Understanding Java Streams Common Creation Operations

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

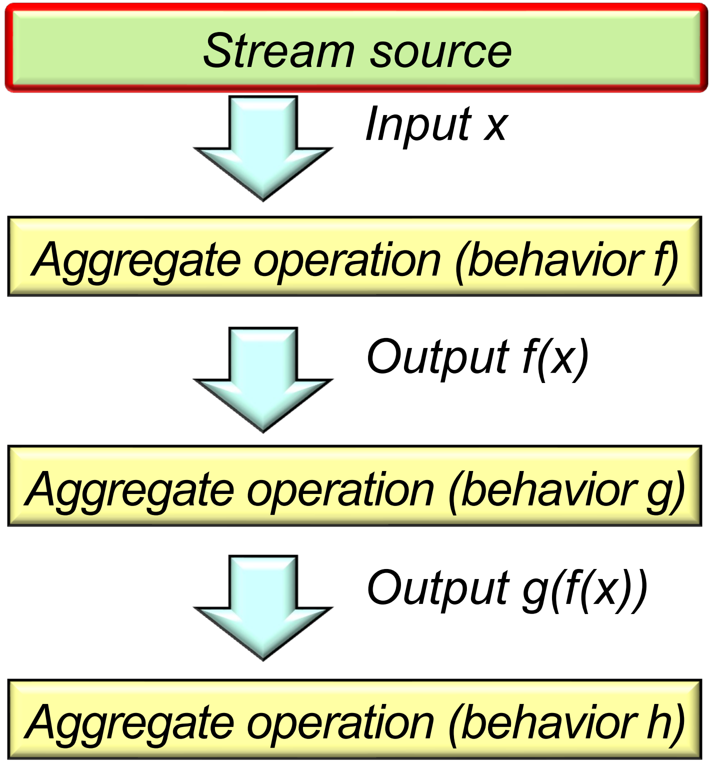
Professor of Computer Science

Institute for Software Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson

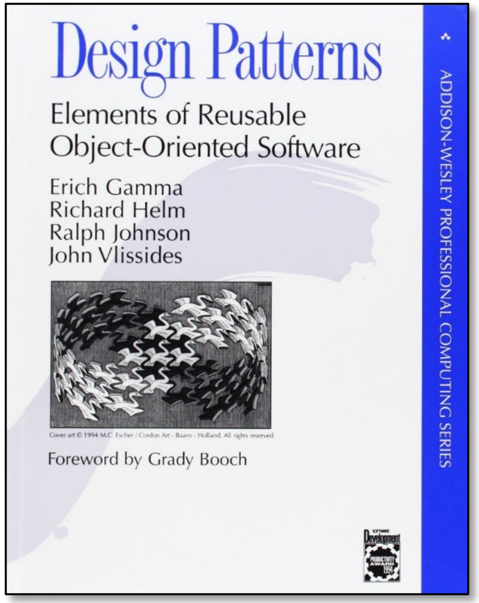
- Understand Java streams structure & functionality, e.g.
  - Fundamentals of streams
  - Three streams phases
  - Operations that create a stream

