# STL бібліотека

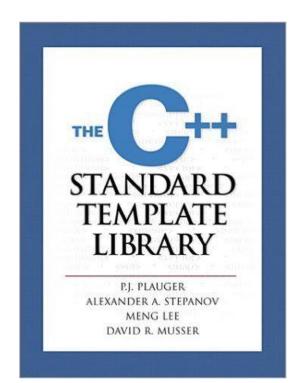
<u>C++</u>



Стандартна бібліотека шаблонів (STL) - це набір класів шаблонів С ++ для забезпечення загальних структур даних та функцій програмування, таких як списки, стеки, масиви тощо. Це бібліотека класів контейнерів, алгоритмів та ітераторів.

#### Компоненти STL:

- Алгоритми
- Контейнери
- Функтори
- Ітератори



Контейнер - це об'єкт, що зберігає колекцію інших об'єктів (його елементів).

Контейнер керує місцем для зберігання своїх елементів і надає функції-члени для доступу до них, безпосередньо або через ітератори (посилальні об'єкти з властивостями, подібними до вказівників ).

## Колекції

Для використання колекції в коді використовується директива #include <T>,

де Т - назва колекції

vector - колекція елементів, збережених в масиві, що змінюється в міру необхідності розміру

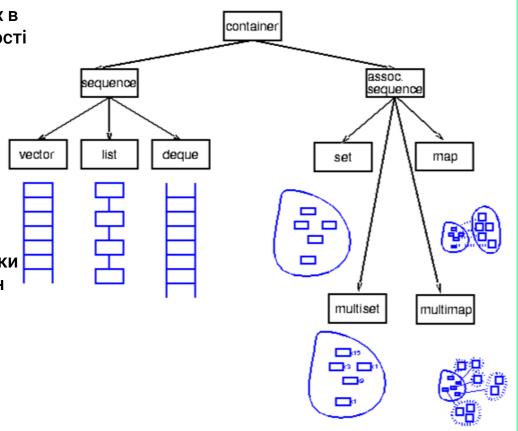
list - зберігає елементи в вигляді зв'язаного списку;

map - зберігає пари виду <const Key, T>, тобто кожен елемент - це пара виду <ключ, значення>

set - відсортована колекція одних тільки значень, д при цьому унікальних - кожен ключ може зустрітися тільки один раз

multimap - map, в якому відсутня умова унікальності ключа

multiset - set з відсутністю умови унікальності ключа.



### forward\_list

```
prev head curr
NULL 1 2 NULL
```

```
while (current != NULL)
{
    next = current->next;
    current->next = prev;
    prev = current;
    current = next;
}
*head_ref = prev;
```

Контейнер послідовностей, що дозволяють постійно виконувати операції вставки та стирання в будь-якому місці послідовності.

Однопов'язані списки можуть зберігати кожен з елементів, які вони містять, у різних і не пов'язаних між собою місцях зберігання.

```
#include <iostream>
#include <forward list>
int main()
{std::forward list<int> numbers
= \{ 1, 2, 3, 4, 5 \};
        int first = numbers.front();
        for (int n : numbers)
               std::cout << n << "\t";
```

set

## set - відсортована колекція одних тільки значень,д при цьому унікальних - кожен ключ може зустрітися тільки один раз

```
#include <iostream>
#include <set>
#include <iterator>
using namespace std;
int main()
    set<char> mySet;
   mySet.insert('s');
   mySet.insert('e');
   mySet.insert('t');
   mySet.insert('t');
   copy( mySet.begin(), mySet.end(), ostream iterator<char>(cout, " "));
   return 0;
```

### Алгоритми

Алгоритм заголовка визначає набір функцій, спеціально розроблених для використання на діапазонах елементів. Вони діють на контейнери та забезпечують засоби для різних операцій щодо вмісту контейнерів.

#### Алгоритми

- Сортування
- Пошук
- Важливі алгоритми STL
- Корисні алгоритми масиву
- Операції розділів
- Числовий
- клас valarray

# Ітератори

# Функтори

The C++ standard library generic containers Size changes at run time? Yes No Elements all the same type? Elements keep insertion ordering? Fixed size Variable size No Yes Yes Homogeneous Anywhere Fast insert and remove? Heterogeneous Sequential access Both ends One end only No std::deque std::vector std::pair Access first only Accessing elements? >Restricted Access last only std::stack std::array std::tuple **Bidirectional** Random access std::queue Sequential access Forward Sequential access std::list std::forward\_list The data itself What is the key? Keyed access std::unordered set std::set std::priority\_queue Separate key std::unordered\_multiset std::multiset Can traverse in sorted order? Cannot traverse std::unordered map std::map Yes Sorted Unsorted std::unordered multimap std::multimap