The Java Streams collect() Terminal Operation (Part 3)

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 common terminal operations, e.g.
  - forEach()
  - collect()
    - Know what a collector does
    - Recognize common Java pre-defined collectors & how to use them with collect()
    - Be aware of the Teeing Collector
      - Introduced in Java 12

teeing

```
public static <T,R1,R2,R> Collector<T,?,R> teeing(Collector<? super T,?,R1> downstream1, Collector<? super T,?,R2> downstream2, BiFunction<? super R1,? super R2,R> merger)
```

Returns a `Collector` that is a composite of two downstream collectors. Every element passed to the resulting collector is processed by both downstream collectors, then their results are merged using the specified merge function into the final result.

The resulting collector functions do the following:

- supplier: creates a result container that contains result containers obtained by calling each collector's supplier
- accumulator: calls each collector's accumulator with its result container and the input element
- combiner: calls each collector's combiner with two result containers
- finisher: calls each collector's finisher with its result container, then calls the supplied merger and returns its result.

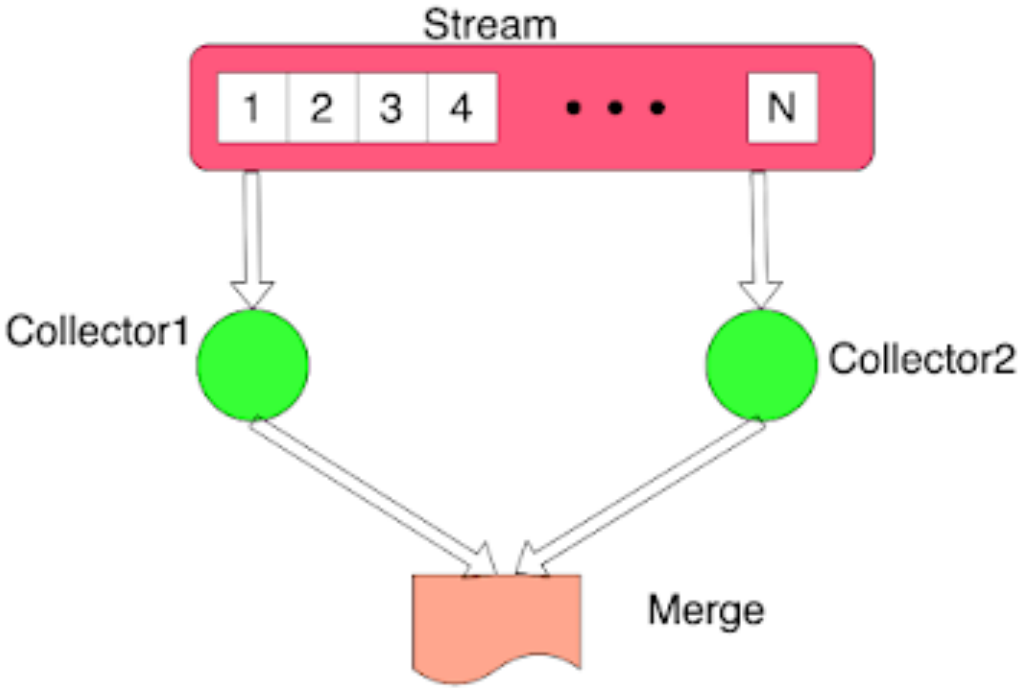
The resulting collector is `Collector.Characteristics.UNORDERED` if both downstream collectors are unordered and `Collector.Characteristics.CONCURRENT` if both downstream collectors are concurrent.

See docs.orade.com/en/java/javase/12/docs/api/java.base/java/util/stream/Collectors.html#teeing

