Java Streams: Other Factory Methods

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

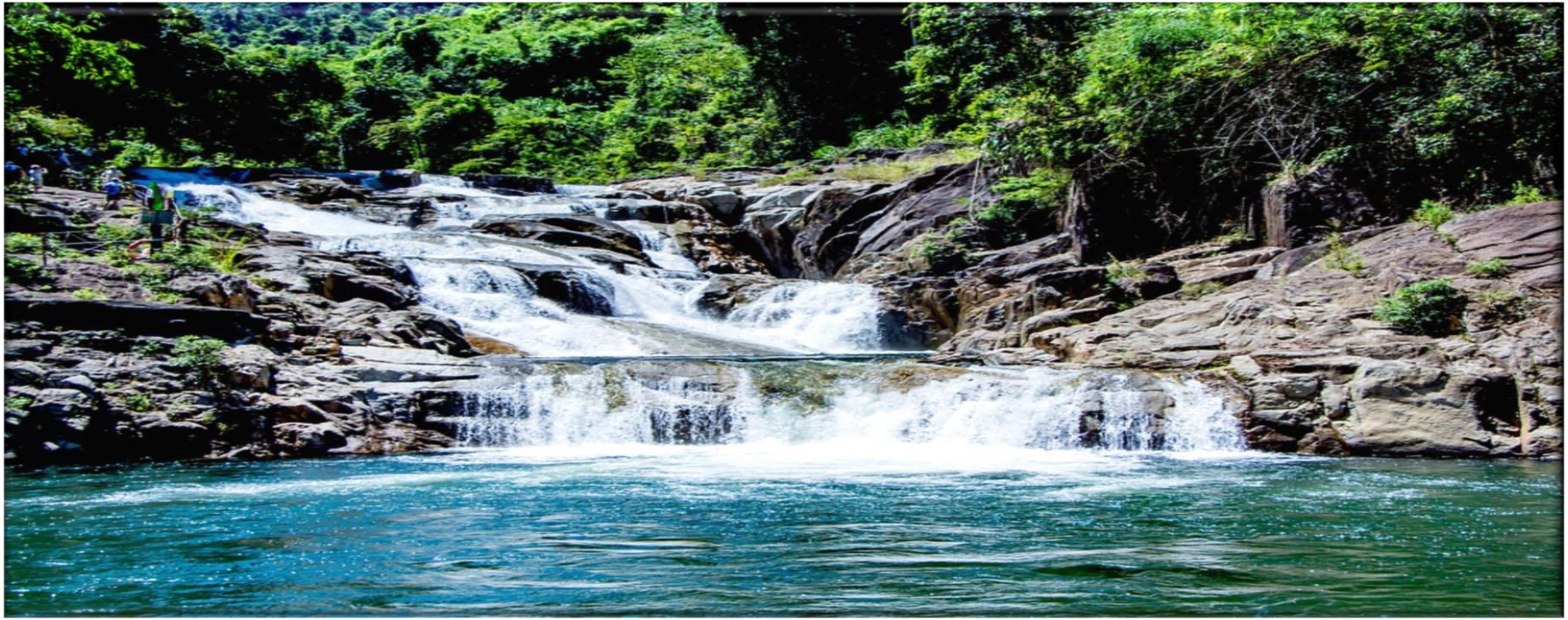
Professor of Computer Science

Institute for Software
Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson

- Understand common factory methods used to create streams
- Recognize other factory methods used to create streams



Other Factory Methods for Creating Streams

Other Factory Methods for Creating Streams

- There are several other ways to obtain a stream

Other Factory Methods for Creating Streams

- There are several other ways to obtain a stream, e.g.
  - I/O stream classes, e.g.
    - 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) {...}
}
```

Create a buffered reader from a given filename.

See docs.oracle.com/javase/8/docs/api/java/io/BufferedReader.html

Other Factory Methods for Creating Streams

- There are several other ways to obtain a stream, e.g.
  - I/O stream classes, e.g.
    - 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) {...}
}
```

The buffered reader will be closed automatically after the try-with-resources block exits.

