# **Java 8 Stream Factory Methods**

Douglas C. Schmidt <u>d.schmidt@vanderbilt.edu</u> www.dre.vanderbilt.edu/~schmidt



**Professor of Computer Science** 

Institute for Software Integrated Systems

Vanderbilt University Nashville, Tennessee, USA



# Learning Objectives in this Lesson

• Recognize common stream factory methods



• Streams can be obtained various ways



See <a href="https://docs/api/java/util/stream/package-summary.html">docs.oracle.com/javase/8/docs/api/java/util/stream/package-summary.html</a>

- Streams can be obtained various ways, e.g.,
  - From a Java collection

```
List<String> wordsToFind =
   Arrays.asList("do", "re", "me", ...);
```

List<SearchResults> results =
wordsToFind.stream()

• •

or

List<SearchResults> results =
 wordsToFind.parallelStream()

• • •

- Streams can be obtained various ways, e.g.,
  - From a Java collection

List<String> wordsToFind =
 Arrays.asList("do", "re", "me", ...);

List<SearchResults> results =
 wordsToFind.stream()

• • •

See <a href="https://docs.oracle.com/javase/tutorial/collections/streams">docs.oracle.com/javase/tutorial/collections/streams</a>

- Streams can be obtained various ways, e.g.,
  - From a Java collection

List<String> wordsToFind =
 Arrays.asList("do", "re", "me", ...);

List<SearchResults> results =
 wordsToFind.stream()

• • •

or

List<SearchResults> results =
 wordsToFind.parallelStream()

• • •

See <a href="https://docs.oracle.com/javase/tutorial/collections/streams/parallelism.html">docs.oracle.com/javase/tutorial/collections/streams/parallelism.html</a>

- Streams can be obtained various ways, e.g.,
  - From a Java collection

List<String> wordsToFind =
 Arrays.asList("do", "re", "me", ...);

List<SearchResults> results =
 wordsToFind.stream()

A call to parallel() can appear anywhere in a stream & will have same effect as parallelStream()

or

List<SearchResults> results =
wordsToFind.stream()

.parallel()

See <u>docs.oracle.com/javase/8/docs/api/java/util/stream/BaseStream.html#parallel</u>

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array

```
String[] array = {
   "a", "b", "c", "d", "e"
};
Stream<String> stream =
   Arrays.stream(array);
```

```
stream.forEach(s ->
    System.out.println(s));
```

```
or
```

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array

```
String[] array = {
    "a", "b", "c", "d", "e"
};
```

Stream<String> stream =

```
Arrays.stream(array);
```

stream.forEach(s ->

*Get & print all the elements in an array* 

System.out.println(s));

or

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array

```
String[] array = {
    "a", "b", "c", "d", "e"
};
Stream<String> stream =
```

```
Arrays.stream(array);
```

stream.forEach(s ->

*Get & print all the elements in an array* 

System.out.println(s));

stream.forEach(System.out::println);

or

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method

```
String[] array = {
  "a", "b", "c", "d", "e"
};
Stream<String> s = Stream.of(array);
s.forEach(s -> System.out.println(s));
or
```

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method

Get & print all

the elements

in an array

String[] array = { "a", "b", "c", "d", "e" **};** Stream<String> s = Stream.of(array); s.forEach(s -> System.out.println(s)); or stream.forEach(System.out::println);

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method

Get & print all

the elements

in an array

String[] array = { "a", "b", "c", "d", "e" **};** Stream<String> s = Stream.of(array); s.forEach(s -> System.out.println(s)); or stream.forEach(System.out::println);

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method

Do not use iterate() in a parallel stream! Use range() instead..

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method

*Generate & print first 100 Fibonacci numbers* 

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method

*Generate & print first 100 Fibonacci numbers* 

.limit(100)

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method

*Generate & print first 100 Fibonacci numbers* 

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method

*Generate & print first 100 Fibonacci numbers* 

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method

*Generate & print first 100 Fibonacci numbers* 

.limit(100)

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method

*Generate & print first 100 Fibonacci numbers* 

```
.forEach(System.out::println);
```

See <u>docs.oracle.com/javase/8/docs/api/java/util/stream/Stream.html#limit</u>

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method

*Generate & print first 100 Fibonacci numbers* 

```
.forEach(System.out::println);
```

