



Cvičení C++

3.12.2018

faltin@ksi.mff.cuni.cz



Lambdy

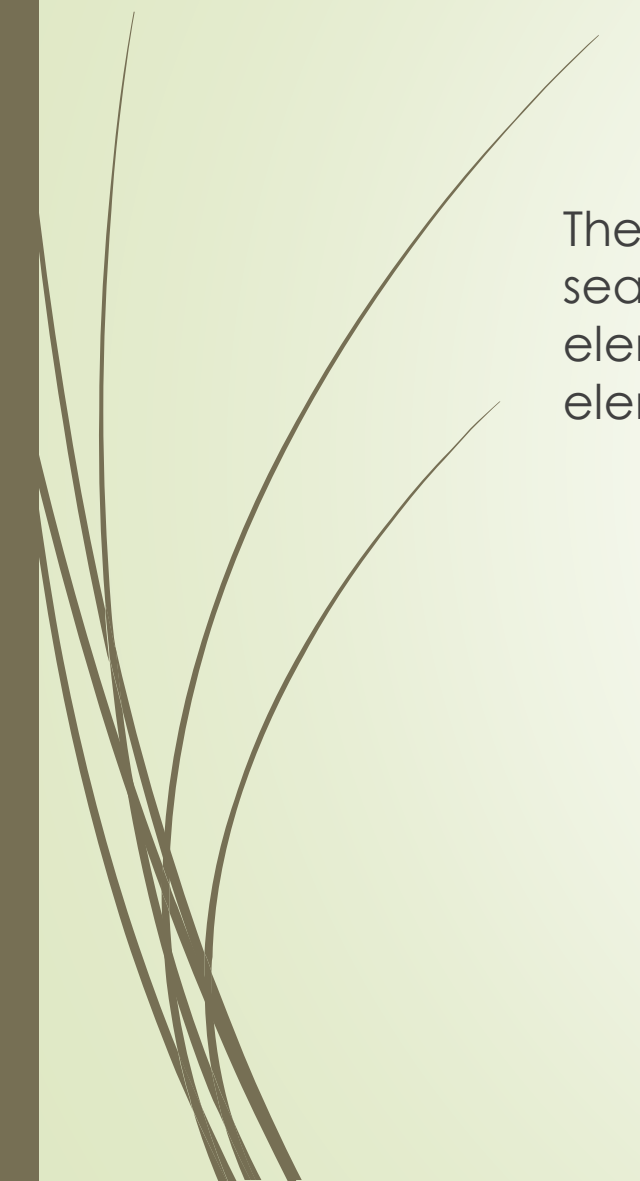
```
auto hello_lambda = []() { cout << "Hello!"; }  
hello_lambda();
```

```
vector<int> vi; int value_to_find;  
std::cin >> value_to_find;  
std::find_if(std::begin(vi), std::end(vi), [value_to_find](int value) {  
    return value == value_to_find;  
});
```

```
template<class InputIt, class UnaryPredicate>  
constexpr InputIt find_if(InputIt first, InputIt last, UnaryPredicate p) {  
    for (; first != last; ++first) {  
        if (p(*first)) {  
            return first;  
        }  
    }  
    return last;  
}
```



<algorithm>



The algorithms library defines functions for a variety of purposes (e.g. searching, sorting, counting, manipulating) that operate on ranges of elements. Note that a range is defined as `[first, last)` where `last` refers to the element past the last element to inspect or modify.

Úkoly

1. Vytvořte pole náhodných čísel
 - `<random>`, `std::uniform_int_distribution`,
2. Seříděte pole
 1. Pomocí `std::sort`
 2. Vlastní mergesort – `std::merge` + `std::sort`
 3. Vlastní heapsort – `std::make_heap`, `std::push_heap`, `std::pop_heap`
3. Sečtěte prvky v poli
 1. Pomocí `std::for_each` + `lambda`
 2. Pomocí `std::for_each` + `functor`
4. Převeďte všechny prvky v poli na pole `bool[]` -> 1 pokud je hodnota liché
 1. `std::transform`
5. Vytvořte pole pouze prvků s lichou hodnotou (k prvkům se sudou přičtěte 1)
 1. `std::transform` + `std::back_inserter`
6. Kalkulačka/expression engine
 1. Dostanete strom složený z výrazů
 2. Převod textu -> na strom
 - Bude umět vyhodnotit například „ $(1 + 2) * (3 - 1) / 2 + 5$ “
7. Boosted Kalkulačka/Expression engine
 - Co nejrychlejší implementace
 - Minimalizovat použití virtual call, uloženo kompaktně v paměti, ...