

Библиотека STL

Стандартная библиотека предоставляет различные типобезопасные контейнеры для хранения коллекций связанных объектов.

Контейнеры — это шаблоны классов. При объявлении переменной контейнера указывается тип элементов, которые будет содержать контейнер. Контейнеры могут создаваться с использованием списков инициализаторов. Они содержат функции-члены для добавления и удаления элементов и выполнения других операций.

Итерация элементов в контейнере и доступ к отдельным элементам осуществляются с помощью итераторов. Можно использовать итераторы явно, с помощью их функций-членов и операторов, а также глобальных функций. Можно также использовать их неявно, например с помощью цикла `range-for`. Итераторы для всех контейнеров STL имеют общий интерфейс, но каждый контейнер определяет собственные специализированные итераторы.

Контейнеры можно разделить на три категории: последовательные контейнеры, ассоциативные контейнеры и контейнеры-адаптеры.

Последовательные контейнеры

Последовательные контейнеры поддерживают указанный пользователем порядок вставляемых элементов.

Контейнер **vector** ведет себя как массив, но может автоматически увеличиваться по мере необходимости. Он поддерживает прямой доступ и связанное хранение и имеет очень гибкую длину. По этим и многим другим причинам контейнер **vector** является наиболее предпочтительным последовательным контейнером для большинства областей применения. Если вы сомневаетесь в выборе вида последовательного контейнера, начните с использования вектора.

Контейнер **array** обладает некоторыми преимуществами контейнера **vector**, однако его длина не обладает такой гибкостью.

Контейнер **deque** (двусторонняя очередь) обеспечивает быструю вставку и удаление в начале и в конце контейнера. Он, как и контейнер **vector**, обладает преимуществами прямого доступа и гибкой длины, но не обеспечивает связанное хранение.

Контейнер **list** — это двунаправленный список, который обеспечивает двунаправленный доступ, быструю вставку и удаления в любом месте контейнера, но не поддерживает прямой доступ к элементам контейнера.

Контейнер **forward_list** — однонаправленный список. Это версия контейнера **list** только с доступом в прямом направлении.

Ассоциативные контейнеры

В ассоциативных контейнерах элементы вставляются в предварительно определенном порядке — например, с сортировкой по возрастанию. Также доступны неупорядоченные ассоциативные контейнеры. Ассоциативные контейнеры можно объединить в два подмножества: сопоставления (`set`) и наборы (`map`).

Контейнер **map**, который иногда называют словарем, состоит из пар "ключ-значение". Ключ используется для упорядочивания последовательности, а значение связано с ключом. Например, **map** может содержать ключи, представляющие каждое уникальное ключевое слово в тексте, и соответствующие значения, которые обозначают количество повторений каждого слова в тексте. **unordered_map** — это неупорядоченная версия **map**.

set — это контейнер уникальных элементов, упорядоченных по возрастанию. Каждое его значение также является и ключом. **unordered_set** — это неупорядоченная версия **set**.

Контейнеры **map** и **set** разрешают вставку только одного экземпляра ключа или элемента. Если необходимо включить несколько экземпляров элемента, следует использовать контейнер **multimap** или **multiset**. Неупорядоченные версии этих контейнеров — **unordered_multimap** и **unordered_multiset**.

Упорядоченные контейнеры `map` и `set` поддерживают двунаправленные итераторы, а их неупорядоченный аналоги — итераторы с перебором в прямом направлении

Контейнеры-адаптеры

Контейнер-адаптер — это разновидность последовательного или ассоциативного контейнера, который ограничивает интерфейс для простоты и ясности. Контейнеры-адаптеры не поддерживают итераторы.

Контейнер **queue** соответствует семантике FIFO (первым поступил — первым обслужен). Первый элемент, который *отправляется*, то есть вставляется, в очередь, должен быть первым элементом, *извлекаемым* из очереди.

Контейнер **priority_queue** упорядочен таким образом, что первым в очереди всегда оказывается элемент с наибольшим значением.

Контейнер **stack** соответствует семантике LIFO (последним поступил — первым обслужен). Последний элемент, отправленный в стек, становится первым извлекаемым элементом.