Overview of the Teeing Collector

Overview of the Teeing Collector

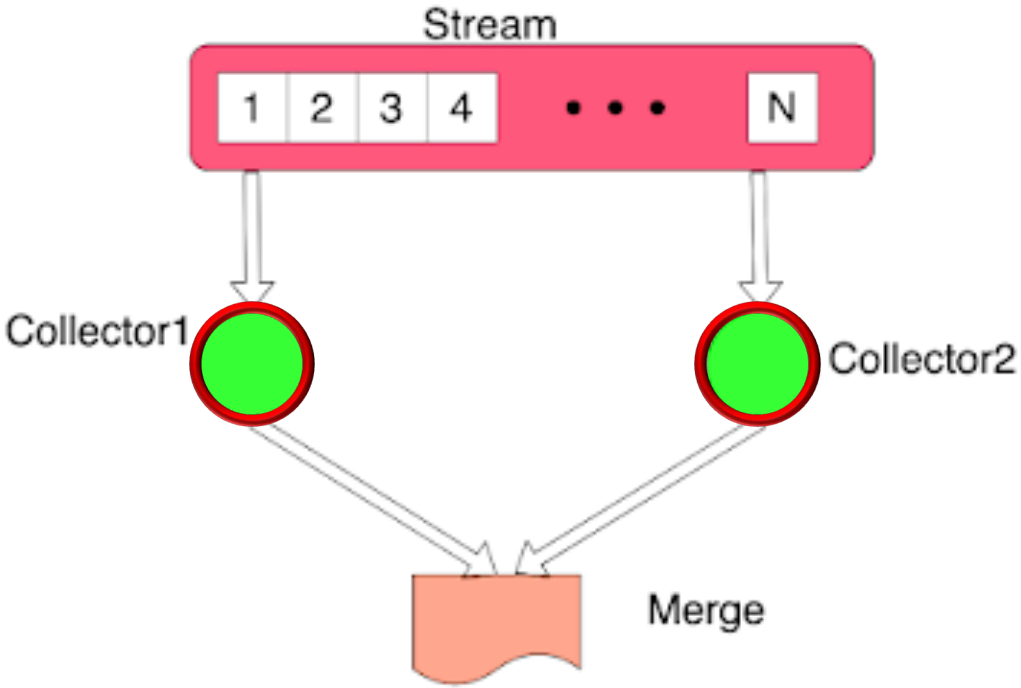
- The Teeing Collector returns a Collector that composes two downstream collectors



See docs.orade.com/en/java/javase/12/docs/api/java.base/java/util/stream/Collectors.html#teeing

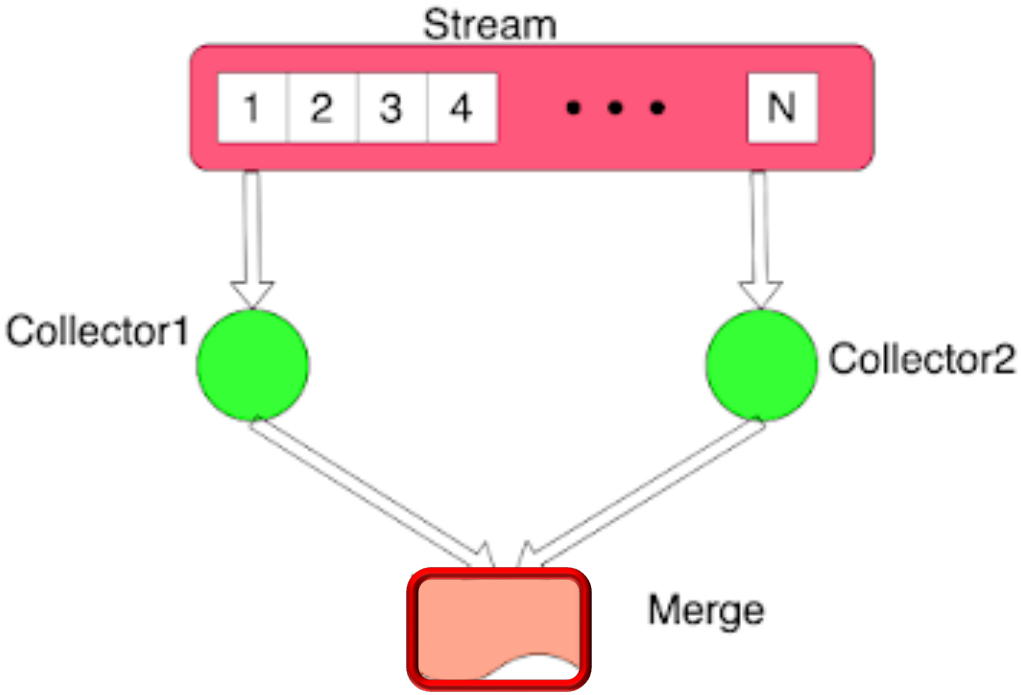
Overview of the Teeing Collector

- The Teeing Collector returns a Collector that composes two downstream collectors
  - Every element passed to the resulting collector is processed by both downstream collectors



Overview of the Teeing Collector

- The Teeing Collector returns a Collector that composes two downstream collectors
  - Every element passed to the resulting collector is processed by both downstream collectors
  - Their results are then merged using the specified merge function into the final result



