



Java Training

Jan - Feb 2023

Java 8

- Lambda Expressions
- Method Reference
- Functional Interface
- Optional
- Predicate
- Default Method
- Base64
- Stream
- String Joiner
- Collectors
- forEach
- Date/Time API
- Nashorn Script Engine
- I/O & Security Improvements

Lambda Expressions

- Helps us to **write** our code in **functional** style.
- Describe what you want, rather than how to get it.
- Provides a clear way to **implement Single Abstract Method** by using an expression
- It is very **useful** in collection library in which it helps to **iterate, filter and extract** data.

Lambda Expressions Syntax

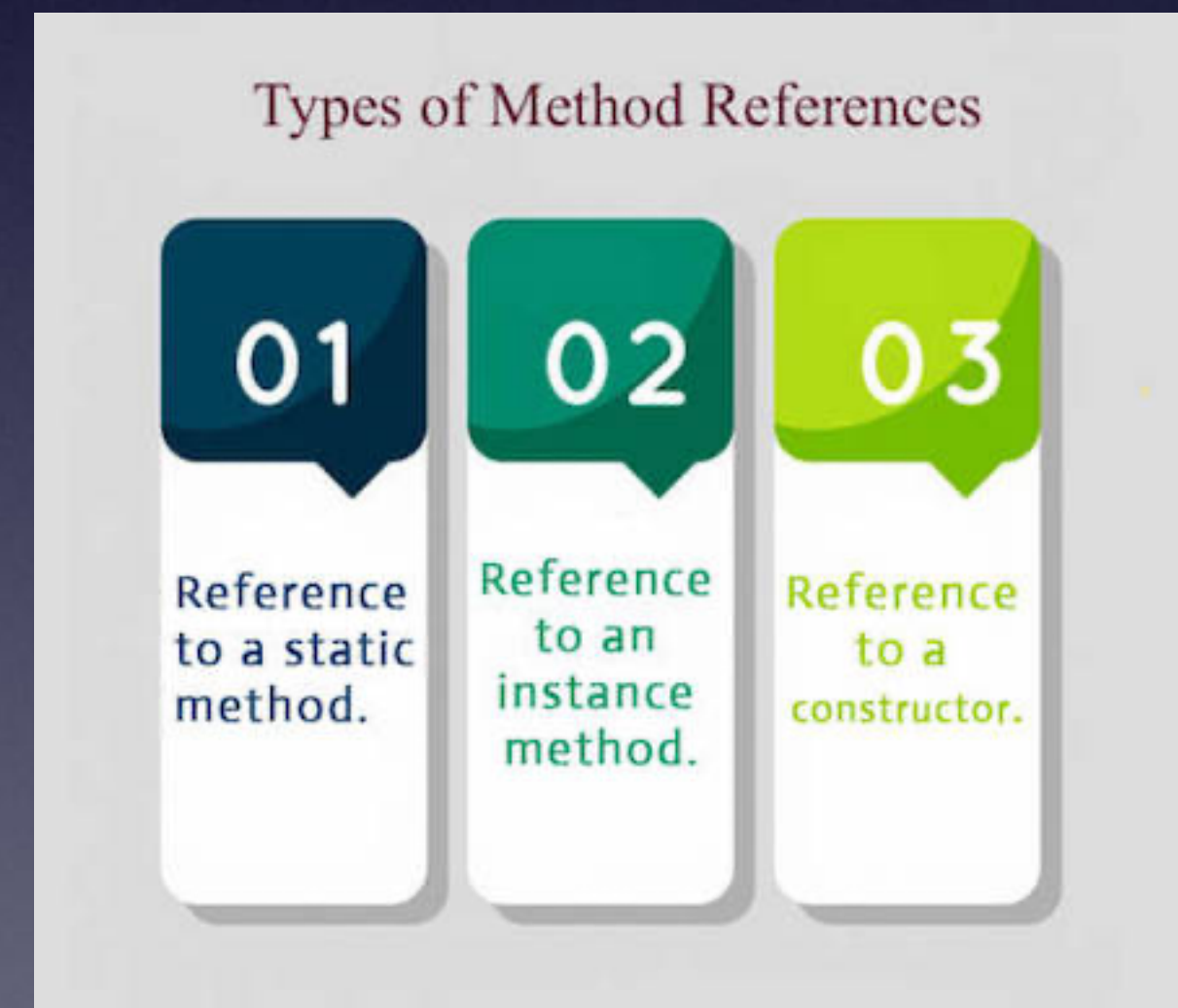
- Standard Syntax
- Parameter Type
- Multiple Lines of Code
- Single Parameter with Inferred Type
- Method References
- No Parameter

