Overview of the BigFraction Case Studies

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

• Understand key classes in the Project Reactor API
• Be aware of the structure & functionality of the BigFraction case studies
• These case studies showcase many operators in the Project Reactor Mono, Flux, & ParallelFlux classes

<<Java Class>>

BigFraction

- mNumerator: BigInteger
- mDenominator: BigInteger

- BigFraction()
- BigFraction(Number): BigFraction
- BigFraction(Short, Number): BigFraction
- BigFraction(Double, Number): BigFraction
- BigFraction(String, Number): BigFraction
- BigFraction(Number, Number, boolean): BigFraction
- reduce(BigFraction): BigFraction
- getNumerator(): BigInteger
- getDenominator(): BigInteger
- add(Number): BigFraction
- subtract(Number): BigFraction
- multiply(Number): BigFraction
- divide(Number): BigFraction
- gcd(Number): BigFraction
- toMixedString(): String
Overview of the BigFraction Class
Overview of the BigFraction Class

- Upcoming lessons show how to apply Project Reactor features in the context of a BigFraction class
- Arbitrary-precision fraction, utilizing BigIntegers for numerator & denominator

```java
<<Java Class>>

BigFraction

- mNumerator: BigInteger
- mDenominator: BigInteger

- BigFraction()
- static valueOf(Number): BigFraction
- static valueOf(Number, Number): BigFraction
- static valueOf(String): BigFraction
- static valueOf(Number, Number, boolean): BigFraction
- reduce(BigFraction): BigFraction
- getNumerator(): BigInteger
- getDenominator(): BigInteger
- add(Number): BigFraction
- subtract(Number): BigFraction
- multiply(Number): BigFraction
- divide(Number): BigFraction
- gcd(Number): BigFraction
- toMixedString(): String
```

See [LiveLessons/blob/master/Java8/ex8/src/utils/BigFraction.java](https://github.com/LiveLessons/blob/master/Java8/ex8/src/utils/BigFraction.java)
Overview of the BigFraction Class

- Upcoming lessons show how to apply Project Reactor features in the context of a BigFraction class
- Arbitrary-precision fraction, utilizing BigIntegers for numerator & denominator
- BigInteger provides arbitrary-precision integers & associated operators

See docs.oracle.com/javase/8/docs/api/java/math/BigInteger.html
Overview of the BigFraction Class

- Upcoming lessons show how to apply Project Reactor features in the context of a BigFraction class.
  - Arbitrary-precision fraction, utilizing BigIntegers for numerator & denominator.
- Factory methods to “reduce” fractions:
  - $\frac{44}{55} \rightarrow \frac{4}{5}$
  - $\frac{12}{24} \rightarrow \frac{1}{2}$
  - $\frac{144}{216} \rightarrow \frac{2}{3}$
Overview of the BigFraction Class

- Upcoming lessons show how to apply Project Reactor features in the context of a BigFraction class
  - Arbitrary-precision fraction, utilizing BigIntegers for numerator & denominator
  - Factory methods to "reduce" fractions
  - Factory methods to create "non-reduced" fractions (& then reduce them)
    - e.g., 12/24 ($\rightarrow$ 1/2)

```java
<<Java Class>>

BigFraction

- mNumerator: BigInteger
- mDenominator: BigInteger

BigFraction()

valueOf(Number): BigFraction
valueOf(Number, Number): BigFraction
valueOf(String): BigFraction

valueOf(Number, Number, boolean): BigFraction

reduce(BigFraction): BigFraction

getNumerator(): BigInteger
getDenominator(): BigInteger
add(Number): BigFraction
subtract(Number): BigFraction
multiply(Number): BigFraction
divide(Number): BigFraction
gcd(Number): BigFraction
toMixedString(): String
```
Overview of the BigFraction Class

- Upcoming lessons show how to apply Project Reactor features in the context of a BigFraction class
  - Arbitrary-precision fraction, utilizing BigIntegers for numerator & denominator
  - Factory methods to “reduce” fractions
  - Factory methods to create “non-reduced” fractions (& then reduce them)
- Arbitrary-precision fraction arithmetic
  - e.g., $18/4 \times 2/3 = 3$
Overview of the BigFraction Class

- Upcoming lessons show how to apply Project Reactor features in the context of a BigFraction class
  - Arbitrary-precision fraction, utilizing BigIntegers for numerator & denominator
  - Factory methods to “reduce” fractions
  - Factory methods to create “non-reduced” fractions (& then reduce them)
  - Arbitrary-precision fraction arithmetic
  - Create a mixed fraction from an improper fraction
    - e.g., $18/4 \rightarrow 4 \ 1/2$

See [www.mathsisfun.com/improper-fractions.html](http://www.mathsisfun.com/improper-fractions.html)
Overview of the BigFraction Case Studies
Overview of the BigFraction Case Studies

- These case studies show how to create, reduce, multiply, & display BigFraction objects synchronously, asynchronously, & concurrently using Project Reactor framework features.
Overview of the BigFraction Case Studies

- The Project Reactor Mono case studies show how to create, reduce, multiply, & display BigFraction objects using many Mono features
- e.g., fromCallable(), just(), zip(), zipWith(), doOnSuccess(), first(), when(), then(), subscribeOn(), & various thread pools

```java
BigFraction unreducedFraction = makeBigFraction(...);

return Mono
    .fromCallable(() -> BigFraction
        .reduce(unreducedFraction))
    .subscribeOn
        (Schedulers.single())
    .map(result ->
        result.toMixedString())
    .doOnSuccess(result ->
        System.out.println("big fraction = " + result + "\n"))
    .then();
```

See [github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/mono](https://github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/mono)
Overview of the BigFraction Case Studies

- The Project Reactor Flux case studies show how to create, reduce, multiply, & display BigFraction objects using many Flux features
  - e.g., fromIterable(), just(), map(), create(), doOnNext(), flatMap(), take(), interval(), subscribeOn(), collectList(), subscribe(), & various thread pools

```flux
Flux
  .create
    (bigFractionEmitter)
  .take(sMAX_FRACTIONS)
  .flatMap(unreducedFraction ->
    reduceAndMultiplyFraction
      (unreducedFraction,
        Schedulers.parallel()))
  .collectList()
  .flatMap(list ->
    BigFractionUtils
      .sortAndPrintList
        (list, sb));
```

Overview of the BigFraction Case Studies

- The Project Reactor Flux case studies show how to create, reduce, multiply, & display BigFraction objects using many Flux features
  - e.g., fromIterable(), just(), map(), create(), doOnNext(), flatMap(), take(), interval(), subscribeOn(), collectList(), subscribe(), & various thread pools
- They also demonstrate how the Java streams framework can be used together with the Project Reactor framework
End of Overview of the BigFraction Case Studies