

C++

Урок 21

A hand-drawn blue oval frame with a slightly irregular, sketchy border, centered on the page. It contains the word 'Повторение' in bold black text.

Повторение

Что такое лямбда-выражение?

Где применяется?

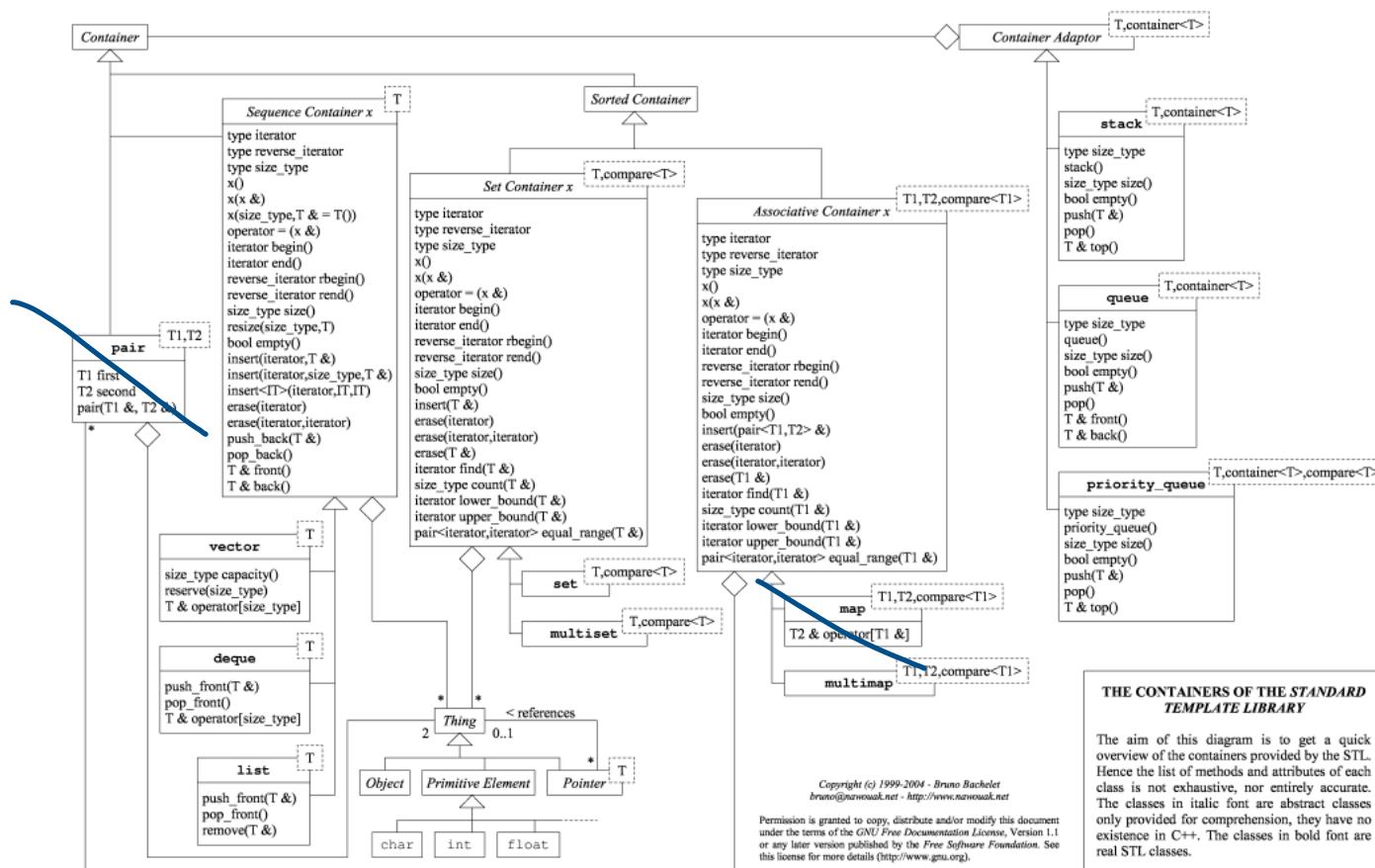
Что лежит под капотом?

Что такое список захвата?

**В каком случае передаем
параметры по ссылке?**

STL

Контейнеры есть стандартные структуры данных, такие как список (**list**), вектор (**vector**), словарь (**map**) и многие другие.



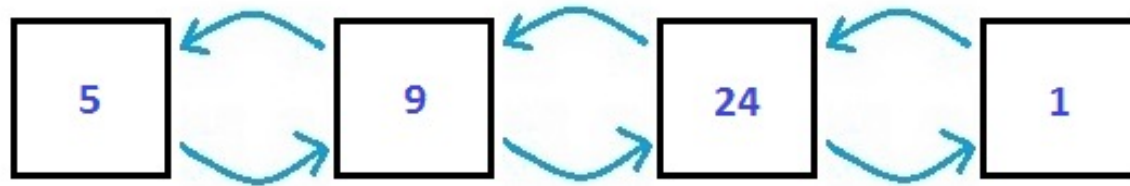


list

Список

Список - структура данных, которая построена на **двусвязных списках**.

Любой элемент знает только о **предыдущем** и о **следующем** элементах.



Список

Особенности

- Быстрое **добавление** и **удаление** значений.
(кроме конца)
- **Медленное** обращение к элементам,
находящимся в **центре**.

