Applying Key Operators in RxJava: Case Study ex4 (Part 2)

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

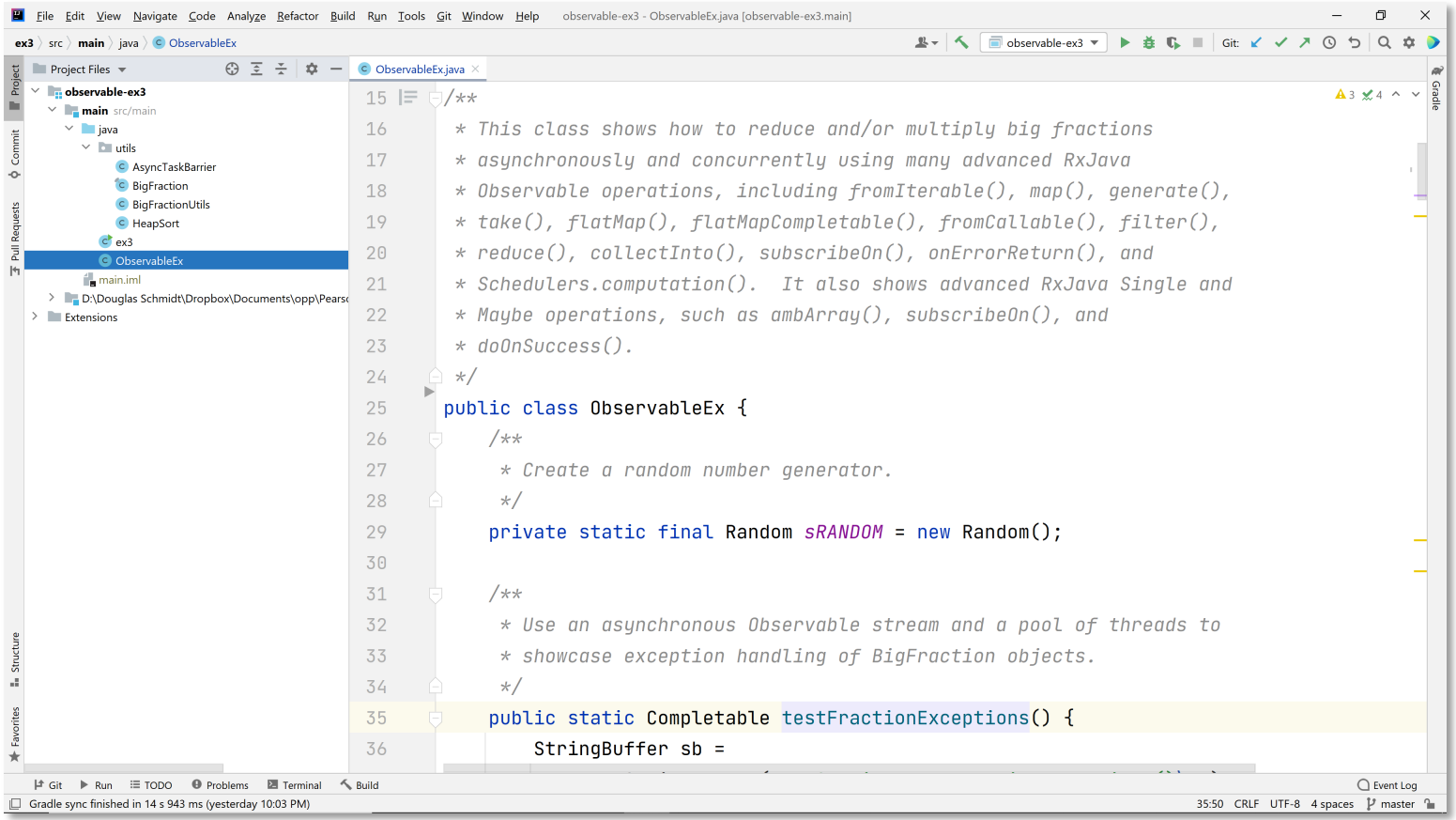
- Part 2 of case study ex4 explores how to combine Java Streams & RxJava Single operators flatMap(), flatMapCompletable(), zipArray(), ignoreElement(), ambArray(), & doOnSuccess() to create, reduce, multiply, & display BigFraction objects asynchronously

```
return Stream
   .generate(() ->
      makeBigFraction(new Random(),
                      false))
   .limit(sMAX_FRACTIONS)
   .map(unreducedBigFraction ->
        reduceAndMultiplyFraction
          (unreducedBigFraction,
           Schedulers.fromExecutor
             (ForkJoinPool
              .commonPool())))
   .collect(toSingle())
   .flatMapCompletable(list ->
      sortAndPrintList(list, sb));
```

See github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/Observable/ex4

Applying Key Operators in the Observable Class to ex4

Applying Key Operators in the Observable Class to ex4



See github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/Observable/ex4

End of Applying Key Methods in the Observable Class: Case Study ex4 (Part 2)