

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

```
return Flowable
    .fromArray(bigFractionArray)

    .parallel()

    .runOn
        (Schedulers.computation())

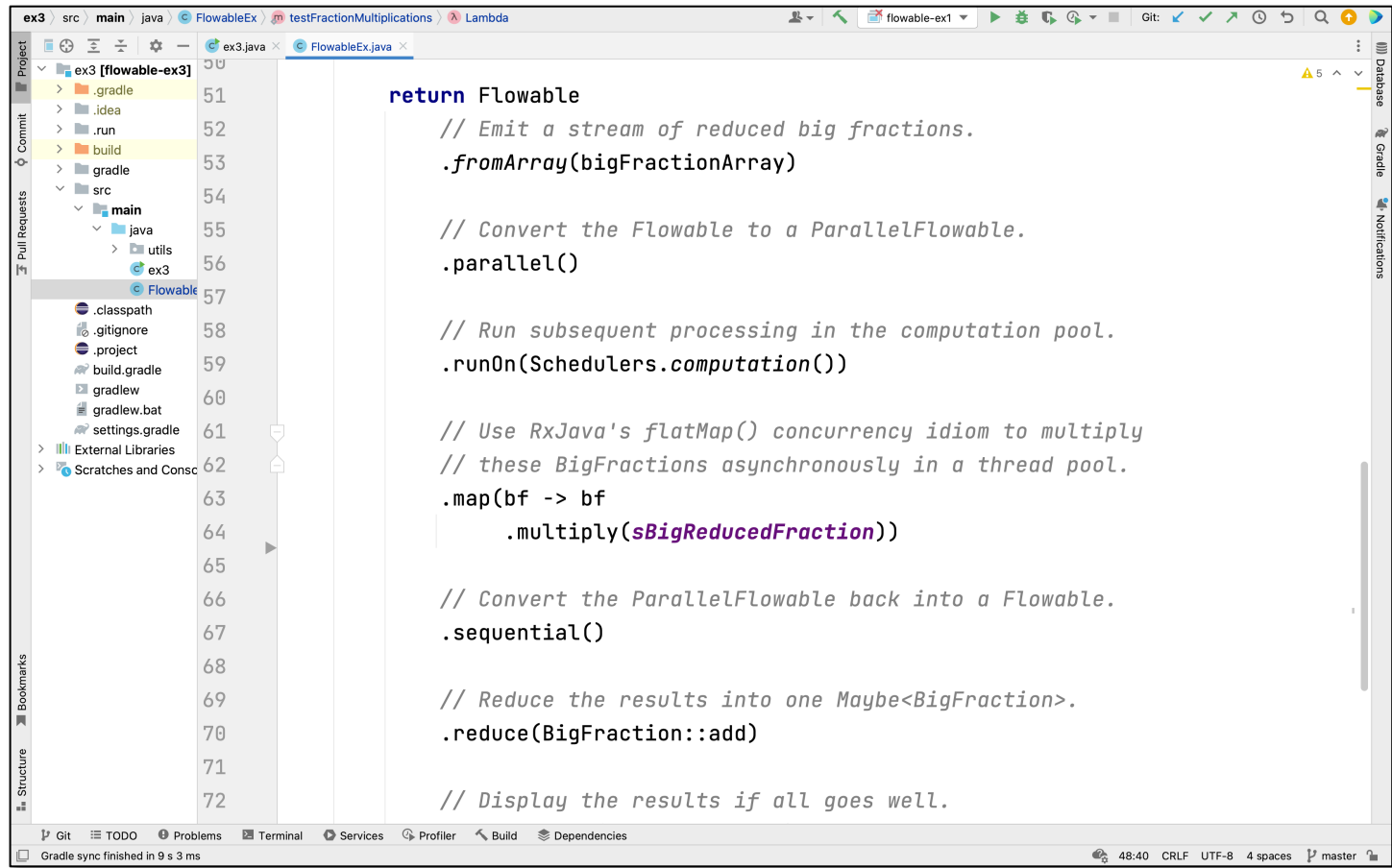
    .map(bf -> bf.multiply
        (sBigReducedFraction))

    .sequential()

    .reduce(BigFraction::add)
```

Applying Key Operators in the Flowable Class to ex3

Applying Key Operators in the Flowable Class to ex3



```
51 return Flowable
52     // Emit a stream of reduced big fractions.
53     .fromArray(bigFractionArray)
54
55     // Convert the Flowable to a ParallelFlowable.
56     .parallel()
57
58     // Run subsequent processing in the computation pool.
59     .runOn(Schedulers.computation())
60
61     // Use RxJava's flatMap() concurrency idiom to multiply
62     // these BigFractions asynchronously in a thread pool.
63     .flatMap(bf -> bf
64         .multiply(sBigReducedFraction))
65
66     // Convert the ParallelFlowable back into a Flowable.
67     .sequential()
68
69     // Reduce the results into one Maybe<BigFraction>.
70     .reduce(BigFraction::add)
71
72     // Display the results if all goes well.
```

See github.com/douglasraigschmidt/LiveLessons/tree/master/Reactive/Flowable/ex3

End of Applying Key Operators in the Flowable Class: Case Study ex3