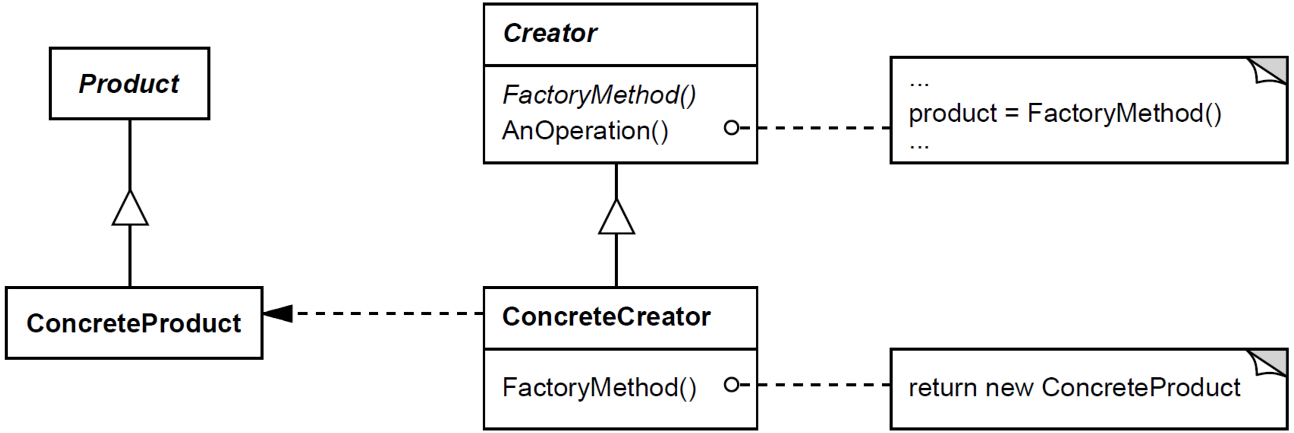

Stream source
Input x
Aggregate operation (behavior f)
Output f(x)
Aggregate operation (behavior g)
Output g(f(x))
Aggregate operation (behavior h)

Operations that Create a Java Stream

Operations that Create a Java Stream

- The GoF *Factory Method* pattern defines an interface for creating an object, but shields implementation details from clients of this interface


Design Patterns
Elements of Reusable Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Foreword by Grady Booch
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

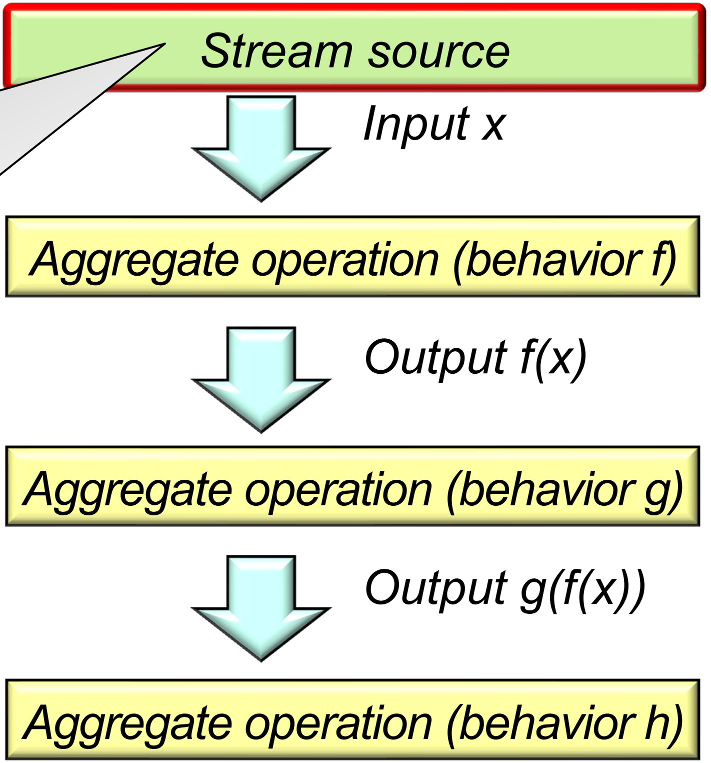


See en.wikipedia.org/wiki/Factory_method_pattern

Operations that Create a Java Stream

- Java Streams use factory methods to create a stream from some source

```
Stream
  .of("horatio",
      "laertes",
      "Hamlet", ...)
  ...
```

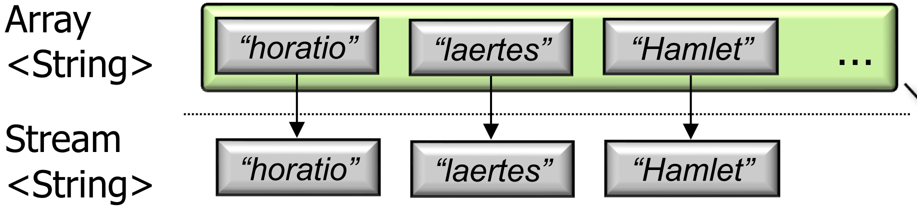

Stream source
Input x
Aggregate operation (behavior f)
Output f(x)
Aggregate operation (behavior g)
Output g(f(x))
Aggregate operation (behavior h)

