

C++

Урок 19

A hand-drawn blue oval frame with a slightly irregular, sketchy border, centered on the page. The word "Повторение" is written inside this frame in a bold, black, sans-serif font.

Повторение

Что такое STL?



**Какие компоненты лежат в
основе STL?**

STL

STL (Standard Template Library – Стандартная Библиотека Шаблонов). Библиотека STL содержит пять основных видов компонентов:

- **Контейнер (container)**: управляет набором объектов в памяти.
- **Итератор (iterator)**: обеспечивает для алгоритма средство доступа к содержимому контейнера.
- **Алгоритм (algorithm)**: определяет вычислительную процедуру.
- **Функтор (function object)**: инкапсулирует функцию в объекте для использования другими компонентами.
- **Адаптер (adaptor)**: адаптирует компонент для обеспечения различного интерфейса.

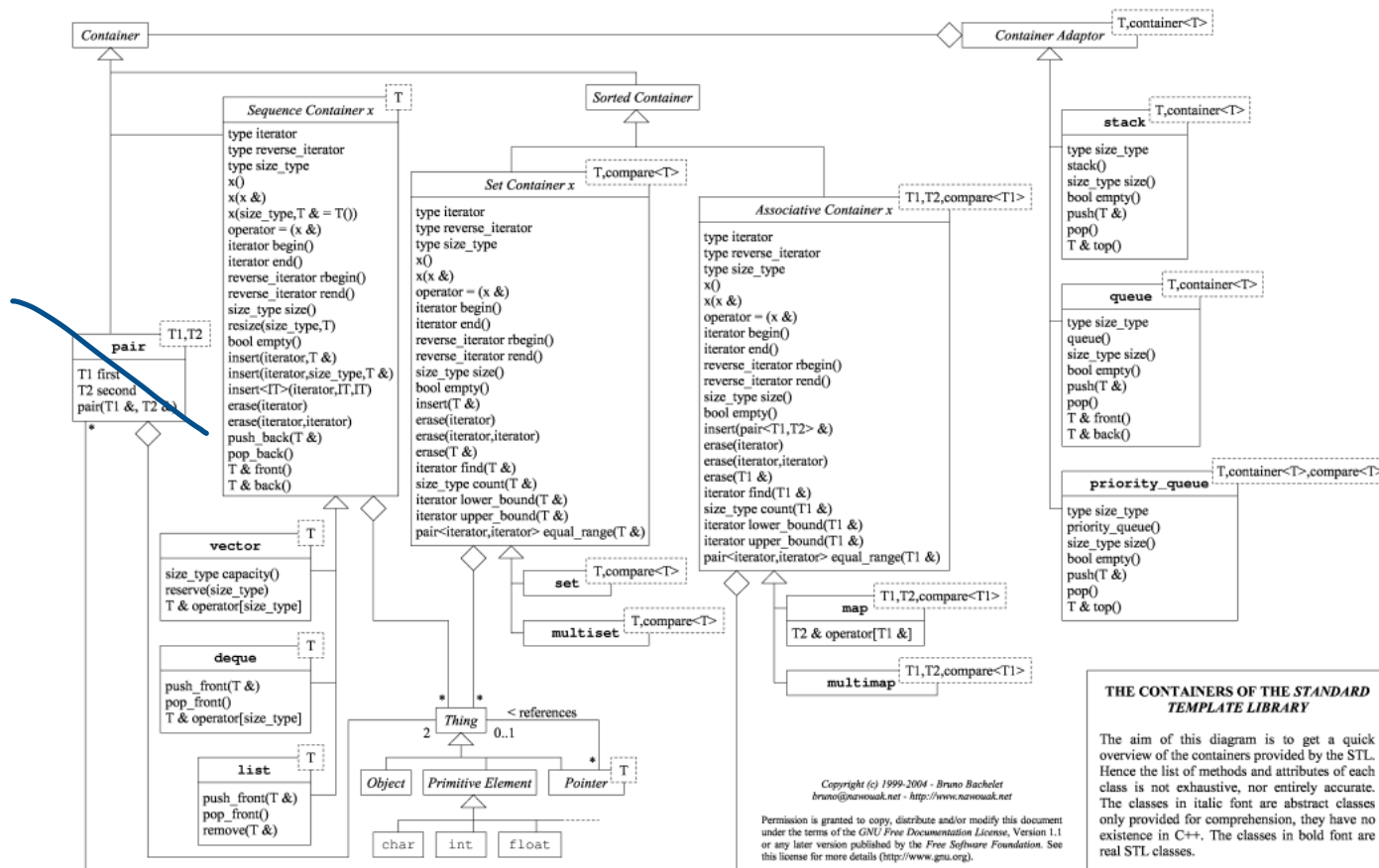
Из чего состоит пара?