The diagram illustrates the syntax of a lambda expression with the example: `(int arg1, String arg2) -> {System.out.println("Two arguments "+arg1+" and "+arg2);}`. A horizontal line is drawn under the entire expression. Three vertical lines descend from this horizontal line to labels below: the first line is under the parameter list `(int arg1, String arg2)` and is labeled "Argument List"; the second line is under the arrow token `->` and is labeled "Arrow token"; the third line is under the body `{System.out.println("Two arguments "+arg1+" and "+arg2);}` and is labeled "Body of lambda expression".

Method Reference

- It is used to refer method of **functional interface** .
- It is **compact** and easy form of lambda expression.

- 01: ContainingClass::staticMethodName
- 02: containingObject::instanceMethodName
- 03: ClassName::**new**



Functional Interface

- An **Interface** that contains only **one** abstract **method**.
- It **can** have **any** number of **default** and static **methods**.
- It **can** also declare **methods** of **object class**.
- Also known as Single Abstract Method Interfaces.

Optional

- It is a `public final` class.
- Which is used to deal with `NullPointerException` in Java application.
- Provides methods to `check` the presence of `value`.

Predicate

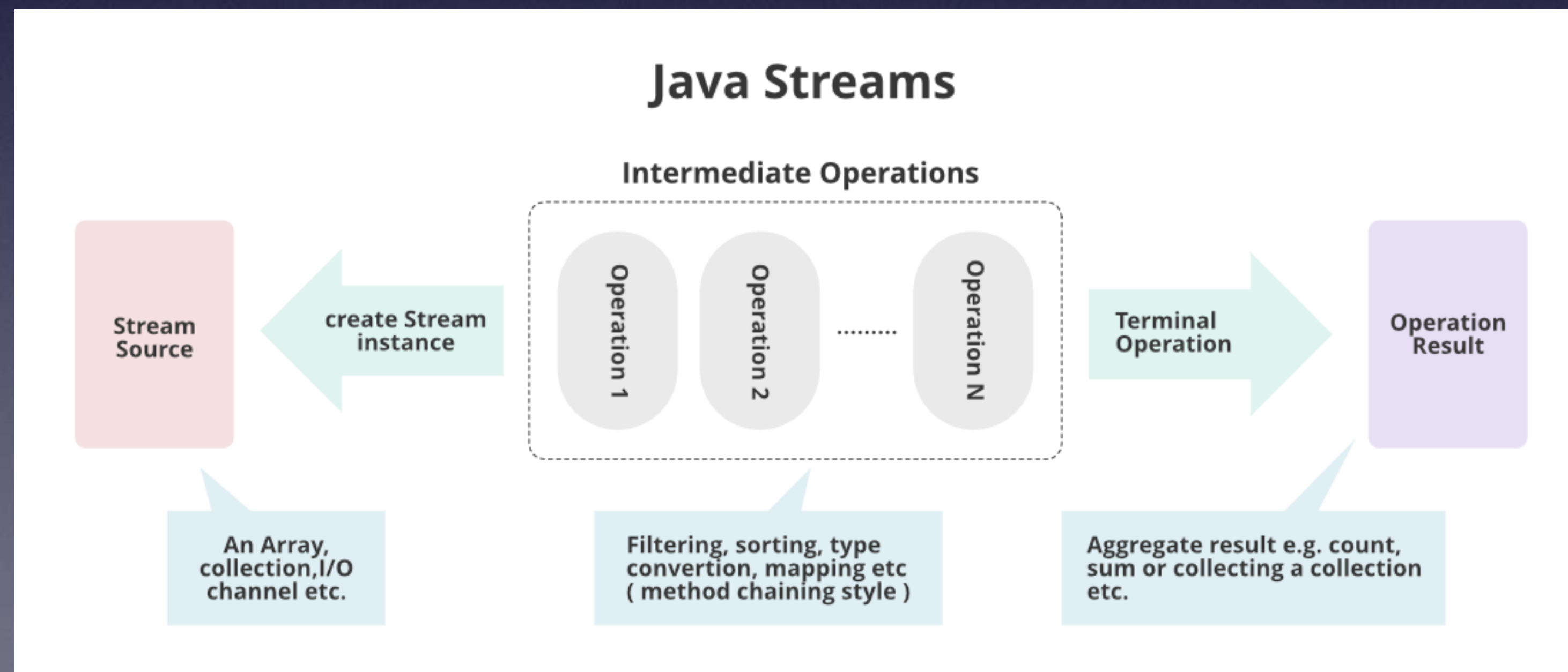
- defined in the `java.util.function` package.
- It improves manageability of code, helps in unit-testing them separately
- Contain some methods like:
 - `isEqual(Object targetRef)`
 - `and(Predicate other)`
 - `negate()`
 - `or(Predicate other)`
 - `test(T t)`

