Applying Key Operators in the Flowable Class: Case Study ex3

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science
Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Part of the Lesson

- Case study ex3 shows how to multiply & add big fractions asynchronously & concurrently using RxJava Flowable operators (e.g., fromArray() & parallel()) & ParallelFlowable operators, (e.g., runOn(), flatMap(), reduce(), & sequential()), & the Schedulers. computation() thread pool

```java
return Flowable
  .fromArray(bigFractionArray)

  .parallel()

  .runOn
    (Schedulers.computation())

  .map(bf -> bf.multiply
    (sBigReducedFraction))

  .sequential()

  .reduce(BigFraction::add)
```

Applying Key Operators in the Flowable Class to ex3
return Flowable

    // Emit a stream of reduced big fractions.
    .fromArray(bigFractionArray)

    // Convert the Flowable to a ParallelFlowable.
    .parallel()

    // Run subsequent processing in the computation pool.
    .runOn(Schedulers.computation())

    // Use RxJava's flatMap() concurrency idiom to multiply
    // these BigFractions asynchronously in a thread pool.
    .map(bf -> bf
        .multiply(sBigReducedFraction))

    // Convert the ParallelFlowable back into a Flowable.
    .sequential()

    // Reduce the results into one Maybe<BigFraction>.
    .reduce(BigFraction::add)

    // Display the results if all goes well.
End of Applying Key Operators in the Flowable Class: Case Study ex3