



# Cvičení C++

26.11.2018

[faltin@ksi.mff.cuni.cz](mailto:faltin@ksi.mff.cuni.cz)



# Kontainery



- `array<T, std::size_t>`, `vector<T>`, `deque<T>`, `forward_list<T>`, `list<T>`
- `set<T>`, `map<T>`, `multiset<T>`, `multimap<T>`
- `unordered_set<T>`, `unordered_map<T>`, `unordered_multiset<T>`, `unordered_multimap<T>`
- `stack<T>`, `queue<T>`, `priority_queue<T>`



# Funktory

```
struct add_x {  
    add_x(int x) : x(x) {}  
    int operator()(int y) const { return x + y; }
```

```
private:  
    int x;  
};
```

```
add_x add42(42); // create an instance of the functor class  
int i = add42(8); // and "call" it  
assert(i == 50); // and it added 42 to its argument
```

```
std::vector<int> in; // assume this contains a bunch of values)  
std::vector<int> out(in.size());
```

```
// Pass a functor to std::transform, which calls the functor on every element  
// in the input sequence, and stores the result to the output sequence  
std::transform(in.begin(), in.end(), out.begin(), add_x(1));
```



# Generování


- `<algorithm>`
  - `std::generate`, `std::generate_n`
  - `std::fill`, `std::fill_n`
- `<numeric>`
  - `std::iota`



# Úkoly



- Agregace v databázi v paměti – GROUP BY (DISTINCT)
  1. Nagenertejte si do paměti několik sloupců pro databázi
    1. <algorithm>
  2. Určete za běhu, podle kterého sloupce chcete agregovat výsledky
    1. map/unordered\_map
  3. Určete za běhu, podle kterého sloupce chcete agregovat **unikátní** výsledky
    1. set/unordered\_set



# Domácí úkol – „databáze“

- V ReCodexu - <https://recodex.mff.cuni.cz>
- Variace na databázi ze cvičení
- 14 dní