Key Suppressing Operators in the Mono Class (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

- Recognize key Mono operators
  - Factory method operators
  - Transforming operators
  - Action operators
  - Suppressing operators
    - These operators create a Mono that ignores its payload
      - e.g., then()

Key Suppressing Operators in the Mono Class

Key Suppressing Operators in the Mono Class

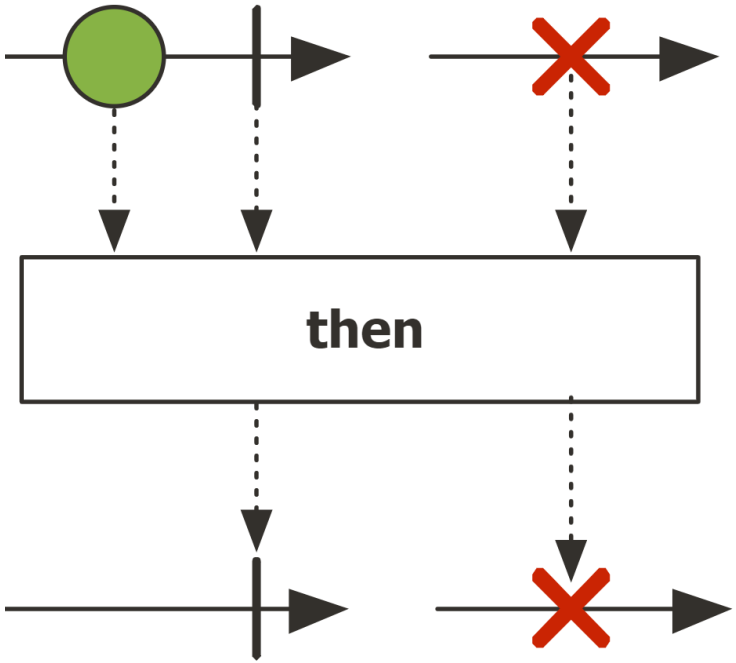
- The then() operator
  - Return a Mono<Void> that only replays complete & error signals from this Mono

```
Mono<Void> then()
```

See projectreactor.io/docs/core/release/api/reactor/core/publisher/Mono.html#then

Key Suppressing Operators in the Mono Class

- The then() operator
  - Return a Mono<Void> that only replays complete & error signals from this Mono
  - This "data-suppressing" operator ignores its payload



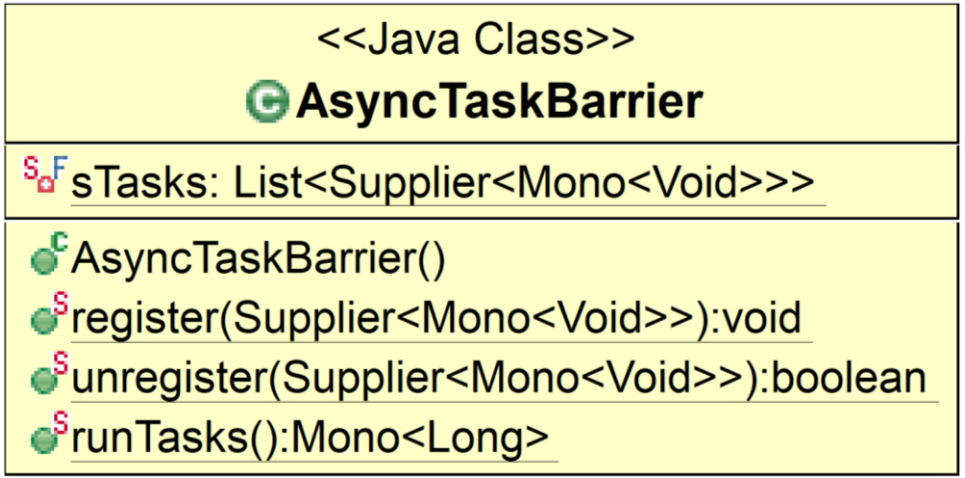

```
return Mono
   .fromCallable(() -> BigFraction
            .reduce(unreducedFrac))
   .doOnSuccess(bf -> logBigFrac
            (unreducedFrac, bf, sb))
   .map(BigFraction::toMixedString)
   .doOnSuccess(bf ->
   .then();
```