В чем удобство?

STL

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



A hand-drawn blue oval frame with a double-line border, centered on the page. The word "Словари" is written inside this frame.

Словари



Что такое словарь?

Как вы им пользовались?

Словари

Словарь - сборник слов в алфавитном порядке, с пояснениями, толкованиями или с переводом на другой язык.



Словари

Словарь - сборник слов в алфавитном порядке, с пояснениями, толкованиями или с переводом на другой язык.

Словарь C++ (map) – это ассоциативный контейнер, который работает по принципу — [ключ — значение].

Словари

Словари часто называют также **ассоциативными массивами** или **отображениями**.

Словарь построен на основе пар значений, первое из которых представляет собой ключ для идентификации элемента, а второе — собственно элемент

Словари

Примеры ассоциативных контейнеров:

- `map` – словарь уникальных ключей,
- `multimap` – словарь ключей с дубликатами,
- `set` - множество,
- `multiset` – мультимножество,
- `bitset` – битовое множество (набор битов).

Easy Peasy



It comes from a 1970's [british TV](#) commercial for [Lemon Squeezy](#) detergent. They were with a little girl who points out dirty greasy dishes to an adult (mom or relative) and then this adult produces Lemon Squeezy and they clean the dishes quickly. At the end of the commercial the girl says "[Easy Peasy Lemon Squeezy](#)".

Today it is a silly way to state something was or will be very easy.

I will be [in an out](#), [easy peasy](#).

Can you open this [jar of pickles](#)? Sure thing easy peasy.

by [Creepy Gnome](#) August 2, 2009

**Приведите свой пример
ассоциативного массива**

Словари

```
#include <map>
```

- Подключаем библиотеку

Создание словаря

Словари

```
#include <map>
```

- Подключаем библиотеку

```
map<тип ключа, тип значения> название;
```

Создание словаря

Словари

```
#include <map>
```

- Подключаем библиотеку

```
map<тип ключа, тип значения> название;
```

```
int main()
{
    map<string, string> urbanDict;    //создали массив с ключем и значением типа string
    map<string, double> medicalMeasure;
    //создали массив с ключем типа string, а значением типа double
}
```

Создание словаря

Словари

```
map<string, double> medicalMeasure = {"Marti", 14.5},  
                                       {"Gloria", 10.15},  
                                       {"Melmon", 30.15}};
```

Инициализация словаря в момент
объявления

Словари

```
int main()
{
    map<string, double> medicalMeasure;

    medicalMeasure["Marti"] = 14.5;
    medicalMeasure["Gloria"] = 10.15;
    medicalMeasure["Melmon"] = 30.15;

    cout << medicalMeasure["Melmon"] << endl; // 30.15
}
```

Добавление элемента

Словари

```
int main()
{
    map<string, double> medicalMeasure;

    medicalMeasure["Marti"] = 14.5;
    medicalMeasure["Gloria"] = 10.15;
    medicalMeasure["Melmon"] = 30.15;
    medicalMeasure["Gloria"] = 25.15;

    cout << medicalMeasure["Gloria"] << endl;
}
```

Что будет выведено?

