JAVA 1

Streams

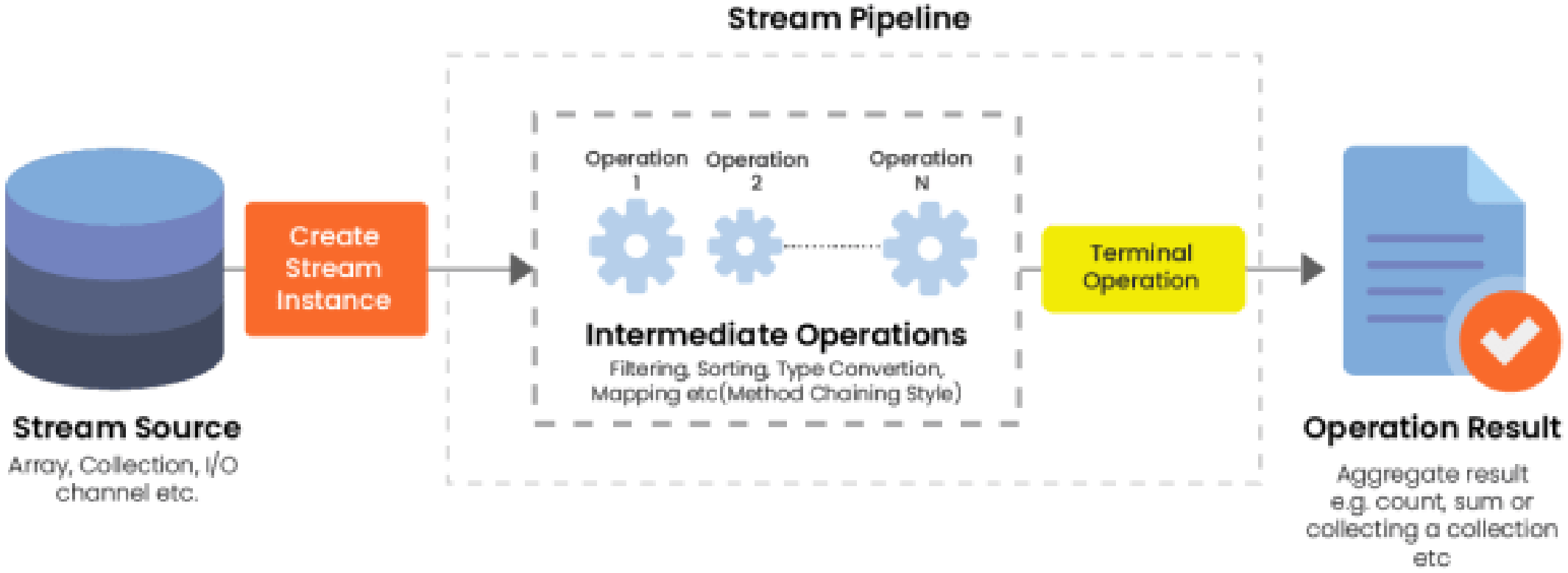
Teme

- Streams
- Stream pipeline
- Source operations
- Intermediary operations
- Terminal operations
- Optional
- Parallel Streams

Streams

- obrada podataka u kolekcijama na višoj razini apstrakcije
- slični iteratorima
- sučelje *Stream* – mnoštvo metoda
- brojni načini instanciranja
- kolekcije imaju *stream()* metodu koja ih pretvara u tokove
- izvorna kolekcija ostaje nepromijenjena!

Stream pipeline



Izvor:http://blogs.innovationm.com/concept-of-stream-api-java1-8/

Stream pipeline

Sastoji se od tri dijela:

- izvor – *source* – proizvode tok
- međuperacije - *intermediate operations*
  - operacije nad tokovima
  - *lazy* – izvršavaju se prilikom poziva završne operacije
  - vraćaju *stream* – *fluent API*
- završne operacije - *terminal operations*
  - proizvode rezultate
  - *eager* – izvršavaju se pri pozivu i okidaju izvršavanje međuoperacija

