

# C++ Ranges

Petar Trifunovic, 21.9.2022

# Pregled

Sta su range-ovi

Kako su nastali

Od cega se sastoje

Zasto bi ih koristili

# Sta su range-ovi

- Par iteratora koji predstavlja pocetak i kraj?
- Iterator i broj elemenata?
- Iterator i uslov prekida iteracije?

# Sta su range-ovi - formalno

```
template< class T >
concept range = requires( T& t ) {
    ranges::begin(t); // equality-preserving for forward iterators
    ranges::end    (t);
};
```

- *[ranges::begin(E), ranges::end(E)) denotes a range, and*
- *both ranges::begin(E) and ranges::end(E) are amortized constant time and do not alter the value of E in a manner observable to equality-preserving expressions*
- *if the type of ranges::begin(E) models forward\_iterator, ranges::begin(E) is equality-preserving (in other words, forward iterators support multi-pass algorithms)*

# Sta su range-ovi - neformalno

- `range::begin(E)` -> metoda za pristup prvom iterator-u range-a
- `range::end(E)` -> metoda za pristup sentinel-u range-a
- Provera da li je iterator dosao do kraja range-a

# Kako su nastali range-ovi

- Ranges v0 => STL
  - Prva je uvela pojam range-a
- Ranges v1
  - Uvedena u boost v1.32
  - Samo je imala koncept range-a i range algoritama
- Range v2
  - Uvedena u boost v1.43
  - Nastala kao spoj boost.Range.v1 i boost.Iterator-a
  - Uvela je koncept ranga
- Range v3
  - Uvedena od strane Erica Niblera sa idejom da modernizuje Ranges v2 na c++11
  - Dodala support za koncepte
  - Dodala support za projekcije

# Od čega se sastoje range-ovi

- koncepti
- algoritmi
- view-ovi

# Od cega se sastoje range-ovi - koncepti

## SUSTINA

- **range**
- borrowed\_range
- viewable\_range
- **view**

## HIJERARHIJA

- output\_range
- input\_range
- forward\_range
- bidirectional\_range
- random\_access\_range
- contiguous\_range

## REPOVI

- sized\_range
- constant\_range
- common\_range



# Od cega se sastoje range-ovi - algoritmi

```
sort(rng, std::less{}, &Person::age)
```

```
lower_bound(rng, 27, std::less{}, &Person::age)
```

- Svi vasi omiljeni algoritmi su sada dobili i range verzije
  - Osim numeric algoritama ofc :)
  - <https://en.cppreference.com/w/cpp/algorithm/ranges>
- Range algoritmi su konstrejnovani konceptima
  - Napokon mozemo da imamo normalne compiler-ske greske
- Range algoritmi su dobili projekcije
  - Projekcije omogucavaju simetricniji api

# Od cega se sastoje range-ovi - view-ovi

```
rng | filter(greater_than{7}) | transform(square)
```

- Lazy
- Composable
- Deklarativno

Zasto bih koristio range-ove?

DEMO