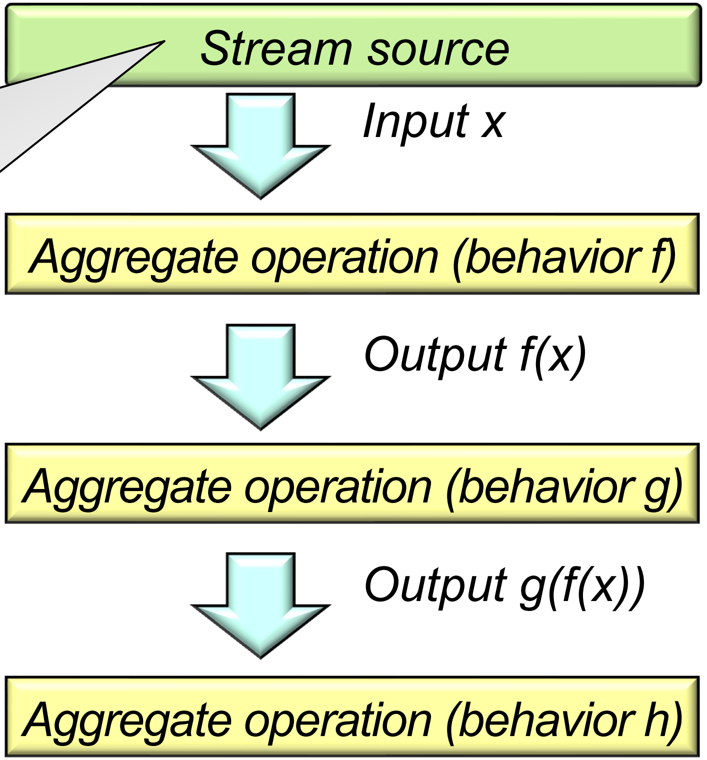
See docs.oracle.com/javase/8/docs/api/java/util/stream/Stream.html#of

Operations that Create a Java Stream

- Java Streams use factory methods to create a stream from some source

```
Stream
  .of("horatio",
      "laertes",
      "Hamlet", ...) ...
```





Stream source
Input x
Aggregate operation (behavior f)
Output f(x)
Aggregate operation (behavior g)
Output g(f(x))
Aggregate operation (behavior h)

The Stream.of() factory method converts an array of T into a stream of T

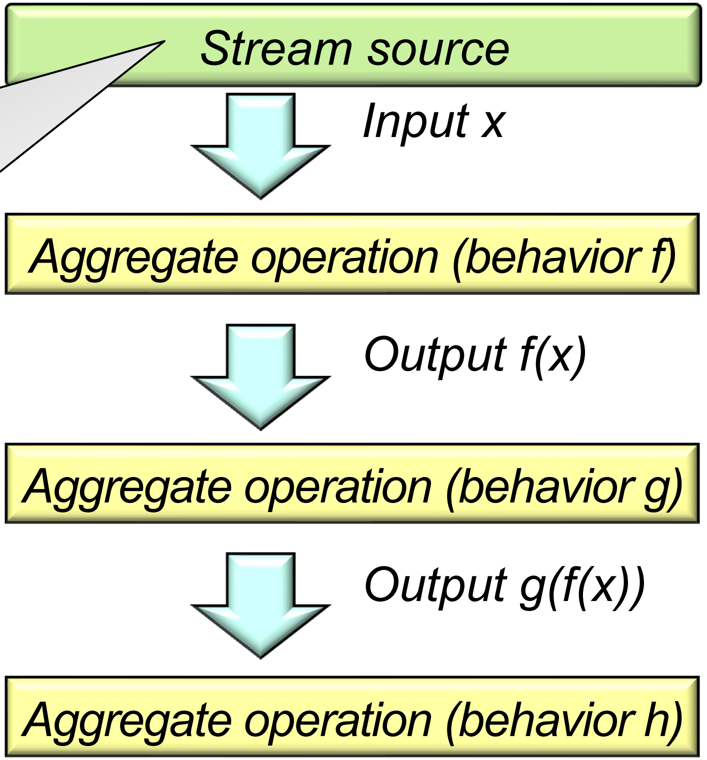
See docs.oracle.com/javase/8/docs/api/java/util/stream/Stream.html#of

