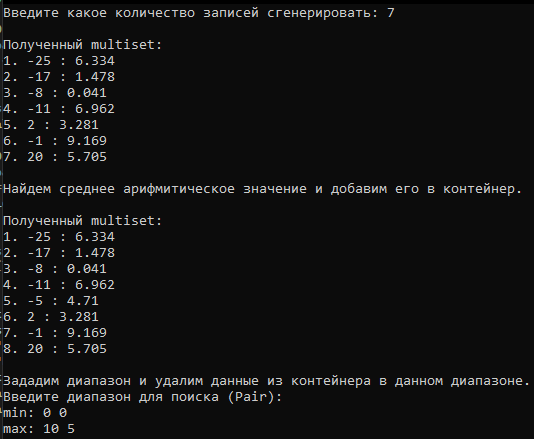


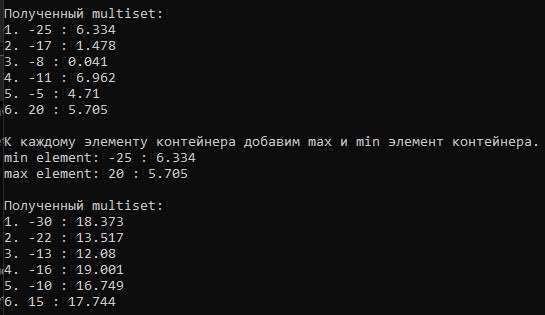




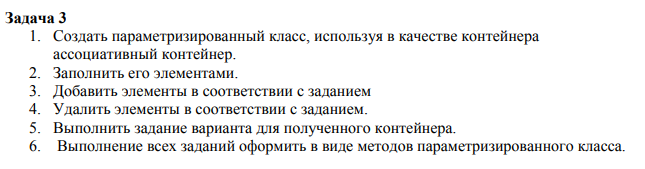


Вывод:





Постановка задачи:



Задача:



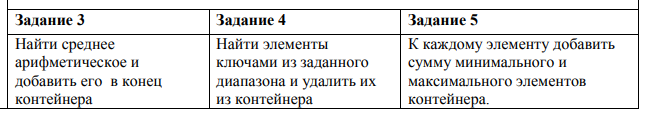
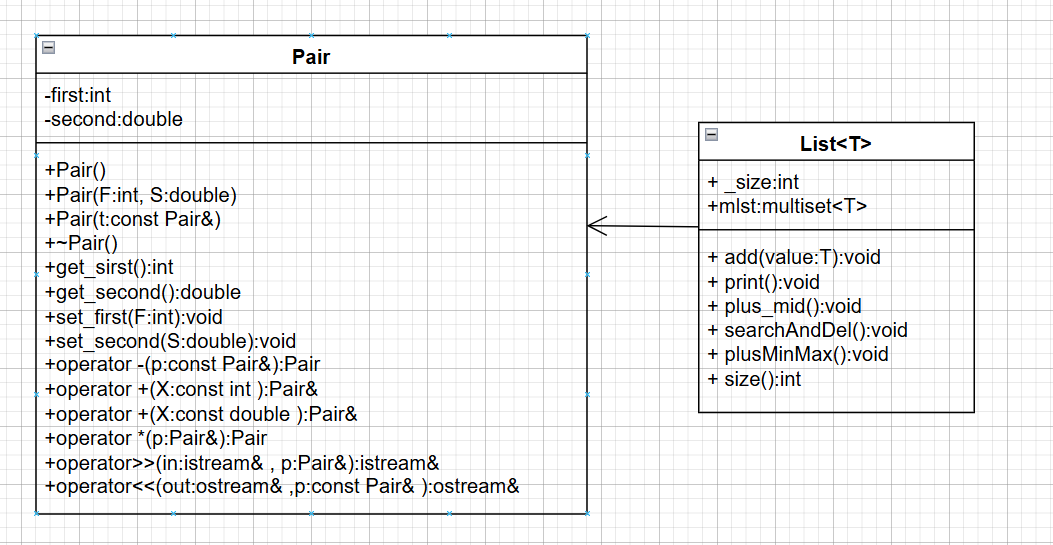
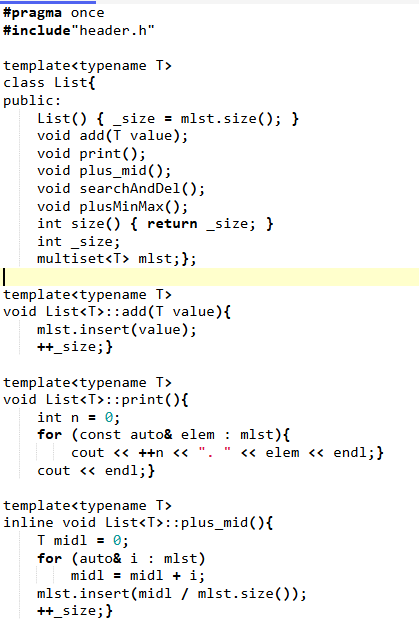
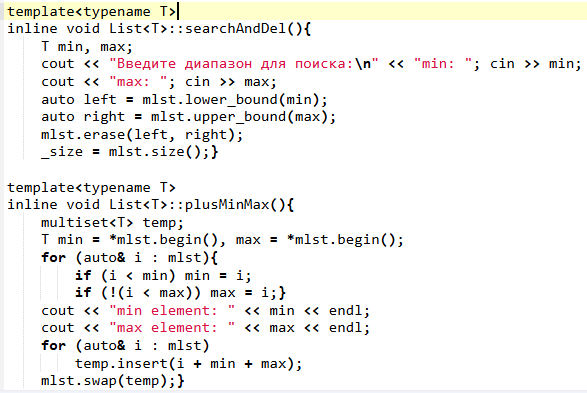


