21.2 — STL containers overview

ALEX SEPTEMBER 19, 2021

By far the most commonly used functionality of the STL library are the STL container classes. If you need a quick refresher on container classes, check out lesson 23.6 -- Container classes (https://www.learncpp.com/cpp-tutorial/container-classes/).

The STL contains many different container classes that can be used in different situations. Generally speaking, the container classes fall into three basic categories: Sequence containers, Associative containers, and Container adapters. We'll just do a quick overview of the containers here.

Sequence Containers

Sequence containers are container classes that maintain the ordering of elements in the container. A defining characteristic of sequence containers is that you can choose where to insert your element by position. The most common example of a sequence container is the array: if you insert four elements into an array, the elements will be in the exact order you inserted them.

As of C++11, the STL contains 6 sequence containers: std::vector, std::deque, std::array, std::list, std::forward_list, and std::basic_string.

- If you've ever taken physics, you probably are thinking of a vector as an entity with both magnitude and direction. The unfortunately named **vector** class in the STL is a dynamic array capable of growing as needed to contain its elements. The vector class allows random access to its elements via operator[], and inserting and removing elements from the end of the vector is generally fast.
The following program inserts 6 numbers into a vector and uses the overloaded [] operator to access them in order to print them.

```
#include <vector>
#include <iostream>

int main()
{

    std::vector<int> vect;
    for (int count=0; count < 6; ++count)
        vect.push_back(10 - count); // insert at end of array

    for (int index=0; index < vect.size(); ++index)
        std::cout << vect[index] << ' ';

    std::cout << '\n';
}
```

This program produces the result:
10 9 8 7 6 5

- The **deque** class (pronounced "deck") is a double-ended queue class, implemented as a dynamic array that can grow from both ends.

```
#include <iostream>
#include <deque>

int main()
{
    std::deque<int> deq;
    for (int count=0; count < 3; ++count)
    {
        deq.push_back(count); // insert at end of array
        deq.push_front(10 - count); // insert at front of array
    }

    for (int index=0; index < deq.size(); ++index)
        std::cout << deq[index] << ' ';

    std::cout << '\n';
}
```

This program produces the result:

8 9 10 0 1 2

- A **list** is a special type of sequence container called a doubly linked list where each element in the container contains pointers that point at the next and previous elements in the list. Lists only provide access to the start and end of the list -- there is no random access provided. If you want to find a value in the middle, you have to start at one end and "walk the list" until you reach the element you want to find. The advantage of lists is that inserting elements into a list is very fast if you already know where you want to insert them. Generally iterators are used to walk through the list.
  We'll talk more about both linked lists and iterators in future lessons.
- Although the STL **string** (and wstring) class aren't generally included as a type of sequence container, they essentially are, as they can be thought of as a vector with data elements of type char (or wchar).

Associative Containers

Associative containers are containers that automatically sort their inputs when those inputs are inserted into the container. By default, associative containers compare elements using operator<.

- A **set** is a container that stores unique elements, with duplicate elements disallowed. The elements are sorted according to their values.
- A **multiset** is a set where duplicate elements are allowed.
- A **map** (also called an associative array) is a set where each element is a pair, called a key/value pair. The key is used for sorting and indexing the data, and must be unique. The value is the actual data.
- A **multimap** (also called a dictionary) is a map that allows duplicate keys. Real-life dictionaries are multimaps: the key is the word, and the value is the meaning of the word. All the keys are sorted in ascending order, and you can look up the value by key. Some words can have multiple meanings, which is why the dictionary is a multimap rather than a map.

Container Adapters

Container adapters are special predefined containers that are adapted to specific uses. The interesting part about container adapters is that you can choose which sequence container you want them to use.

- A **stack** is a container where elements operate in a LIFO (Last In, First Out) context, where elements are inserted (pushed) and removed (popped) from the end of the container. Stacks default to using deque as their default sequence container (which seems odd, since vector seems like a more natural fit), but can use vector or list as well.
- A **queue** is a container where elements operate in a FIFO (First In, First Out) context, where elements are inserted (pushed) to the back of the container and removed (popped) from the front. Queues default to using deque, but can also use list.
- A **priority queue** is a type of queue where the elements are kept sorted (via operator<). When elements are pushed, the element is sorted in the queue. Removing an element from the front returns the highest priority item in the priority queue.

Next lesson
21.3 STL iterators overview

Back to table of contents

Previous lesson
21.1 The Standard Library

Leave a comment...

Name*

Email*

Notify me about replies:

POST COMMENT

Find a mistake? Leave a comment above!

Avatars from https://gravatar.com/ are connected to your provided email address.

90 COMMENTS

Newest