Operations that Create a Java Stream

- Java Streams use factory methods to create a stream from some source

```
Stream
   .of("claudius")
   ...
   .findFirst().orElse(...);
```

It's perfectly reasonable to use Stream.of() to create a stream with one element in it

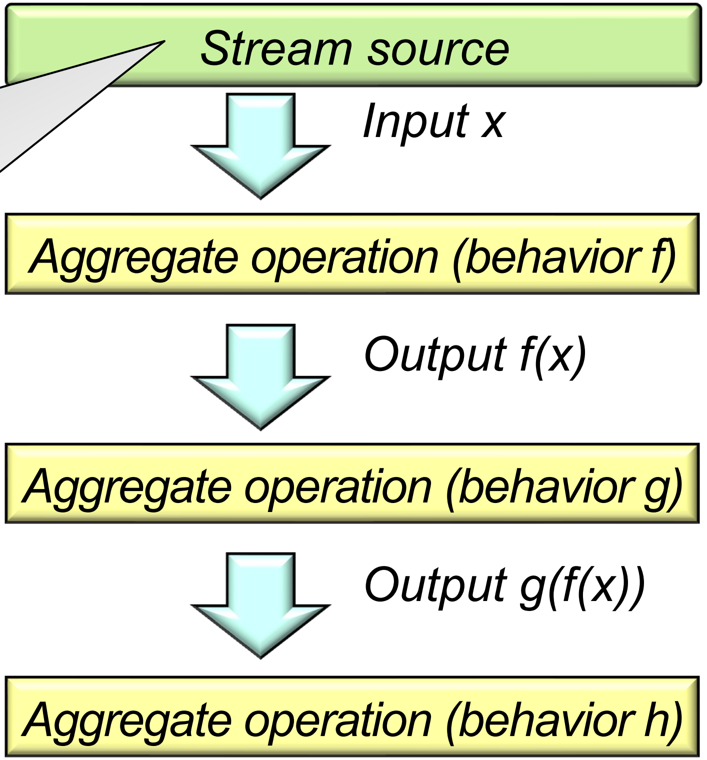

Stream source
Input x
Aggregate operation (behavior f)
Output f(x)
Aggregate operation (behavior g)
Output g(f(x))
Aggregate operation (behavior h)

Operations that Create a Java Stream

- Java Streams use factory methods to create a stream from some source

```
Stream
   .of("claudius")
   ...
   .findFirst().orElse(...);
```

Stream operations like filter() and/or map() can be applied to that single stream element



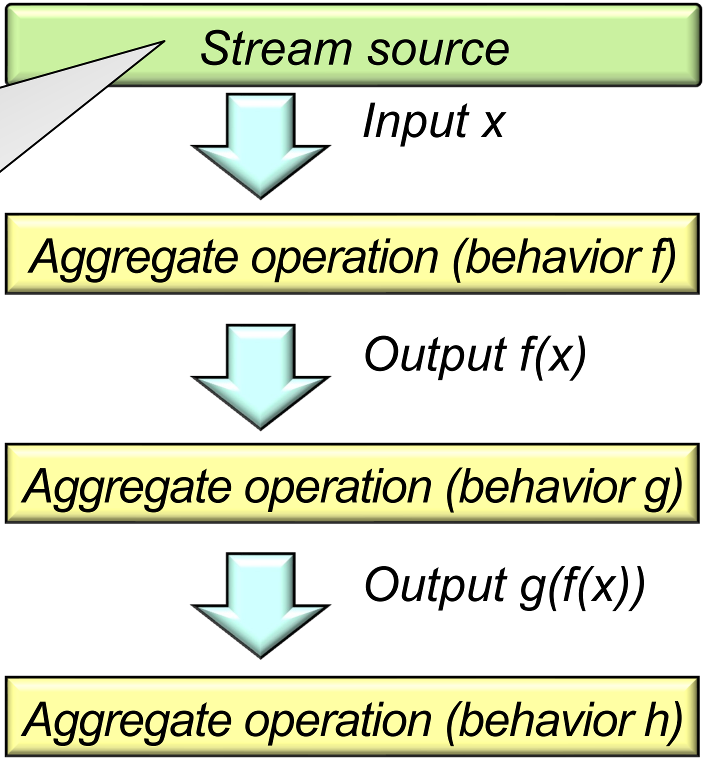
Stream source
Input x
Aggregate operation (behavior f)
Output f(x)
Aggregate operation (behavior g)
Output g(f(x))
Aggregate operation (behavior h)