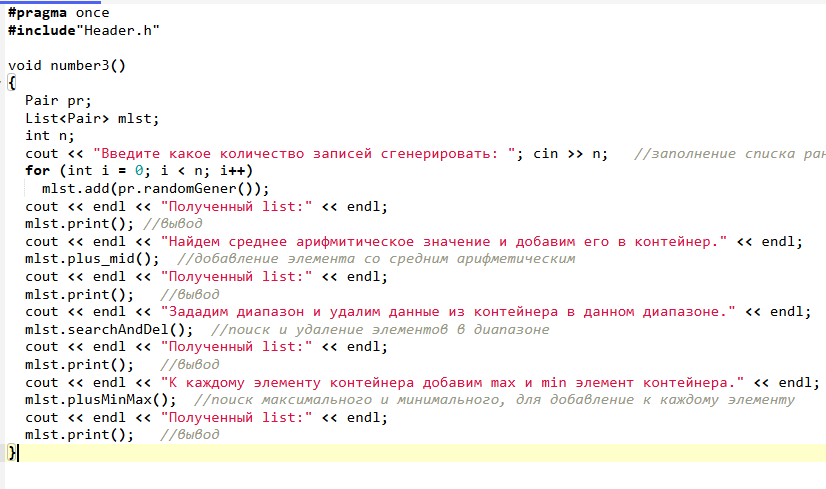
Диаграмма:



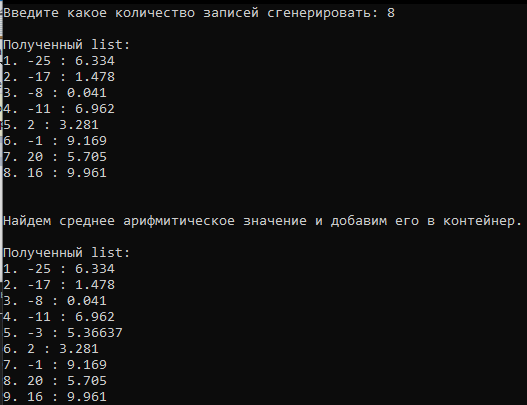
Код:

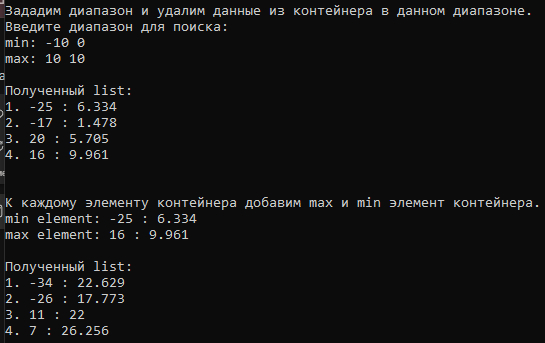




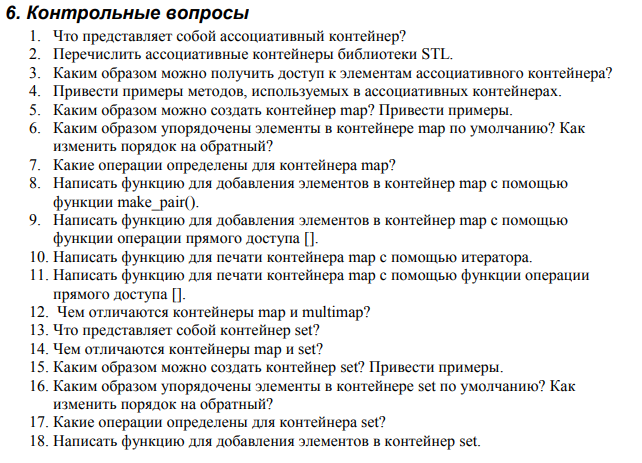


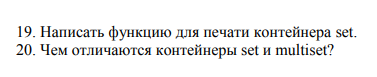
Вывод:





Контрольные вопросы:





1. Ассоциативный контейнер - это контейнер, который содержит упорядоченный набор элементов, представленных в форме пар ключ-значение. Элементы располагаются в контейнере в соответствии с заданным отношением порядка, определенным в компараторе. Ключи уникальны в контейнере, т.е. в контейнере не может быть двух элементов с одинаковым ключом. Основным преимуществом ассоциативных контейнеров является возможность быстрого поиска элемента по ключу благодаря использованию бинарного дерева поиска (например, красно-черного дерева) для хранения элементов. Примерами ассоциативных контейнеров в STL являются map и set.
2. .

* set - контейнер, хранящий уникальные элементы в отсортированном порядке. Все элементы в set уникальны, то есть не могут дублироваться.
* map - контейнер, хранящий пары ключ-значение, отсортированные по ключу. Ключи в map должны быть уникальными.
* multiset - контейнер, хранящий уникальные элементы в отсортированном порядке, но допускающий дублирование элементов.
* multimap - контейнер, хранящий пары ключ-значение, отсортированные по

1. Для доступа к элементам ассоциативного контейнера в STL используются итераторы. В отличие от последовательных контейнеров, в ассоциативных контейнерах элементы хранятся не в порядке их добавления, а в отсортированном порядке на основе ключа. Поэтому для доступа к элементам по индексу, как в последовательных контейнерах, в ассоциативных контейнерах используют итераторы. С помощью итераторов можно получить доступ к ключу и соответствующему ему значению элемента контейнера.
2. .
   1. insert(): добавляет элемент в контейнер.
   2. erase(): удаляет элемент из контейнера.
   3. find(): ищет элемент по заданному ключу.
   4. count(): возвращает количество элементов с заданным ключом в контейнере.
   5. size(): возвращает количество элементов в контейнере.
   6. empty(): возвращает значение true, если контейнер пуст, и false в противном случае.
   7. begin(): возвращает итератор, указывающий на первый элемент в контейнере.
   8. end(): возвращает итератор, указывающий на элемент следующий за последним \* элементом контейнера.
   9. lower\_bound(): возвращает итератор на первый элемент в контейнере, не меньший \* заданного ключа.
   10. upper\_bound(): возвращает итератор на первый элемент в контейнере, больший заданного ключа.
   11. equal\_range(): возвращает диапазон элементов в контейнере, соответствующих заданному ключу.
3. Каким образом можно создать контейнер map? Привести примеры
   1. ⦁ С помощью конструктора по умолчанию:



* 1. ⦁ С помощью списка инициализации:



* 1. ⦁ С помощью пары итераторов:

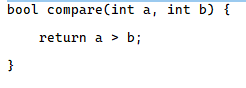


* 1. ⦁ С помощью списка пар ключ-значение:



1. Элементы отсортированы в порядке возрастания ключей.