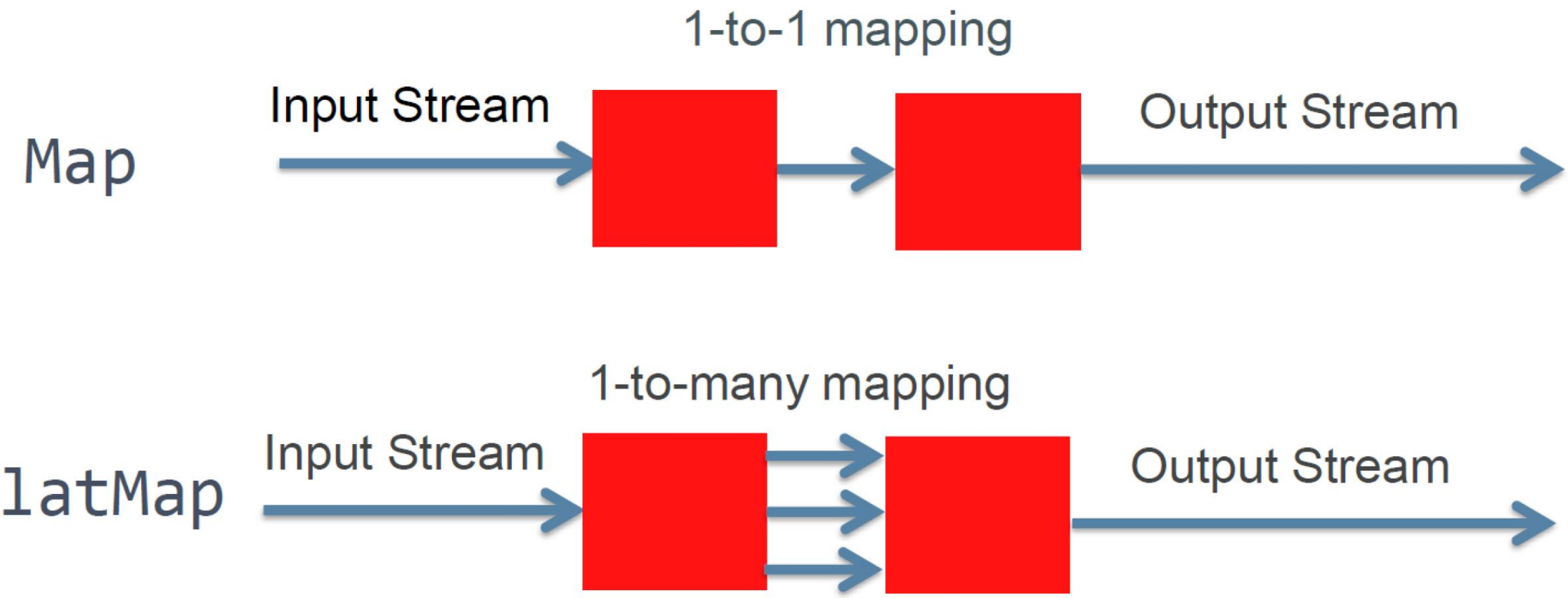
Source operations

- *{collection}.stream()*
- *Arrays.stream(array[])*
- *Stream<T>.builder()*
- *Stream.of()*
- *Stream.generate()*
- *Stream.iterate()*
- *IntStream, LongStream…*

Intermediary operations

- *filter* – filtriranje prema predikatu
- *distinct* – eliminiranje duplikata
- *limit* – ograničavanje količine podataka
- *sorted* – sortiranje
- *skip* – preskakanje elemenata
- *map* – mapiranje izvorne na željenu vrijednost
- *flatMap* – mapiranje izvorne na niz željenih vrijednosti
- *peek* – pregledavanje elemenata u toku (*debug*)

