

# Итератори

Калин Георгиев

12 януари 2020 г.

# Приложение

Как достъпваме последователно (итерираме) елементите на контейнер?

- чрез индекс

```
for (size_t i = 0; i < v.size(); ++i){...v[i]...}
```

Проблеми на простото индексване:

- производителност
- не можем да се абстрахираме от контейнера  
`void map (T (*f) (const T&), [...]) WHAT)`
- separation of concerns
- понякога поредният номер е заблуждаващ

## “Речник” на линейното обхождане

```
for (size_t i = 0; i < v.size(); ++i){...v[i]...}
```

- Започни!
- Дай текущия елемент!
- Промени текущия елемент!
- Дай следващия (предишния)!
- Има ли още елементи?

```
>for (size_t j = 0; j < v.size(); ++j){if (i==j)...}
```

- Тези позиции еднакви ли са?

# Итератори

Създаване на итератор в контейнерния клас:

- `begin`
- `end`

Операции с итератора

- `*`
- `++`, `--`
- `==` (*Итератор към "края"*)
- копиране
- `for`

Интервали и адаптори  
`std::ranges::range` & `std::ranges::views`  
C++20!

# Интервали и адаптори

```
int main()
{
    std::vector<int> ints{0,1,2,3,4,5};
    std::function<bool(int)>
        even = [](int i){ return 0 == i % 2; };
    std::function<int(int)>
        square = [](int i) { return i * i; };

    for (int i : ints |
          std::views::filter(even) |
          std::views::transform(square)) {
        std::cout << i << '␣';
    }
}
```

Благодаря за вниманието!