

РАЗРАБОТКА НА C++

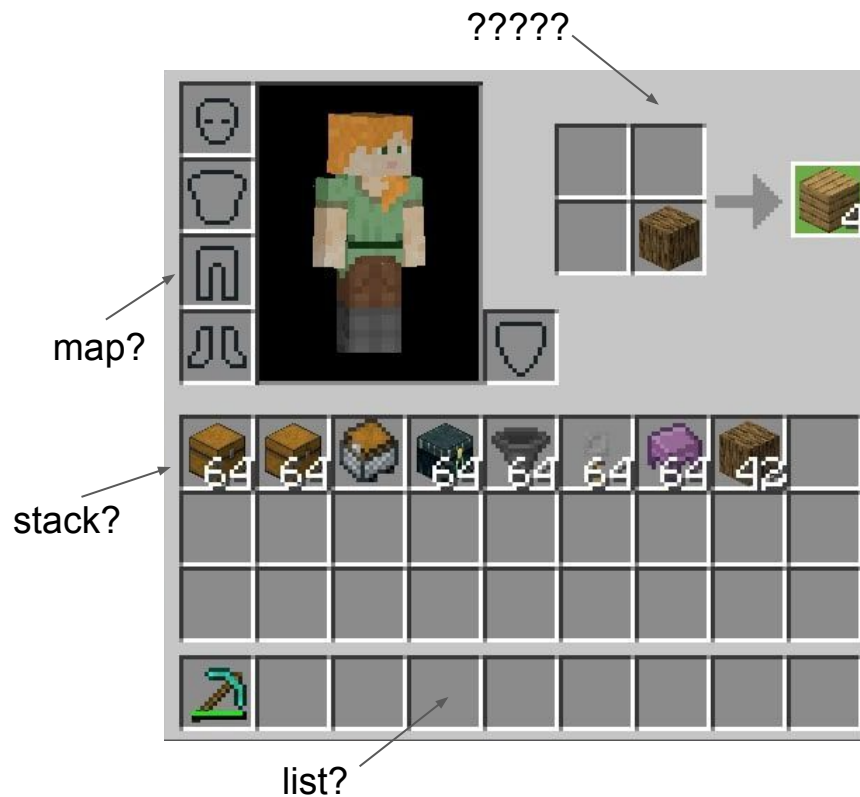
Урок 3. Контейнеры STL

План

1. Что такое Standard Template Library?
2. Что такое контейнер?
3. Вектор, итераторы
4. Стеки
5. Очереди
6. Деки

Standard Template Library (STL)

Standard Template Library - это набор согласованных обобщённых алгоритмов, контейнеров, средств доступа к их содержимому и различных вспомогательных функций в C++



Что такое контейнер?



В современных проектах, написанных на с++, очень часто появляются такие понятия как *“контейнер”*, *“алгоритмическая сложность”*, *“итератор”* и многие другие страшные для новичков слова.

Если сравнивать это с реальной жизнью можно привести в пример огромный гараж какого-то очень богатого человека, у которого есть велосипеды, машины, вертолеты, грузовики, лыжи и вообще все виды транспорта. И теперь нам нужно выбрать подходящий транспорт для перевозки багажа.

Контейнеры STL

Библиотека STL - это огромный гараж

Контейнер - это “транспорт”

Контейнер - это тип, который реализует в себе некоторую структуру хранения и обработки данных.

От программиста зависит правильность применения контейнеров в проектах. Конечно в маленьких проектах будет незаметное падение скорости или засорение памяти, но как только проект станет больше уже будет поздно что-то менять.



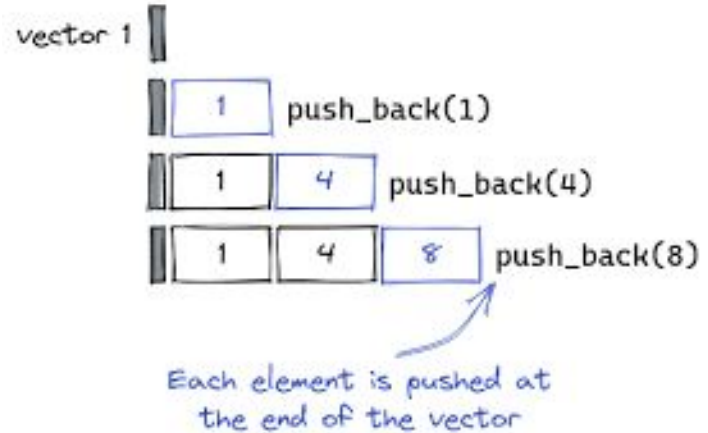
Что такое вектор?

Это контейнер, созданный на основе обычного **динамического массива**, который может самостоятельно менять размер во время выполнения кода.

Вектор обладает множеством полезных методов для поиска, удаления и добавления элементов

```
#include <vector>
```

```
vector<int> v1;
```



Работа с вектором

Объявление вектора:

```
vector<int> vec; // создание пустого вектора
```

```
vector<char> vec2{'s'}; // создание вектора, содержащего элемент 's'
```