Map vs. FlatMap

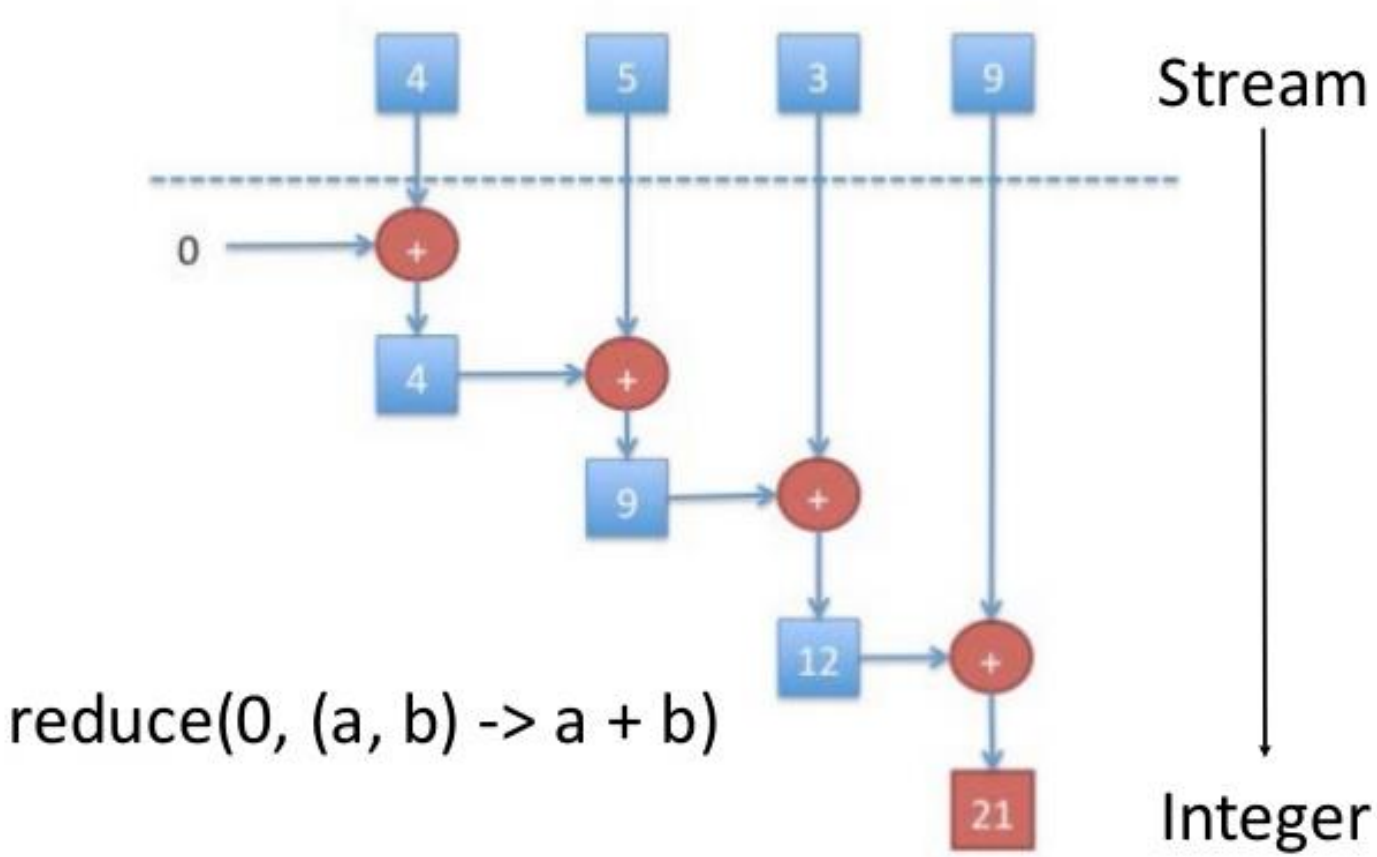


Terminal operations

- *foreach*– procesiranje svakog elementa
- *reduce* – jedinstveni rezultat iz toka
- *collect* – kreiranje nove kolekcije iz toka

- *max, min, average, count*
- *findFirst, findAny*
- *anyMatch, allMatch, noneMatch*

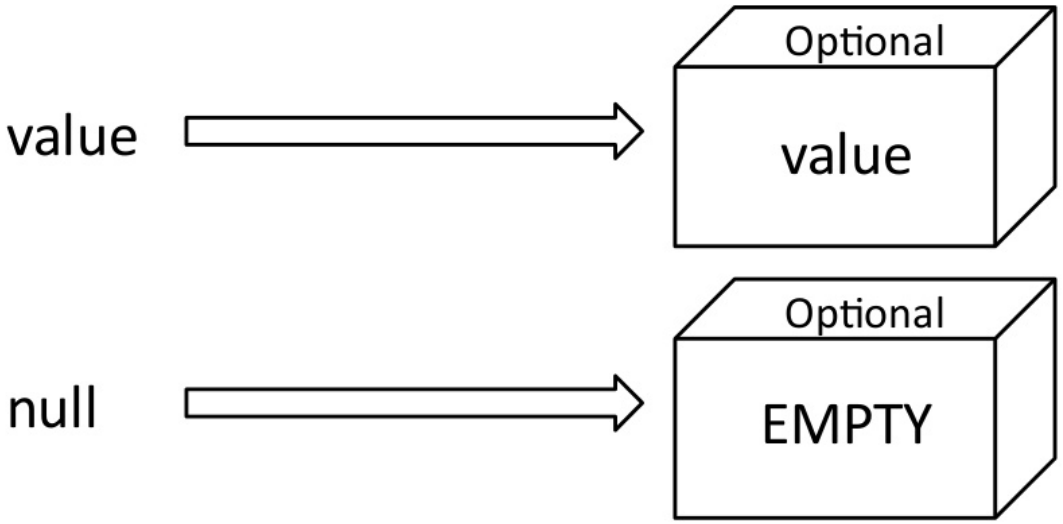
Reduce



Izvor:https://www.slideshare.net/mariofusco/java-8-workshop

Optional

- vrijednost koja može, ali ne mora postojati
- izbjegavanje *NullPointerException*
- metoda *isPresent()* za provjeru
- metoda *get()* za dohvat



Izvor:https://javarevisited.blogspot.com/2017/04/10-examples-of-optional-in-java-8.html

Parallel Streams

- maksimalno iskorištavanja snage višejezgrenih procesora
- može uvelike ubrzati izvršavanje operacija
  - *stateless* – stanje jednog elementa ne utječe na stanje drugog
  - *non-interfering* – izvor podataka se ne mijenja
  - *associative* – rezultat ne ovisi o rasporedu elemenata (avg, max, min, count)
- nije uvijek prikladno

NQ model

N * Q > 10000

- N – number of data
- Q – broj operacija nad podacima

Demo

- Project

Izvor:http://www.jnhsolutions.com/contact-us/request-a-demo/

Hvala na pažnji!