Operations that Create a Java Stream

- Java Streams use factory methods to create a stream from some source

```
Stream
   .of("claudius")
   ...
   .findFirst().orElse(...);
```

Can be used in conjunction with findFirst().orElse(...) to obtain the update element


Stream source
Input x
Aggregate operation (behavior f)
Output f(x)
Aggregate operation (behavior g)
Output g(f(x))
Aggregate operation (behavior h)

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

Operations that Create a Java Stream

- Many factory methods create streams

```
collection.stream()
collection.parallelStream()
```

These are the most common factory methods used in Streams

```
Pattern.compile(...)
        .splitAsStream()
Stream.of(value1, ..., valueN)
StreamSupport.stream
   (iterable.spliterator(), ...)
...
```

```
Arrays.stream(array)
Arrays.stream(array, start, end)
Files.lines(file_path)
"string".chars()
Stream.iterate(init_value,
               generate_expression)
Stream.builder().add(...).build()
Stream.generate(supplier)
Files.list(file_path)
Files.find(file_path, max_depth,
           matcher)
...
```

These methods are inherited from the Java Collection interface

Operations that Create a Java Stream

- Many factory methods create streams

```
collection.stream()
collection.parallelStream()
Pattern.compile(...)
        .splitAsStream()
Stream.of(value1, ..., valueN)
StreamSupport.stream
   (iterable.spliterator(), ...)
...
```

```
Arrays.stream(array)
Arrays.stream(array, start, end)
Files.lines(file_path)
"string".chars()
Stream.iterate(init_value,
               generate_expression)
Stream.builder().add(...).build()
Stream.generate(supplier)
Files.list(file_path)
Files.find(file_path, max_depth,
           matcher)
...
```

We show examples of these types of factory methods throughout the course

See the upcoming lesson on "*Java Streams: Common Factory Methods*"

Operations that Create a Java Stream

- Many factory methods create streams

```
collection.stream()
collection.parallelStream()
```

This factory method implements these two factory methods

```
Pattern.compile(...)
        .splitAsStream()
Stream.of(value1, ..., valueN)
StreamSupport.stream
    (iterable.spliterator(), ...)
...
```

```
interface Collection<E> {
  ...
  default Stream<E> stream() {
    return StreamSupport
      .stream(spliterator(), false);
  }

  default Stream<E> parallelStream() {
    return StreamSupport
      .stream(spliterator(), true);
  }

  ...
}
```

See the upcoming lesson on "*Java Streams Internals: Splitting & Combining*"

Operations that Create a Java Stream

- Many factory methods create streams

```
collection.stream()
collection.parallelStream()
Pattern.compile(...)
       .splitAsStream()
Stream.of(value1, ..., valueN)
StreamSupport.stream
   (iterable.spliterator(), ...)
...
```

```
Arrays.stream(array)
Arrays.stream(array, start, end)
Files.lines(file_path)
"string".chars()
Stream.iterate(init_value,
               generate_expression)
Stream.builder().add(...).build()
Stream.generate(supplier)
Files.list(file_path)
Files.find(file_path, max_depth,
           matcher)
...
```

There are also many other factory methods that create Java streams

See the upcoming lesson on "*Java Streams: Other Factory Methods*"

End of Understanding Java Streams Common Creation Operations