

Applying Key Operators in the Observable

Class: Case Study ex4 (Part 1)

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 1 of case study ex4 applies Observable operators `fromArray()`, `toFlowable()`, `map()`, `subscribeOn()`, `flatMap()`, `subscribe()`, & a thread pool to create, multiply, & display `BigFraction` objects asynchronously

Single

```
.fromCallable(() ->
    .makeBigFraction(sRANDOM,
                    true))

.flatMapObservable(bf1 ->
    Observable
    .fromArray(bigFractionArray)
    .flatMap(bf2 -> Observable
        .fromCallable(() -> bf2)
        .subscribeOn(scheduler)
        .map(____ -> bf2
            .multiply(bf1))))

.toFlowable(LATEST)

.subscribe(blockingSubscriber);
```

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

Learning Objectives in this Part of the Lesson

- Part 1 of case study ex4 explores Observable operators `fromArray()`, `toFlowable()`, `map()`, `subscribeOn()`, `flatMap()`, the Flowable operator `subscribe()`, & the `computation()` thread pool to asynchronously create, multiply, & display `BigFraction` objects
- It also shows Single operators `subscribeOn()`, `fromCallable()`, `flatMapObservable()`, & `flatMapCompletable()`

Single

```
.fromCallable(() ->
    .makeBigFraction(sRANDOM,
                      true))

.flatMapObservable(bf1 ->
    Observable
    .fromArray(bigFractionArray)
    .flatMap(bf2 -> Observable
        .fromCallable(() -> bf2)
        .subscribeOn(scheduler)
        .map(____ -> bf2
            .multiply(bf1))))

.toFlowable(LATEST)

.subscribe(blockingSubscriber);
```

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

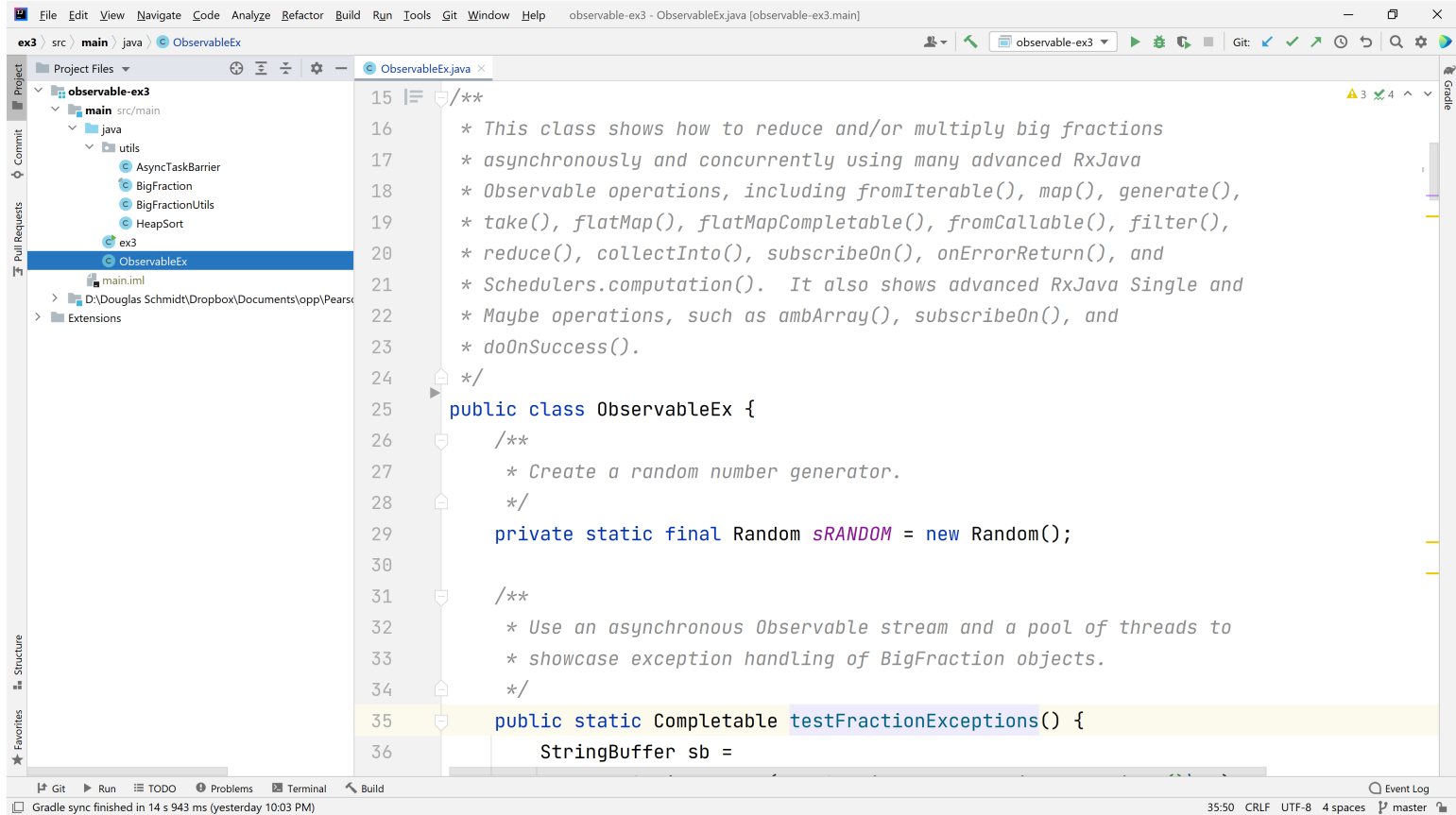
Learning Objectives in this Part of the Lesson

- Part 1 of case study ex4 explores Observable operators fromArray(), toFlowable(), map(), subscribeOn(), flatMap(), the Flowable operator subscribe(), & the computation() thread pool to asynchronously create, multiply, & display BigFraction objects
 - It also shows Single operators subscribeOn(), fromCallable(), flatMapObservable(), & flatMapCompletable()
 - In addition, it shows how to create & use a generic blocking Subscriber

```
class BlockingSubscriber<T>
    implements Subscriber<T> {
    ...
    final CountDownLatch mLatch;
    ...
    @Override
    public void onComplete() {
        ...
        mLatch.countDown();
    }
    ...
}
```

Applying Key Operators in the Observable Class to ex4

Applying Key Operators in the Observable Class to ex4



```
15  /**
16   * This class shows how to reduce and/or multiply big fractions
17   * asynchronously and concurrently using many advanced RxJava
18   * Observable operations, including fromIterable(), map(), generate(),
19   * take(), flatMap(), flatMapCompletable(), fromCallable(), filter(),
20   * reduce(), collectInto(), subscribeOn(), onErrorReturn(), and
21   * Schedulers.computation(). It also shows advanced RxJava Single and
22   * Maybe operations, such as ambArray(), subscribeOn(), and
23   * doOnSuccess().
24   */
25  public class ObservableEx {
26      /**
27       * Create a random number generator.
28       */
29      private static final Random sRANDOM = new Random();
30
31      /**
32       * Use an asynchronous Observable stream and a pool of threads to
33       * showcase exception handling of BigFraction objects.
34       */
35      public static Completable testFractionExceptions() {
36          StringBuffer sb =
```

Gradle sync finished in 14 s 943 ms (yesterday 10:03 PM)

35:50 CRLF UTF-8 4 spaces master

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

End of Applying Key Methods in the Observable Class: Case Study ex4 (Part 1)