

# Razvoj softvera

Vežbe 10

# Range (Opseg)

- biblioteka: ranges-v3
- Abstrakcija koja predstavlja kolekciju elemenata.
- Radi jednostavnosti, Range možemo posmatrajmi strukturom koja ima iterator na pocetak kolekcije, iterator na element nakon poslednjeg u kolekciji i omogućava pristup svakom elementu
- Omogućava vraćanje raspona kao rezultat funkcije i njegovo prosleđivanje drugoj funkciji bez pravljenja lokalnih promenljivih za međurezultat

# Range

- Biblioteka obezbedjuje cev (eng. pipe) sintaksu za ugnježdene pozive funkcija
  - operator |

# Primer

Iz kolekcije koja sadrži objekte klase *Car* izdvojiti identifikatore automobila crvene boje

```
std::vector<std::string> ids = cars |  
    filter(is_red) | transform(id);
```

- **filter** vraća elemente koji zadovoljavaju predikat
- **transform** vraća izmenjen element

# Primer

`cars | filter(is_red)`

- je novi pogled (view) na postojeće podatke, tj. Range struktura

- Korisnik može da koristi pogled kao da je kolekcija samo sa automobilima crvene boje

`| transform(id)`

- novi pogled koji ne prikazuje originalne vrednosti elemenata, vec transformisane

- Pogledi se lenjo izvršavaju - samo definišu pogled, ne prolaze kroz kolekciju. Na kraju se pravi kolekcija prolaskom kroz poglede.
- Za gornji primer samo jedan prolaz kroz kolekciju

# Transformacije u Range

- Transformacije koje mogu da se primene se dele na:
- poglede (views) :
  - imenski prostor `range::view`
  - ne menjanu originalnu kolekciju, samo prave lenji pogled nad originalnom kolekcijom
  - dobro su rešenje kada ne očekujemo da će svi elementi kolekcije biti procesirani, i kada ih je dovoljno procesirati jedanput
- akcije (actions)
  - imenski prostor `range::actions`
  - izvršava transformacije nad originalnoj kolekciji
  - za računanje elemenata rezultujuće kolekcije kada znamo da će se koristiti često