Словари

```
int main()
{
    map<string, double> medicalMeasure;

    medicalMeasure["Marti"] = 14.5;
    medicalMeasure["Gloria"] = 10.15;
    medicalMeasure["Melmon"] = 30.15;

    // 1 способ
    for (auto &it : medicalMeasure)
    {
        cout << it.first << " " << it.second << endl;
    }

    // 2 способ
    for (auto it = medicalMeasure.begin(); it != medicalMeasure.end(); it++)
    {
        cout << it->first << " " << it->second << endl;
    }

    // 3 способ
    for (auto it = medicalMeasure.begin(); it != medicalMeasure.end(); it++)
    {
        cout << (*it).first << " " << (*it).second << endl;
    }
}
```

Вывод эл-тов

Задача

Задача: Реализуйте словарь buildings, в котором содержится информация о самых высоких зданиях в мире.

Входные данные: ввод информации в словарь с **клавиатуры**.

Выходные данные: высота здания по его названию.

Задача

Задача: Реализуйте словарь buildings, в котором содержится информация о самых высоких зданиях в мире.

Входные данные: название строения.

Выходные данные: высота строения.

Словари

Ассоциативные массивы относятся к STL контейнерам.
Также в STL есть vector.

Словари

Ассоциативные массивы относятся к STL контейнерам.
Также в STL есть vector.

Какие методы мы использовали в vector?

Словари

Ассоциативные массивы относятся к STL контейнерам.
Также в STL есть vector.

Какие методы мы использовали в vector?

- find(ключ)
- count (ключ)
- erase (ключ)
- size()
- clear()
- empty()
- swap(контейнер)

Словари

Ассоциативные массивы относятся к STL контейнерам. Также в STL есть vector.

Какие методы мы использовали в vector?

- `find(ключ)` – возвращает итератор на найденный элемент. Если не нашел эл-т, то итератор на конец.
- `count (ключ)` - возвращает кол-во эл-тов по заданному ключу.
- `erase (ключ)`
- `size()`
- `clear()`
- `empty()`
- `swap(контейнер)`

[map документация](#)

Словари

Ассоциативные массивы относятся к STL контейнерам. Также в STL есть vector.

Какие методы мы использовали в vector?

- `find(ключ)` – возвращает итератор на найденный элемент. Если не нашел эл-т, то итератор на конец.
- `count (ключ)` - возвращает кол-во эл-тов по заданному ключу. В `map` вернет либо 0, либо 1.
- `erase (ключ)` – удаляет эл-т по заданному ключу.
- `size()` – кол-во эл-тов в словаре.
- `clear()` – удаление всех эл-тов из словаря.
- `empty()` – определяет пустой ли словарь.
- `swap(контейнер)` – меняет элементы между двумя сопоставлениями

Задача

Задача: Реализуйте словарь buildings, в котором содержится информация о самых высоких зданиях в мире.

Входные данные: ввод информации в словарь с **клавиатуры**.

Выходные данные: высота здания по его названию.

A hand-drawn blue oval frame with a double-line border, centered on the page. The word "Шаблоны" is written inside this frame in a bold, black, sans-serif font.

Шаблоны

Шаблоны

Шаблон — это конструкция, которая создает обычный тип или функцию во время компиляции на основе аргументов, которые пользователь предоставляет для параметров шаблона.

Шаблоны позволяют определять операции класса или функции и позволяют пользователю указать конкретные типы, с помощью которых должны работать эти операции.

Шаблоны

```
template <typename T>
T minimum(const T &lhs, const T &rhs)
{
    return lhs < rhs ? lhs : rhs;
}
```

Пример шаблонной ф-ии

Шаблоны

```
int main()
{
    int a, b;
    a = 10;
    b = 15;

    cout << minimum(a, b);
}
```

Параметры на вход

```
int minimum(const int &lhs, const int &rhs)
{
    return lhs < rhs ? lhs : rhs;
}
```

Вид ф-ии при входных
параметрах выше

Шаблоны

Ограничений на кол-во параметров типа **нет**.

```
template <typename T, typename U, typename V>
class Foo
{
};

template <class T, class U, class V>
class Foo
{
};
```

Различий в строчках нет