# Applying Key Operators in the Parallel Flux Class: Case Study ex5 (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

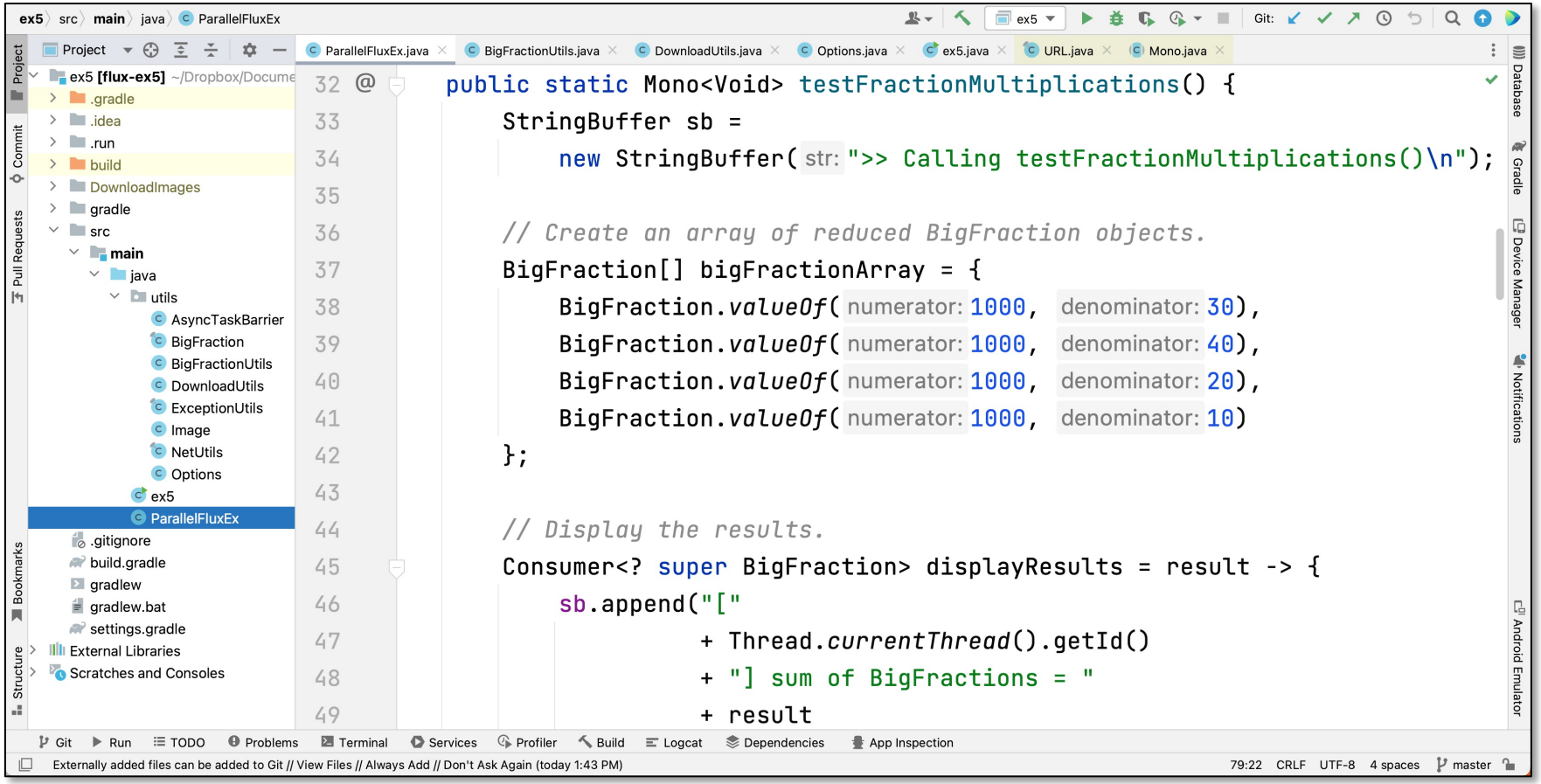

# Learning Objectives in this Part of the Lesson

- Part 1 of case study ex5 shows how to multiply & add big fractions asynchronously & concurrently using Project Reactor operators, e.g.,
  - Flux operators fromArray(), zipWith(), doOnNext(), doFinally(), then(), & parallel())
  - ParallelFlux operators runOn(), map(), & reduce())
  - The Schedulers.parallel() thread pool

```
return Flux
   .fromArray(bigFractionArray)
   .parallel()
   .runOn
      (Schedulers.parallel())
   .map(bf -> bf
        .multiply(sBigReducedFrac))
   .reduce(BigFraction::add)
   .doOnSuccess(displayResults)
   .then();
```

# Applying Key Operators in the ParallelFlux Class to ex5

# Applying Key Operators in the ParallelFlux Class to ex5




See github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/flux/ex5

# End of Applying Key Operators in the ParallelFlux Class: Case Study ex5 (Part 2)