See docs.oracle.com/javase/tutorial/essential/exceptions/tryResourceClose.html

Other Factory Methods for Creating Streams

- There are several other ways to obtain a stream, e.g.
  - I/O stream classes, e.g.
    - 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) {...}
}
```

Create stream containing all of the lines in a file.

See docs.oracle.com/javase/8/docs/api/java/io/BufferedReader.html#lines

Other Factory Methods for Creating Streams

- There are several other ways to obtain a stream, e.g.
  - I/O stream classes, e.g.
    - 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) {...}
}
```

Print each of the lines in the stream.

Other Factory Methods for Creating Streams

- There are several other ways to obtain a stream, e.g.
  - I/O stream classes, e.g.
    - BufferedReader.lines() obtains lines of a file
    - Streams of file paths & lines can be obtained from Files methods

```
void printFileLines
                    (String filename){
  try(Stream<String> stream =
      Files.lines
        (Paths.get(fileName))) {
     stream.forEach
        (System.out::println);
  } catch (IOException ex) {...}
}
```

Create stream containing all of the lines in a file

See docs.oracle.com/javase/8/docs/api/java/nio/file/Files.html#lines

Other Factory Methods for Creating Streams

- There are several other ways to obtain a stream, e.g.
  - I/O stream classes, e.g.
    - BufferedReader.lines() obtains lines of a file
    - Streams of file paths & lines can be obtained from Files methods

```
void printFileLines
                    (String filename){
  try(Stream<String> stream =
      Files.lines
        (Paths.get(fileName))) {
     stream.forEach
       (System.out::println);
  } catch (IOException ex) {...}
}
```

Print each of the lines in the stream

Other Factory Methods for Creating Streams

- There are several other ways to obtain a stream, e.g.
  - A stream of random #'s can be obtained from Random.ints()

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

Other Factory Methods for Creating Streams

- There are several other ways to obtain a stream, e.g.
  - A stream of random #'s can be obtained from Random.ints()

Generate an "unbounded" stream of random #'s ranging between 0 & 100

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

See docs.oracle.com/javase/8/docs/api/java/util/Random.html#ints

Other Factory Methods for Creating Streams

- There are several other ways to obtain a stream, e.g.
  - A stream of random #'s can be obtained from Random.ints()

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

Limit the size of the stream to 50 elements

Other Factory Methods for Creating Streams

- There are several other ways to obtain a stream, e.g.
  - A stream of random #'s can be obtained from Random.ints()

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

Print each random # in the stream

Other Factory Methods for Creating Streams

- There are several other ways to obtain a stream, e.g.
  - StreamSupport.stream() factory method

```
SearchResults searchForPhrase
   (String phrase, CharSequence input,
    String title, boolean parallel) {
   return new SearchResults
      (..., phrase, ..., StreamSupport
        .stream(new PhraseMatchSpliterator
                               (input, phrase),
                parallel)
        .collect(toList()));
}
```

Create a stream that contains all the phrases that match in an input string

See docs.oracle.com/javase/8/docs/api/java/util/stream/StreamSupport.html#stream

Other Factory Methods for Creating Streams

- There are several other ways to obtain a stream, e.g.
  - 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

See docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html#splitAsStream

Other Factory Methods for Creating Streams

- There are several other ways to obtain a stream, e.g.
  - Other JDK stream-bearing methods

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

Generate an "infinite" stream of TreeMaps

See docs.oracle.com/javase/8/docs/api/java/util/stream/Stream.html#generate

Other Factory Methods for Creating Streams

- There are several other ways to obtain a stream, e.g.
  - Other JDK stream-bearing methods

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

Limit the stream to 100 elements

Other Factory Methods for Creating Streams

- There are several other ways to obtain a stream, e.g.
  - Other JDK stream-bearing methods

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

Create a list of 100 TreeMaps

End of Java Streams: Other Factory Methods