

Applying Java Method References

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 Lesson

- Understand how method (& constructor) references provide another foundational functional programming feature in Modern Java
- Know how to apply method references in practice

```
String[] nameArray = {"Barbara", "James", "Mary",  
                      "John", "Robert", "Michael",  
                      "Linda", "james", "mary"};
```

```
Stream.of(nameArray).forEach(System.out::print);
```

Applying Method References in Practice

Applying Method References in Practice

- Method references can be used to print a collection or an array

```
String[] nameArray = {"Barbara", "James", "Mary", "John",  
                      "Robert", "Michael", "Linda", "james", "mary"};
```

Array of names represented as strings

Applying Method References in Practice

- Method references can be used to print a collection or an array

```
String[] nameArray = {"Barbara", "James", "Mary", "John",  
                    "Robert", "Michael", "Linda", "james", "mary"};
```

- System.out.println() can be used to print out an array

```
System.out.println(List.of(nameArray));
```

prints

```
[Barbara, James, Mary, John, Linda, Michael, Linda, james, mary]
```

Applying Method References in Practice

- Method references can be used to print a collection or an array

```
String[] nameArray = {"Barbara", "James", "Mary", "John",  
                    "Robert", "Michael", "Linda", "james", "mary"};
```

- System.out.println() can be used to print out an array

```
System.out.println(List.of(nameArray));
```

prints

Factory method returns a fixed-size list backed by the array.

```
[Barbara, James, Mary, John, Linda, Michael, Linda, james, mary]
```

Applying Method References in Practice

- Method references can be used to print a collection or an array

```
String[] nameArray = {"Barbara", "James", "Mary", "John",  
                    "Robert", "Michael", "Linda", "james", "mary"};
```

- `System.out.println()` can be used to print out an array
- Java's `forEach()` methods can be used to print out values of an array

Applying Method References in Practice

- Method references can be used to print a collection or an array

```
String[] nameArray = {"Barbara", "James", "Mary", "John",  
                    "Robert", "Michael", "Linda", "james", "mary"};
```

- `System.out.println()` can be used to print out an array
- Java's `forEach()` methods can be used to print out values of an array, e.g.
- In conjunction with a stream & method reference

```
Stream.of(nameArray).forEach(ex7::printString);
```

prints

Factory method that creates a stream from an array

```
Barbara James Mary John Linda Michael Linda jame smary
```


Applying Method References in Practice

- Method references can be used to print a collection or an array

```
String[] nameArray = {"Barbara", "James", "Mary", "John",  
                    "Robert", "Michael", "Linda", "james", "mary"};
```

- `System.out.println()` can be used to print out an array
- Java's `forEach()` methods can be used to print out values of an array, e.g.
- In conjunction with a stream & method reference

```
Stream.of(nameArray).forEach(ex7::printString);
```

prints

```
Barbara James Mary John Linda Michael / Linda james mary
```

Performs method reference action on each stream element

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

Applying Method References in Practice

- Method references can be used to print a collection or an array

```
String[] nameArray = {"Barbara", "James", "Mary", "John",  
                    "Robert", "Michael", "Linda", "james", "mary"};
```

- `System.out.println()` can be used to print out an array
- Java's `forEach()` methods can be used to print out values of an array, e.g.
 - In conjunction with a stream & method reference
 - In conjunction with a collection (e.g., `List`)

```
List.of(nameArray).forEach(ex7::printString);
```

prints

Factory method converts an array into a list.

```
Barbara James Mary John Linda Michael Linda james mary
```

Applying Method References in Practice

- Method references can be used to print a collection or an array

```
String[] nameArray = {"Barbara", "James", "Mary", "John",  
                    "Robert", "Michael", "Linda", "james", "mary"};
```

- `System.out.println()` can be used to print out an array
- Java's `forEach()` methods can be used to print out values of an array, e.g.
 - In conjunction with a stream & method reference
 - In conjunction with a collection (e.g., List)

```
List.of(nameArray).forEach(ex7::printString);
```

prints

Performs method reference action on each list element

```
Barbara James Mary John Linda Michael Linda james mary
```

Applying Method References in Practice

- Method references can be used to print a collection or an array

```
String[] nameArray = {"Barbara", "James", "Mary", "John",  
                      "Robert", "Michael", "Linda", "james", "mary"};
```

- `System.out.println()` can be used to print out an array
- Java's `forEach()` methods can be used to print out values of an array, e.g.
 - In conjunction with a stream & method reference
 - In conjunction with a collection (e.g., `List`)
 - `forEach()` on a stream differs slightly from `forEach()` on a collection