Список

```
#include <list>
```

- Подключаем библиотеку

```
int main()
{
    // list<тип данных><имя контейнера>;
    list<double> weights = {1.34, 0.12, 4.12, 2.12}; // объявили + инициализировали список
}
```

Создание списка

Список

```
int main()
{
    // list<тип данных><имя контейнера>;
    list<double> weights = {1.34, 0.12, 4.12, 2.12}; // объявили + инициализировали список

    for (auto &data : weights)
    {
        cout << data << " ";
    }

    cout << endl;

    for (auto it = weights.begin(); it != weights.end(); it++)
    {
        cout << *it << " ";
    }

    cout << endl;

    for_each(weights.begin(), weights.end(), [](auto val)
        { cout << val << " "; });
}
```

Варианты прохода по
контейнеру

Задача

Задача: введите/выведите данные в список.

Задача

Задача: заполните список числами и выведите “положительных”, если список содержит больше положительных чисел, “отрицательных”, если отрицательных или “ровно” при одинаковом кол-ве.

Список

Особенности

- Для итераторов списков нельзя применить арифметические операции;
- Нельзя взять индекс;
- Можно: `advance()`, `next()`, `prev()`.

Список

Передвигаем итератор на нужное нам место.

```
advance(<итератор>, <значение>);
```

```
next(<итератор>, <значение>);
```

```
prev(<итератор>, <значение>);
```

Отличие:

- advance ничего не возвращает;
- в next и prev по умолчанию стоит вторым аргументом 1.

Список

Методы в **list**:

Element access	
front	access the first element (public member function)
back	access the last element (public member function)
Iterators	
begin cbegin (C++11)	returns an iterator to the beginning (public member function)
end cend (C++11)	returns an iterator to the end (public member function)
rbegin crbegin (C++11)	returns a reverse iterator to the beginning (public member function)
rend crend (C++11)	returns a reverse iterator to the end (public member function)
Capacity	
empty	checks whether the container is empty (public member function)
size	returns the number of elements (public member function)
max_size	returns the maximum possible number of elements (public member function)
Modifiers	
clear	clears the contents (public member function)
insert	inserts elements (public member function)
emplace (C++11)	constructs element in-place (public member function)
erase	erases elements (public member function)
push_back	adds an element to the end (public member function)
emplace_back (C++11)	constructs an element in-place at the end (public member function)
pop_back	removes the last element (public member function)
push_front	inserts an element to the beginning (public member function)
emplace_front (C++11)	constructs an element in-place at the beginning (public member function)
pop_front	removes the first element (public member function)
resize	changes the number of elements stored (public member function)
swap	swaps the contents (public member function)
Operations	
merge	merges two sorted lists (public member function)
splice	moves elements from another list (public member function)
remove remove_if	removes elements satisfying specific criteria (public member function)
reverse	reverses the order of the elements (public member function)
unique	removes consecutive duplicate elements (public member function)
sort	sorts the elements (public member function)

Задача

Задача: Удалить элемент в векторе равный 42.

Задача

Задача: Удалить все элементы элементы, которые равны введенному.

Задача

Задача: На вход подаются id студентов, которые посетили лекцию, какие-то id записаны повторно, удалите повторяющиеся элементы и посчитайте кол-во студентов, побывавших на лекции.

Список

Плюсы

- **Вставка** элемента в любом месте списка.
- **Удаление** любого элемента.

Минусы

- Сложность поиска элемента по его номеру (индексу) $O(n)$.
- На указатель выделяется дополнительная память.
- Элементы списка могут располагаться в памяти дроблено (сложности при кэшировании).



Стек

Список

Стек - структура данных, которая построена на **односвязном списке**.

Принцип **LIFO** = Last In First Out, «последним пришел, первым вышел».

Любой элемент знает только о **следующем** элементе.





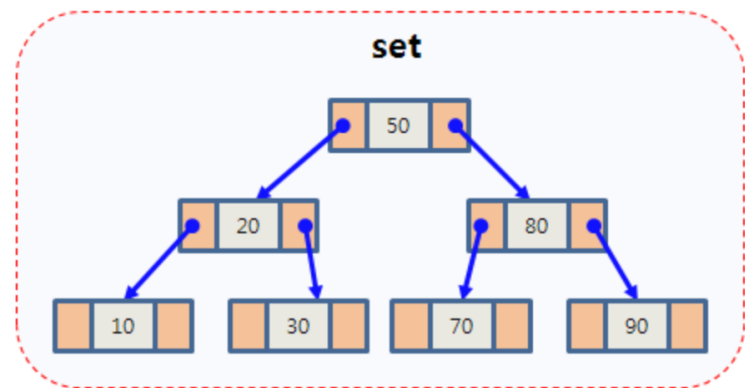
Set

Множество

set — это контейнер, который автоматически сортирует добавляемые элементы в порядке возрастания.

При добавлении одинаковых значений, set будет хранить только **один** его экземпляр.

По другому его еще называют **МНОЖЕСТВОМ**.



Множество

Применение

сортирует добавляемые

A hand-drawn blue oval border surrounds the word "Multiset".

Multiset

Мультимножество

multiset — это контейнер, который также будет содержать элементы в отсортированном порядке при добавлении, но он хранит **повторяющиеся** элементы, по сравнению с множеством `set`.

Часто его называют **мультимножество**.

