

Lecture 19

Iterators

Джигитовка `for`:

```
1 | for (auto it = cont.begin(); it != cont.end(); ++it; ++elts) { /* ... */ }
```

Range-based for

```
1 | for (init_statement; range_declaration : range_expression)
2 |     loop_statement;
3 |
4 | // <=>
5 |
6 | auto && __range = range expression;
7 |
8 | auto __begin = begin(__range); // обычно std::begin
9 | auto __end = end(__range); // обычно std::end
10 |
11 | for ( ; __begin != __end; ++__begin) {
12 |     range_declaration = *__begin;
13 |     loop_statement;
14 | }
```

Вызов `std::begin`, а не `range.begin` связан с тем, что у нас может быть перелан
встроенный массив

cppinsights.io - заменяет код на код стандартной библиотеки.

Итератор это не наследник - это обещание на функционал.

Свойства указателей

Создание по умолчанию, копирование, копирующее присваивание

Разыменование как `rvalue` и доступ к полям по разыменованию

Разыменование как `lvalue` и присваивание значения элементу под ним

Инкремент и постинкремент за $O(1)$

Сравнимость на равенство и неравенство за $O(1)$

Декремент и постдекремент за $O(1)$

Индексирование квадратными скобками, сложение с целыми, сравнение на
больше и меньше за $O(1)$

Многократный проход по одной и той же последовательности

Output iterator

Например, ostream

- Создание по умолчанию, копирование, копирующее присваивание
- Разыменование как rvalue и доступ к полям по разыменованию
- Разыменование как lvalue и присваивание значения элементу под ним
- Инкремент и постинкремент за $O(1)$
- Сравнимость на равенство и неравенство за $O(1)$
- Декремент и постдекремент за $O(1)$
- Индексирование квадратными скобками, сложение с целыми, сравнение на больше и меньше за $O(1)$
- Многократный проход по одной и той же последовательности

11

Input iterator

Например, istream

- Создание по умолчанию, копирование, копирующее присваивание
- Разыменование как rvalue и доступ к полям по разыменованию
- Разыменование как lvalue и присваивание значения элементу под ним
- Инкремент и постинкремент за $O(1)$
- Сравнимость на равенство и неравенство за $O(1)$
- Декремент и постдекремент за $O(1)$
- Индексирование квадратными скобками, сложение с целыми, сравнение на больше и меньше за $O(1)$
- Многократный проход по одной и той же последовательности

12

Forward iterator

Итератор по псевдослучайным числам

Создание по умолчанию, копирование, копирующее присваивание
Разыменование как `rvalue` и доступ к полям по разыменованию
Разыменование как `lvalue` и присваивание значения элементу под ним
Инкремент и постинкремент за $O(1)$
Сравнимость на равенство и неравенство за $O(1)$
Декремент и постдекремент за $O(1)$
Индексирование квадратными скобками, сложение с целыми, сравнение на больше и меньше за $O(1)$
Многочисленный проход по одной и той же последовательности

13

Bidirectional iterator

Создание по умолчанию, копирование, копирующее присваивание
Разыменование как `rvalue` и доступ к полям по разыменованию
Разыменование как `lvalue` и присваивание значения элементу под ним
Инкремент и постинкремент за $O(1)$
Сравнимость на равенство и неравенство за $O(1)$
Декремент и постдекремент за $O(1)$
Индексирование квадратными скобками, сложение с целыми, сравнение на больше и меньше за $O(1)$
Многочисленный проход по одной и той же последовательности

Random access iterator

Создание по умолчанию, копирование, копирующее присваивание
Разыменование как `rvalue` и доступ к полям по разыменованию
Разыменование как `lvalue` и присваивание значения элементу под ним
Инкремент и постинкремент за $O(1)$
Сравнимость на равенство и неравенство за $O(1)$
Декремент и постдекремент за $O(1)$
Индексирование квадратными скобками, сложение с целыми, сравнение на больше и меньше за $O(1)$
Многочисленный проход по одной и той же последовательности

15

+ Сложение с целыми за $O(1)$

Итерационные функции

```
1 std::distance(Iter fst, int n); // snd - fst, либо цикл
2 std::advance(Iter sat, int n); // fst + n, либо цикл
```

У таких функций, в отличие от интерфейса итераторов, неопределенная асимптотическая сложность.

`prev = std::exchange(cur, cur + prev)` - записать в `cur` новое значение, а старое `cur` выдать в `prev`.