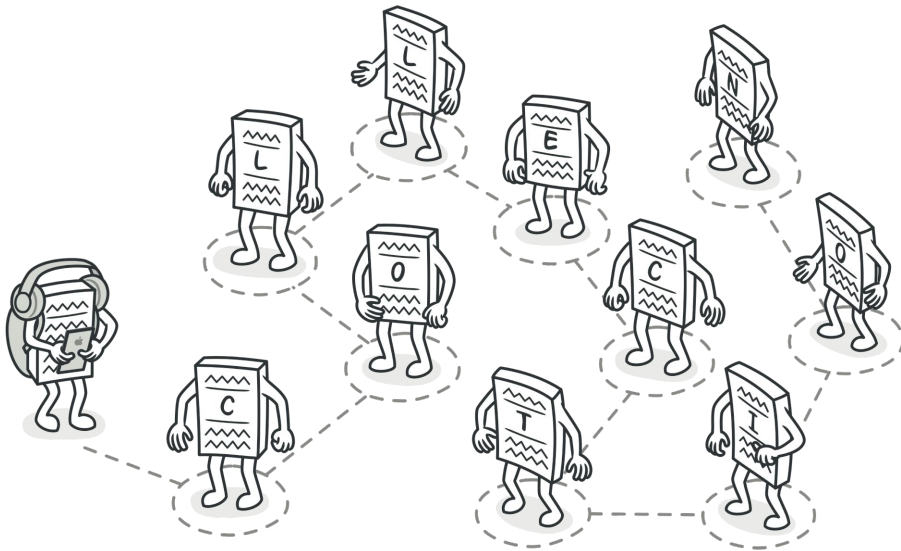
```
vector<double> vec3(2); // создание пустого вектора из 2-х элементов
```

```
vector<float> vec4(3) = {1.2, 4.1, 5}; // создание вектора из 3-х элементов
```

Что такое итератор?

Это объект, который может перебирать элементы в контейнерах стандартной библиотеки C++ и предоставлять доступ к отдельным элементам.

У многих контейнеров есть крайне полезные встроенные функции (find, sort, insert и т.д.) которые будут работать только с итераторами, далее мы рассмотрим как это происходит и заметим, что логика очень похожа на указатели



Как выглядят итераторы в коде?

```
vector<int> v{ 1, 5, 8, 9, 6, 7, 3, 4, 2, 0 };  
sort(v.begin(), v.end());
```

Вектор хранит в себе числа, которые мы хотим отсортировать по возрастанию, но самим придумывать сортировку нам не обязательно. Разработчики уже придумали функцию `sort()` для этих целей.

`v.begin()` - итератор указывающий на первый элемент

`v.end()` - итератор указывающий на последний элемент

Мы буквально говорим функции: Отсортируй элементы с первого по последний.

Самая любимая **ошибка** новичков, попытаться отсортировать контейнер вот так: `sort(v);`

Правильно будет вот так: `sort(v.begin(), v.end());`

Работа с вектором

begin() - возвращает итератор на первый элемент вектора (элемент с индексом 0)

end() - возвращает итератор на последний элемент вектора

push_back(n) - добавляет элемент n в конец вектора

insert(m, n) - вставляет элемент n на позицию, на которую указывает итератор m

size() - возвращает размер вектора (количество элементов в векторе)

back() - возвращает последний элемент вектора

front() - возвращает первый элемент вектора

pop_back() - удаляет последний элемент вектора

erase(m) - удаляет элемент, на который указывает итератор p

clear() - удаляет все элементы из вектора

empty() - возвращает 1, если вектор пуст, 0 - если в векторе содержатся элементы

Что такое стек?

Это контейнер, в котором взаимодействовать можно только с **последним элементом**.

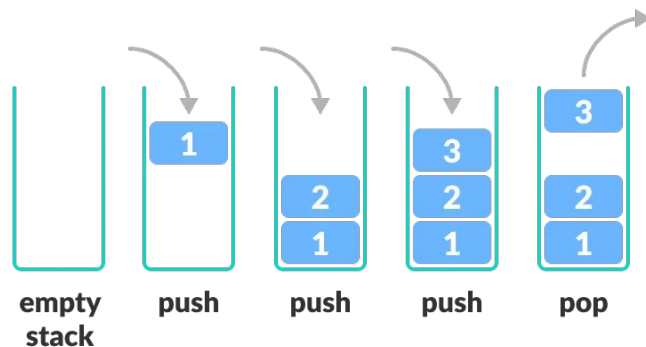
Пока мы не достанем последний добавленный элемент мы не сможем обратиться к следующему

Для стека работает правило LIFO (Last in First out)

Последний вошел, первый вышел

```
#include <stack>
```

```
stack<int> s1;
```



Что такое очередь?

Это контейнер, в котором взаимодействовать можно только с **первым элементом**.

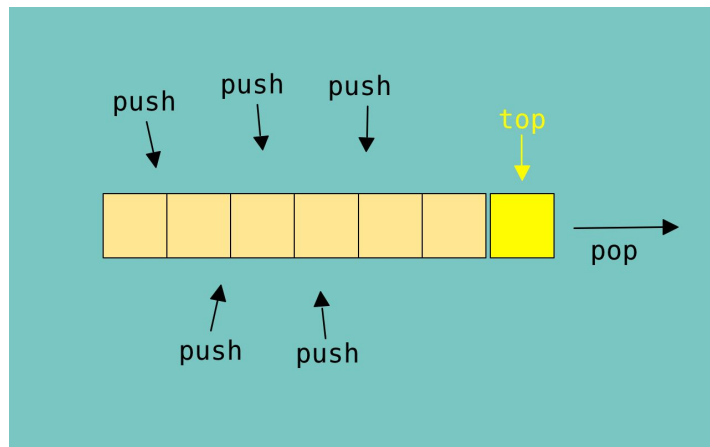
Но при этом мы можем обращаться к последнему элементу в очереди, в отличие от стека. Последний элемент удалить не получится

Для очереди работает правило FIFO (First in First out)

Первый вошел, первый вышел

```
#include <queue>
```

```
queue<int> q1;
```



Что такое дек?

Это контейнер, в котором взаимодействовать можно только с **первым элементом и последним элементом**.

Дек называют двухсторонней очередью

```
#include <deque>
```

```
deque<int> d1;
```