Applying Method References in Practice

- Method references can be used to print a collection or an array

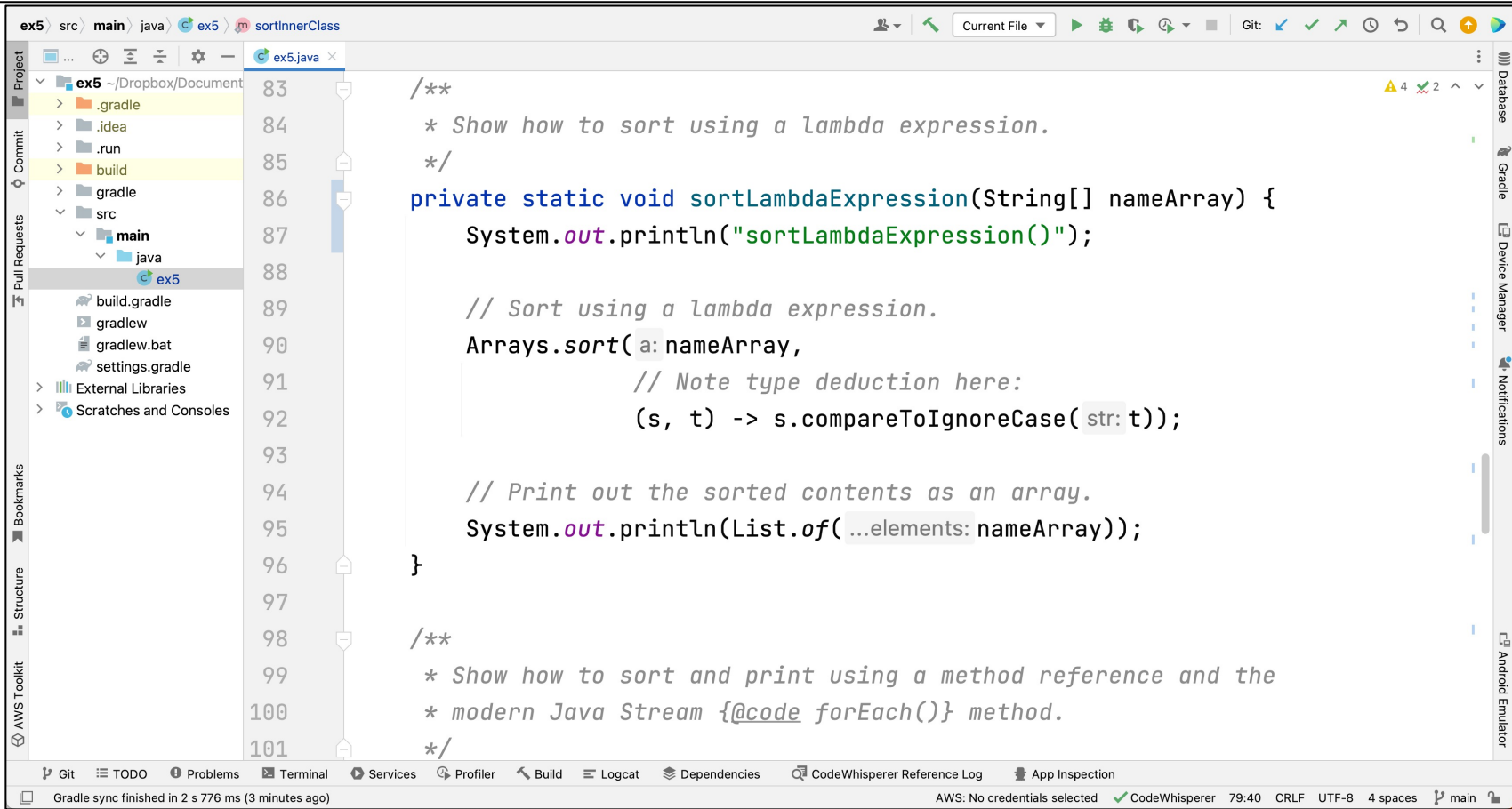
```
String[] nameArray = {"Barbara", "James", "Mary", "John",  
                    "Robert", "Michael", "Linda", "james", "mary"};
```

- `System.out.println()` can be used to print out an array
- Java's `forEach()` methods can be used to print out values of an array, e.g.
 - In conjunction with a stream & method reference
 - In conjunction with a collection (e.g., `List`)
- `forEach()` on a stream differs slightly from `forEach()` on a collection
 - e.g., `forEach()` ordering is undefined on a stream, whereas it's defined for a collection



Applying Java Method References in Case Study ex7

Applying Java Method References in Case Study ex7



The screenshot shows an IDE window with a project structure on the left and a code editor on the right. The code editor displays the following Java code:

```
83  /**
84   * Show how to sort using a lambda expression.
85   */
86  private static void sortLambdaExpression(String[] nameArray) {
87      System.out.println("sortLambdaExpression()");
88
89      // Sort using a lambda expression.
90      Arrays.sort(a: nameArray,
91                // Note type deduction here:
92                (s, t) -> s.compareToIgnoreCase(str: t));
93
94      // Print out the sorted contents as an array.
95      System.out.println(List.of(...elements: nameArray));
96  }
97
98  /**
99   * Show how to sort and print using a method reference and the
100  * modern Java Stream {@code forEach()} method.
101  */
```

The IDE interface includes a top toolbar with icons for file operations, a search bar, and a status bar at the bottom. The status bar shows "Gradle sync finished in 2 s 776 ms (3 minutes ago)" and "AWS: No credentials selected CodeWhisperer 79:40 CRLF UTF-8 4 spaces main".

See github.com/douglasraigschmidt/ModernJava/tree/main/FP/ex7

End of Overview of Java Method References