Applying Key Operators in the Observable Class: Case Study ex2 (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

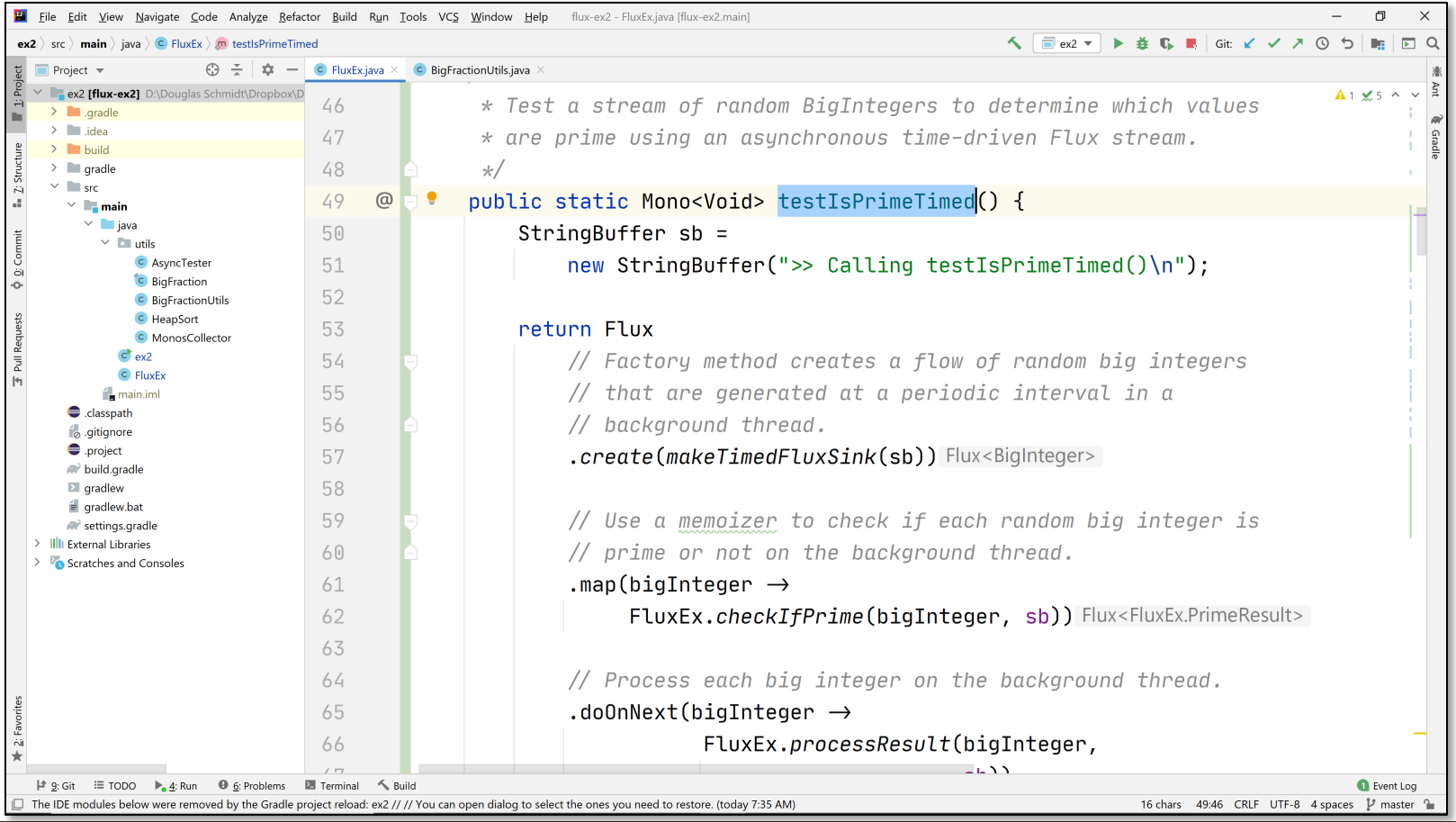
- Part 2 of case study ex2 explores how to use the RxJava Observable operators create(), rangeLong(), map(), take(), filter(), subscribe(), doFinally(), subscribeOn(), ignoreElements(), observeOn(), & doOnNext() to create large random BigInteger objects in one thread & asynchronously check if they are prime in another thread

```
Observable
   .create
     (ObservableEx::emitAsync)
   ...
   .observeOn(Schedulers
                .newThread())
   .map(bigInteger ->
        ObservableEx.checkIfPrime
          (bigInteger, sb))
   ...
   .doFinally(() ->
          BigFractionUtils.
          display(sb.toString()))
   .ignoreElements();
```

See github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/Observable/ex2

Applying Key Operators in the Observable Class to ex2

Applying Key Operators in the Flux Class to ex2



See github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/Observable/ex2

End of Applying Key Methods in the Observable Class: Case Study ex2 (Part 2)