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method
  - BufferedReader.lines() obtains lines of a file

void printFileLines (String filename){ try (BufferedReader reader = Files.newBufferedReader (Paths.get(filename)) { reader .lines() .forEach (System.out::println); catch (IOException ex) {...} Get & print all lines in a file

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method
  - BufferedReader.lines() obtains lines of a file

void printFileLines (String filename){ try (BufferedReader reader = Files.newBufferedReader (Paths.get(filename)) { reader .lines() .forEach (System.out::println); catch (IOException ex) {...} Get & print all lines in a file

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method
  - BufferedReader.lines() obtains lines of a file
  - Streams of file paths & lines can be obtained from Files methods



- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method
  - BufferedReader.lines() obtains lines of a file
  - Streams of file paths & lines can be obtained from Files methods
  - A stream of random numbers can be obtained from Random.ints()

```
new Random()
.ints(0,100)
.limit(50)
.forEach(System.out::println);
```

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method
  - BufferedReader.lines() obtains lines of a file
  - Streams of file paths & lines can be obtained from Files methods
  - A stream of random numbers can be obtained from Random.ints()

Generate & print 50
random numbers
between 0 & 100

new Random()
.ints(0,100)
.limit(50)
.forEach(System.out::println);

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method
  - BufferedReader.lines() obtains lines of a file
  - Streams of file paths & lines can be obtained from Files methods
  - A stream of random numbers can be obtained from Random.ints()

Generate & print 50
random numbers
between 0 & 100

new Random()
.ints(0,100)
.limit(50)
.forEach(System.out::println);

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method
  - BufferedReader.lines() obtains lines of a file
  - Streams of file paths & lines can be obtained from Files methods
  - A stream of random numbers can be obtained from Random.ints()



- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method
  - BufferedReader.lines() obtains lines of a file
  - Streams of file paths & lines can be obtained from Files methods
  - A stream of random numbers can }
     be obtained from Random.ints()
  - Other JDK stream-bearing methods

```
(String filename,
        String splitter){
return Pattern
.compile(splitter)
.splitAsStream
   (new String
   (Files.readAllBytes
        (Paths.get(filename)
        .toURI())));
```

Stream<String> getInputData

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method
  - BufferedReader.lines() obtains lines of a file
  - Streams of file paths & lines can be obtained from Files methods
  - A stream of random numbers can }
     be obtained from Random.ints()
  - Other JDK stream-bearing methods

```
Stream<String> getInputData
                (String filename,
                 String splitter){
  return Pattern
   .compile(splitter)
   .splitAsStream
     (new String
       (Files.readAllBytes
           (Paths.get(filename)
            .toURI()));
      Splits a file into a
       stream of strings
```

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method
  - BufferedReader.lines() obtains lines of a file
  - Streams of file paths & lines can be obtained from Files methods
  - A stream of random numbers can be obtained from Random.ints()
  - Other JDK stream-bearing methods



See <a href="https://docs/api/java/util/stream/Stream.html#generate">https://docs/api/java/util/stream/Stream.html#generate</a>

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method
  - BufferedReader.lines() obtains lines of a file
  - Streams of file paths & lines can be obtained from Files methods
  - A stream of random numbers can be obtained from Random.ints()
  - Other JDK stream-bearing methods



- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method
  - BufferedReader.lines() obtains lines of a file
  - Streams of file paths & lines can be obtained from Files methods
  - A stream of random numbers can be obtained from Random.ints()
  - Other JDK stream-bearing methods

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method
  - BufferedReader.lines() obtains lines of a file
  - Streams of file paths & lines can be obtained from Files methods
  - A stream of random numbers can be obtained from Random.ints()
  - Other JDK stream-bearing methods

Li

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method
  - BufferedReader.lines() obtains lines of a file
  - Streams of file paths & lines can be obtained from Files methods
  - A stream of random numbers can be obtained from Random.ints()
  - Other JDK stream-bearing methods

- Streams can be obtained various ways, e.g.,
  - From a Java collection
  - From an array
  - From a static factory method
  - BufferedReader.lines() obtains lines of a file
  - Streams of file paths & lines can be obtained from Files methods
  - A stream of random numbers can be obtained from Random.ints()
  - Other JDK stream-bearing methods

```
We'll use this idiom in Java 8 programs covered later
```

```
List<TreeMap<Long, String>>
  listOfTreeMaps =
    Stream.generate
    (TreeMap<Long, String>::new)
    .limit(100)
    .collect(toList());
```

# End of Java 8 Stream Factory Methods