See dzone.com/articles/java-12-the-teeing-collector

Applying the Teeing Collector

Applying the Teeing Collector

- Merge a stream so 'H'/'h' Hamlet characters comes first followed by the non-'H'/'h' characters

```
void runTeeingCollector() {
  var results = sCharacters
    .stream()
    .map(Generators::capitalize)
    .sorted()
    .collect(teeing
      (filtering
         (startsWithHh(true),
          toList()),
       filtering
          (startsWithHh(false),
           toList()),
       Utils::concat);
```

See github.com/douglascraigschmidt/LiveLessons/tree/master/Java8/ex12

Applying the Teeing Collector

- Merge a stream so 'H'/'h' Hamlet characters comes first followed by the non-'H'/'h' characters

Create a stream of all the Hamlet characters

```
void runTeeingCollector() {
  var results = sCharacters
     .stream()
     .map(Generators::capitalize)
     .sorted()
     .collect(teeing
       (filtering
          (startsWithHh(true),
           toList()),
        filtering
           (startsWithHh(false),
            toList()),
        Utils::concat);
```

Applying the Teeing Collector

- Merge a stream so 'H'/'h' Hamlet characters comes first followed by the non-'H'/'h' characters

Capitalize Hamlet characters consistently

```
void runTeeingCollector() {
  var results = sCharacters
     .stream()
     .map(Generators::capitalize)
     .sorted()
     .collect(teeing
       (filtering
          (startsWithHh(true),
           toList()),
        filtering
           (startsWithHh(false),
            toList()),
        Utils::concat);
```

Applying the Teeing Collector

- Merge a stream so 'H'/'h' Hamlet characters comes first followed by the non-'H'/'h' characters

Sort Hamlet characters in ascending order

```
void runTeeingCollector() {
  var results = sCharacters
     .stream()
     .map(Generators::capitalize)
     .sorted()
     .collect(teeing
       (filtering
          (startsWithHh(true),
           toList()),
        filtering
           (startsWithHh(false),
            toList()),
        Utils::concat);
```

Applying the Teeing Collector

- Merge a stream so 'H'/'h' Hamlet characters comes first followed by the non-'H'/'h' characters

Merge the results from two Collector operations on a Stream

```
void runTeeingCollector() {
  var results = sCharacters
     .stream()
     .map(Generators::capitalize)
     .sorted()
     .collect(teeing
        (filtering
           (startsWithHh(true),
            toList()),
         filtering
            (startsWithHh(false),
             toList()),
         Utils::concat);
...
```

See docs.oracle.com/en/java/javase/12/docs/api/java.base/java/util/stream/Collectors.html#teeing

Applying the Teeing Collector

- Merge a stream so 'H'/'h' Hamlet characters comes first followed by the non-'H'/'h' characters

```
void runTeeingCollector() {
  var results = sCharacters
    .stream()
    .map(Generators::capitalize)
    .sorted()
    .collect(teeing
      (filtering
        (startsWithHh(true),
         toList()),
       filtering
         (startsWithHh(false),
          toList()),
       Utils::concat);
...
```

Collect a List of characters that start with 'H' or 'h'

See docs.oracle.com/en/java/javase/12/docs/api/java.base/java/util/stream/Collectors.html#filtering

Applying the Teeing Collector

- Merge a stream so 'H'/'h' Hamlet characters comes first followed by the non-'H'/'h' characters

```
void runTeeingCollector() {
  var results = sCharacters
    .stream()
    .map(Generators::capitalize)
    .sorted()
    .collect(teeing
      (filtering
         (startsWithHh(true),
          toList()),
       filtering
          (startsWithHh(false),
           toList()),
       Utils::concat);
...
```

Collect a List of characters that don't start with 'H' or 'h'

See docs.oracle.com/en/java/javase/12/docs/api/java.base/java/util/stream/Collectors.html#filtering

Applying the Teeing Collector

- Merge a stream so 'H'/'h' Hamlet characters comes first followed by the non-'H'/'h' characters

```
void runTeeingCollector() {
  var results = sCharacters
     .stream()
     .map(Generators::capitalize)
     .sorted()
     .collect(teeing
       (filtering
          (startsWithHh(true),
           toList()),
        filtering
           (startsWithHh(false),
            toList()),
        Utils::concat);
...
```