This empty mono synchronizes with AsyncTaskBarrier

See Reactive/mono/ex1/src/main/java/MonoEx.java

Key Suppressing Operators in the Mono Class

- The then() operator
  - Return a Mono<Void> that only replays complete & error signals from this Mono
  - This "data-suppressing" operator ignores its payload
    - It can be used to indicate when an async operation completes



then()'s semantics are leveraged by the AsyncTaskBarrier framework to ensure an async computation doesn't complete prematurely

See Reactive/mono/ex1/src/main/java/utils/AsyncTaskBarrier.java

Key Suppressing Operators in the Mono Class

- The then() operator
  - Return a Mono<Void> that only replays complete & error signals from this Mono
  - This "data-suppressing" operator ignores its payload
  - RxJava does not have a direct equivalent, though Completable can be used in a similar way

Class Completable

java.lang.Object
io.reactivex.rxjava3.core.Completable

All Implemented Interfaces:
CompletableSource

Direct Known Subclasses:
CompletableSubject

```
public abstract class Completable
extends Object
implements CompletableSource
```

The `Completable` class represents a deferred computation without any value but only indication for completion or exception.

`Completable` behaves similarly to `Observable` except that it can only emit either a completion or error signal (there is no `onNext` or `onSuccess` as with the other reactive types).

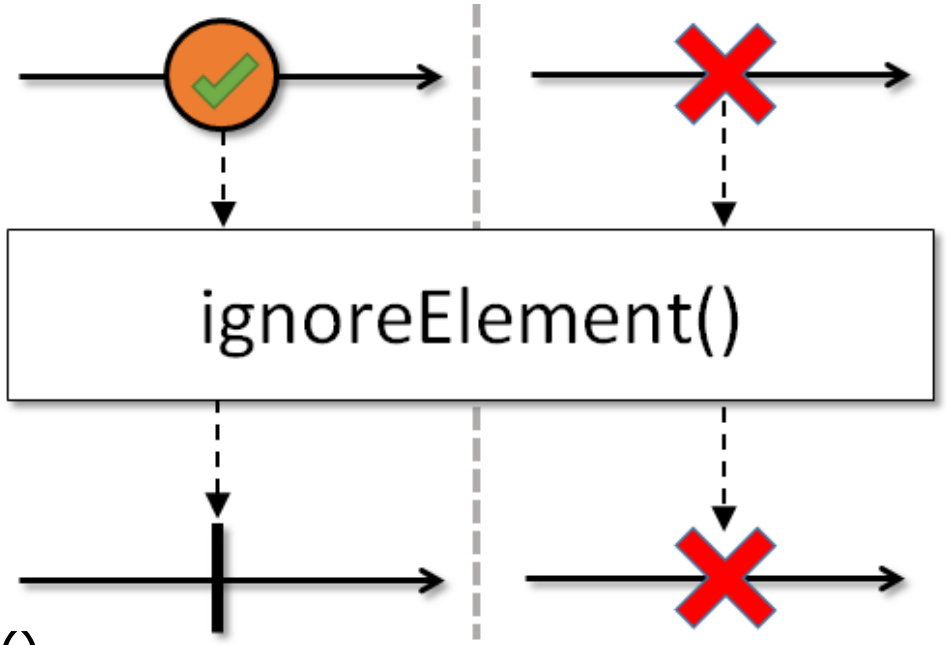
The `Completable` class implements the `CompletableSource` base interface and the default consumer type it interacts with is the `CompletableObserver` via the `subscribe(CompletableObserver)` method. The `Completable` operates with the following sequential protocol:

```
onSubscribe (onError | onComplete)?
```

See reactivex.io/RxJava/3.x/javadoc/io/reactivex/rxjava3/core/Completable.html

Key Suppressing Operators in the Mono Class

- The then() operator
  - Return a Mono<Void> that only replays complete & error signals from this Mono
  - This "data-suppressing" operator ignores its payload
  - RxJava does not have a direct equivalent, though Completable can be used in a similar way, e.g.
    - Created via Single.ignoreElement()
      - Returns a Completable that ignores the success value of this Single & signals onComplete() instead



See reactivex.io/RxJava/3.x/javadoc/io/reactivex/rxjava3/core/Single.html#ignoreElement

End of Key Suppressing Operators in the Mono Class (Part 1)