

Neke korisne funkcije biblioteke range-v3

Lenje transformacije:

- `xs | view::chunk(n)`
deli `xs` na delove od po `n` elemenata
- `xs | view::drop(n)`
preskace prvih `n` elemenata iz `xs`
- `xs | view::drop_while(pred)`
preskace elemente sa pocetka koji zadovoljavaju predikat `pred`
- `xs | view::take(n)`
uzima prvih `n` elemenata iz `xs`
- `xs | view::take_while(pred)`
uzima elemente sa pocetka koji zadovoljavaju predikat `pred`
- `xs | view::group_by(eq_cmp)`
grupise uzastopne elemente `x[i]` i `x[i+1]` koji zadovoljavaju `eq_cmp(x[i], x[i+1])`
- `xs | view::join`
pegla ugnjezdene kolekcije
- `xs | view::filter(pred)`
ostavlja elemente koji zadovoljavaju predikat `pred`
- `xs | view::remove_if(pred)`
uklanja elemente koji zadovoljavaju predikat `pred`
- `xs | view::transform(f)`
primenjuje `f` na svako `x` i vraca range rezultata
- `xs | view::reverse`
obrce redosled elemenata
- `view::zip(xs, ys)`
pravi range parova `(x, y)`
- `view::zip_with(f, xs, ys)`
kao `zip(xs, ys) | transform(f)`, samo sto je `f` binarna f-ja, ne f-ja koja prihvata `std::pair`

- `accumulate(xs, init, f)`
kao `std::accumulate`, samo umesto para iteratora se prosledjuje range ili kolekcija

Neki osnovni range objekti:

- `view::ints(n)`
prirodni brojevi pocevsi od `n`
- `view::ints(n, m)`
prirodni brojevi od `n`, do `m`
- `view::repeat(x)`
ponavlja istu vrednost
- `view::repeat_n(x, n)`
ponavlja istu vrednost `n` puta

Ostale transformacije i akcije:

- `xs | to_vector`
pravi vektor od range objekta
- `xs | action::sort`
sortira kolekciju