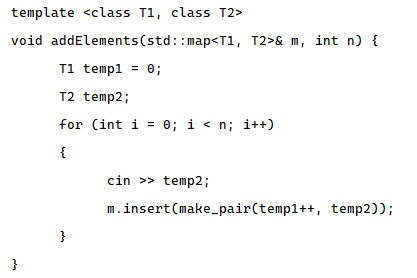
Чтобы изменить порядок, можно определить пользовательскую функцию сравнения, которая будет сравнивать ключи в обратном порядке. Например:



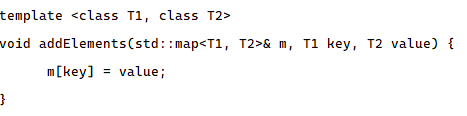
Затем, мы можем создать map следующим образом:



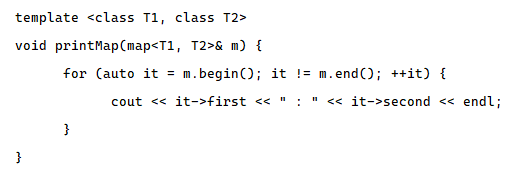
1. Контейнер map поддерживает операции добавления и удаления элементов, поиска и доступа к элементам по ключу, а также проверки наличия элементов в контейнере. Кроме того, контейнер map поддерживает итераторы для обхода содержимого.
2. .



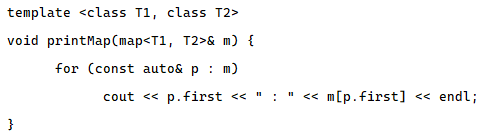
1. .



1. .



1. .



1. Отличие между map и multimap заключается в том, что map хранит только уникальные ключи и соответствующие значения, тогда как multimap может хранить несколько значений для одного и того же ключа. Другими словами, map — это контейнер с уникальными ключами, а multimap - контейнер с неуникальными ключами.
2. Контейнер set - упорядоченное множество уникальных элементов. Он реализован в виде бинарного дерева поиска и обеспечивает быстрый доступ, вставку и удаление элементов в отсортированном порядке.
3. Контейнер map предназначен для хранения пары "ключ-значение", где каждый ключ уникален, а контейнер set используется для хранения уникальных элементов, без пары "ключ-значение".

Таким образом, map используется для хранения и доступа к значению по ключу, а set используется для хранения элементов в отсортированном порядке и быстрого поиска элементов по значению.

1. Каким ообразом можно создать контейнер set? Привести примеры.
   1. Создание пустого контейнера с помощью конструктора по умолчанию:



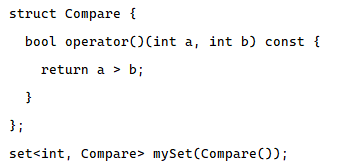
* 1. Создание с заданными начальными значениями с помощью списка инициализации:



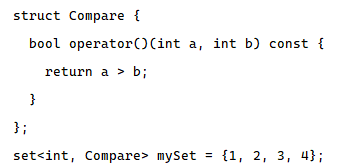
* 1. Создание с помощью диапазона значений другого контейнера:



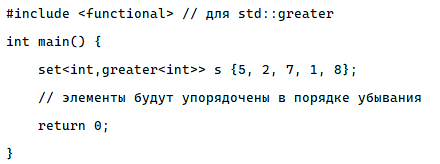
* 1. Cоздание пустого контейнера с заданным компаратором:



* 1. Создание с заданными начальными значениями и компаратором:

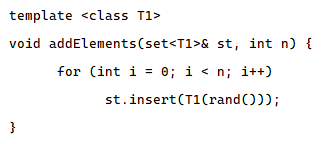


1. Элементы в контейнере set упорядочены по возрастанию. Чтобы изменить порядок на убывание, можно задать компаратор при создании контейнера, который будет сравнивать элементы в обратном порядке. Например:

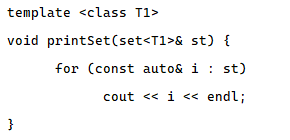


Здесь std::greater<int> - это функциональный объект, который сравнивает элементы в порядке убывания. Он передается вторым параметром шаблона контейнера set.

1. Какие методы определены для контейнера set?
   1. insert() - добавляет элемент в контейнер
   2. erase() - удаляет элемент из контейнера по значению или по итератору
   3. find() - ищет элемент в контейнере и возвращает итератор на найденный элемент, либо итератор на конец контейнера, если элемент не найден
   4. size() - возвращает количество элементов в контейнере
   5. empty() - возвращает true, если контейнер пуст, иначе – false
   6. clear() - удаляет все элементы из контейнера
2. .



1. .



1. Чем отличается контейнер set и multiset?

Отличие между ними заключается в том, что set может хранить только уникальные элементы, а multiset может хранить несколько одинаковых элементов.