

**multimap** в C++ является структурой данных, которая представляет собой упорядоченный ассоциативный контейнер, состоящий из пар ключ-значение, где каждый ключ может иметь несколько значений.

Эта структура данных может использоваться во многих случаях, в которых требуется упорядоченное хранение пар ключ-значение. Некоторые примеры использования **multimap**:

- Хранение множества элементов, которые могут иметь одинаковый ключ. Например, список студентов, отсортированный по фамилиям, но в случае совпадения фамилий для каждого студента может быть несколько оценок.
- Реализация многократного вхождения элементов в ассоциативном контейнере. Например, список посещений сайта для каждого пользователя, отсортированный по дате.
- Реализация мультимножества, в котором ключ может повторяться. Например, хранение информации об акциях на бирже, где каждый тикер может иметь несколько цен на покупку и продажу.

Использование **multimap** обычно предпочтительно в случае, когда необходимо хранить неупорядоченные данные, которые могут иметь повторяющиеся ключи, и когда требуется выполнить операции вставки, удаления и поиска в хэш-таблицах, например, недоступны. Однако, если количество данных относительно небольшое, то для этих целей также могут использоваться другие структуры данных, такие как **vector**, **deque**, **list** или **set**.

Некоторые методы MultiMap:

1. **insert()** - добавляет элемент в MultiMap. Если элемент уже существует, он не добавляется.
2. **emplace()** - создает и добавляет элемент в MultiMap.
3. **find()** - ищет элемент по ключу и возвращает итератор на него. Если элемент не найден, возвращает итератор на конец MultiMap.
4. **count()** - возвращает количество элементов в MultiMap, которые имеют указанный ключ.
5. **lower\_bound()** - возвращает итератор на первый элемент в MultiMap с ключом, не меньшим, чем указанный.
6. **upper\_bound()** - возвращает итератор на первый элемент в MultiMap с ключом, большим, чем указанный.
7. **equal\_range()** - возвращает диапазон элементов с указанным ключом.

8. **erase()** - удаляет элемент из MultiMap по указанному ключу или итератору.
9. **clear()** - удаляет все элементы из MultiMap.
10. **size()** - возвращает количество элементов в MultiMap.
11. **empty()** - возвращает true, если MultiMap пуст, и false в противном случае.
12. **swap()** - обменивает содержимое двух MultiMap.