

std::unordered_map

Контейнер **std::unordered_map** в C++ представляет собой реализацию хеш-таблицы, которая представлена в виде отображения ключей на значения. Ключи могут быть любого типа, для которого определен хеш-функция и оператор сравнения, а значения могут быть любого типа.

Хеш-таблица - это структура данных, которая использует хеш-функцию для быстрого поиска значений в таблице. Хеш-функция преобразует ключ в индекс таблицы, где находится значение. Если в таблице есть несколько значений с одним и тем же индексом, они хранятся в списке связанных элементов.

std::unordered_map реализован как массив корзин, где каждая корзина представляет собой связанный список пар ключ-значение. Ключи хешируются, чтобы определить, в какую корзину поместить значение, а затем используются для быстрого поиска значения. Поиск, вставка и удаление элементов в **std::unordered_map** имеют среднюю сложность $O(1)$, хотя в худшем случае (когда все элементы попадают в одну корзину) сложность может достигать $O(n)$.

Операции, которые можно выполнить над **std::unordered_map**, включают:

- **insert(key, value)** - добавляет элемент в отображение.
- **erase(key)** - удаляет элемент из отображения по ключу.
- **find(key)** - возвращает итератор на элемент с указанным ключом.
- **clear()** - удаляет все элементы из отображения.
- **size()** - возвращает количество элементов в отображении.

Кроме того, **std::unordered_map** предоставляет методы для обхода элементов в произвольном порядке, такие как **begin()**, **end()**, **cbegin()**, **cend()**.

Стандарт C++ также предоставляет другие хеш-контейнеры, такие как **std::unordered_set** и **std::unordered_multimap**, которые основаны на том же принципе хеш-таблицы, что и **std::unordered_map**. Однако, они используются для решения других задач и имеют свои особенности.

Что такое хэш-функция?

Хеш-функция - это функция, которая преобразует произвольный входной набор данных (например, строку, число или любой другой объект) в некоторое фиксированное значение определенной длины, называемое хеш-значением или хеш-кодом. Обычно, длина хеш-значения является постоянной, независимо от длины входных данных.

Хеш-функции используются во многих алгоритмах и приложениях для быстрого поиска, проверки целостности данных, шифрования и других целей. Они имеют несколько свойств, которые делают их полезными в этих приложениях:

1. Для одного и того же входного значения хеш-функция всегда выдает одинаковый результат.
2. Хеш-функция должна быть быстрой, чтобы обработка данных не занимала слишком много времени.
3. Даже небольшие изменения во входных данных должны приводить к значительным изменениям в хеш-значении.

В контексте хеш-таблиц, хеш-функция используется для определения индекса, по которому элемент будет храниться в таблице. Хорошая хеш-функция должна распределять элементы равномерно по всей таблице, чтобы минимизировать количество конфликтов, когда несколько элементов имеют одинаковый индекс.

Хеш-функции могут быть реализованы для разных типов данных, и различные алгоритмы могут использоваться для их вычисления. В C++ и других языках программирования есть встроенные хеш-функции для стандартных типов данных, а также пользовательские хеш-функции могут быть созданы для любых пользовательских типов данных.