Applying Key Operators in the Mono Class:
Case Study ex2 (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
Part 1 of case study ex2 applies the Mono operators `fromCallable()`, `map()`, `then()`, `doOnSuccess()`, `block()`, `subscribeOn()`, & `Schedulers.single()` to create, reduce, multiply, & display `BigFraction` objects asynchronously.

```java
return Mono
    .fromCallable(reduceFraction)
    .subscribeOn
    (Schedulers.single())
    .map(convertToMixedString)
    .doOnSuccess(printResult)
    .then();
```

See [github.com/douglasraicschmidt/LiveLessons/tree/master/Reactive/mono/ex2](https://github.com/douglasraicschmidt/LiveLessons/tree/master/Reactive/mono/ex2)
Part 1 of case study ex2 applies the Mono operators fromCallable(), map(), then(), doOnSuccess(), block(), subscribeOn(), & Schedulers.single() to create, reduce, multiply, & display BigFraction objects asynchronously.

• Both fully asynchronous & hybrid asynchronous/synchronous models are shown.

```java
BigFraction result = Mono
    .fromCallable(reduceFraction)
    .subscribeOn
    (Schedulers.single())
    .map(convertToMixedString)
    .block(sBLOCK_TIME);
```

...
Applying Key Methods in the Mono Class in ex2
Applying Key Methods in the Mono Class in ex2

See [github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/mono/ex2](https://github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/mono/ex2)