Merge the two lists together

See Java8/ex12/src/main/java/utils/Utils.java

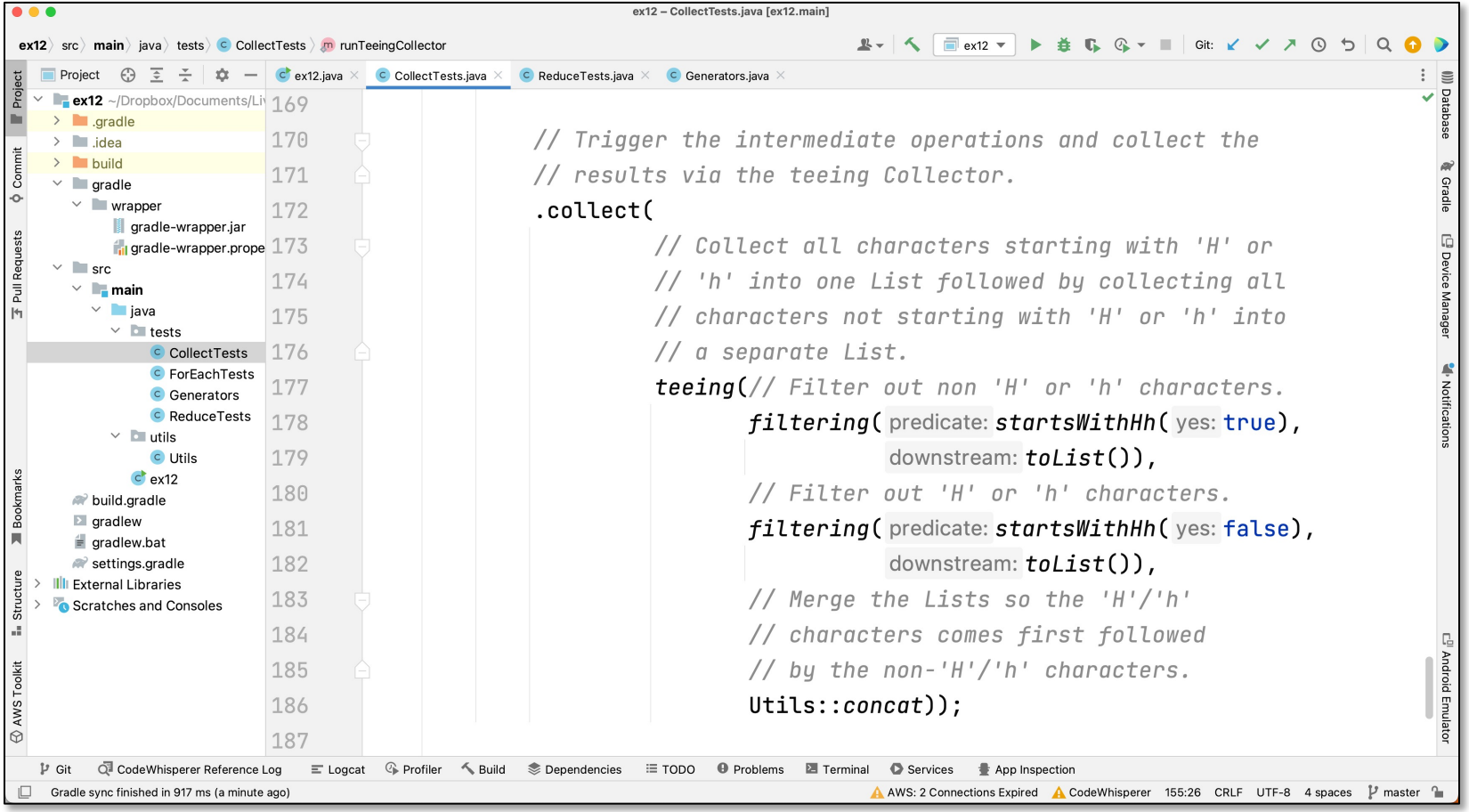
Applying the Teeing Collector

- Merge a stream so 'H'/'h' Hamlet characters comes first followed by the non-'H'/'h' characters

```
void runTeeingCollector() {
  var results = sCharacters
     .stream()
     .map(Generators::capitalize)
     .sorted()
     .collect(teeing
        (filtering
           (startsWithHh(true),
            toList()),
         filtering
            (startsWithHh(false),
             toList()),
         Utils::concat);
...
```

Return the merged lists

Applying the Teeing Collector



See github.com/douglascraigschmidt/LiveLessons/tree/master/Java8/ex12

End of the Java Streams collect() Terminal Operation (Part 3)