Applying Key Operators in the Flux 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 shows how to use Flux operators `create()`, `map()`, `filter()`, `take()`, `subscribe()`, `subscribeOn()`, `publishOn()`, `then()`, `subscribeOn()`, `subscribeOn()`, `publishOn()`, `then()`, `range()`, `doOnNext()`, & `doFinally()` to create large random `BigInteger` objects & asynchronously check if they are prime via publisher & subscriber threads created using `Schedulers.newParallel()`

```java
Scheduler publisher = Schedulers
    .newParallel("publisher", 1));
Flux
    .range(1, sMAX_ITERATIONS)
    ...
    .subscribeOn(publisher)
    .map(__ -> BigInteger
        .valueOf(lowerBound + rand
            .nextInt(sMAX_ITERATIONS)))
    ...
    .doFinally(() -> publisher
        .dispose())
    .subscribe(sink::next,
        err -> ...,
        sink::complete);
```
Applying Key Operators in the Flux Class to ex2
Applying Key Operators in the Flux Class to ex2

```java
public static Mono<Void> testIsPrimeAsync() {
    StringBuffer sb =
        new StringBuffer(">> Calling testIsPrimeAsync()\n");

    return Flux
        // Factory method creates a flow of random big integers
        // that are generated in a background thread.
        .create(makeAsyncFluxSink(sb))
        // Arrange to perform the prime-checking computations in the
        // "subscriber" thread.
        .publishOn(Schedulers.newParallel(name: "subscriber", parallelism: 1))
        // Use a memoizer to check if each random big integer is
        // prime or not in the "subscriber" thread.
        .map(bigInteger -> FluxEx.checkIfPrime(bigInteger, sb))};
```

See [github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/flux/ex2](https://github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/flux/ex2)
End of Applying Key Methods in the Flux Class: Case Study ex2 (Part 2)