Default Methods

- Java provides a facility to create default **methods** inside the **interface**.
- Methods which are defined inside the interface and **tagged** with **default keyword** are known as default methods.
- These methods are **non-abstract** methods and can **have** method **body**.

Stream

- Java 8 java.util.stream package consists of classes, interfaces and an enum to allow functional-style operations on the elements.
- It performs lazy computation. So, it executes only when it requires



StringJoiner

- Used for **joining** Strings making use of a **delimiter**, **prefix**, & **suffix**.
- The **default** value is returned only when the StringJoiner is **empty**.
- **merge()** : It **adds** the contents of the given StringJoiner **without prefix and suffix** as the **next element**.
- **Collectors.joining()** **internally** uses StringJoiner to perform the joining operation.

Collectors

- Collectors is a final class that extends Object class.
- It provides reduction operations, such as accumulating elements into collections, summarizing elements according to various criteria etc.

forEach

- Java provides a new method `forEach()` to iterate the elements.
- It is defined in `Iterable` and `Stream` interfaces.
- It is a default method defined in the `Iterable` interface.
- `Collection` classes which extends `Iterable` interface can use `forEach()` method to iterate elements.
- This method takes a single parameter which is a functional interface.
- So, you can pass lambda expression as an argument.

Date/Time API

- Java has introduced a new Date and Time API since Java 8.
- The `java.time` package contains Java 8 Date and Time classes.

Drawbacks of existing Date/Time API's

- **Thread safety:** The existing classes such as Date and Calendar does not provide thread safety. Hence it leads to hard-to-debug concurrency issues that are needed to be taken care by developers. The new Date and Time APIs of Java 8 provide thread safety and are immutable, hence avoiding the concurrency issue from developers.
- **Bad API designing:** The classic Date and Calendar APIs does not provide methods to perform basic day-to-day functionalities. The Date and Time classes introduced in Java 8 are ISO-centric and provides number of different methods for performing operations regarding date, time, duration and periods.
- **Difficult time zone handling:** To handle the time-zone using classic Date and Calendar classes is difficult because the developers were supposed to write the logic for it. With the new APIs, the time-zone handling can be easily done with Local and ZonedDateTime APIs.

Why Default Method ?

- Before Java 8, interfaces could have only abstract methods.
- The implementation of these methods has to be provided in a separate class.
- So, if a new method is to be added in an interface, then its implementation code has to be provided in the class implementing the same interface.
- To overcome this issue, Java 8 has introduced the concept of default methods which allow the interfaces to have methods with implementation without affecting the classes that implement the interface.

Nashorn JavaScript Engine

- Nashorn is a JavaScript engine.
- It is used to execute JavaScript code dynamically at JVM.
- You can execute JavaScript code by two ways
 - Using jjs command-line tool, and
 - By embedding into Java source code.

Nashorn cont.

- The new default JavaScript engine for the JVM as of Java 8.
- Many sophisticated techniques have been used to make Nashorn orders of magnitude more performant than its predecessor called Rhino, so it is a worthwhile change.

Java Base64

- Basic Encoding and Decoding
 - A-Za-z0-9+/_
- URL and Filename Encoding and Decoding
 - A-Za-z0-9+_
• url and file name safe
- MIME
 - MIME friendly format
 - Represented in lines of no more than 76 characters each

Java Parallel Array Sorting

- The parallelSort() method has added to java.util.Arrays class
- It uses the JSR 166 Fork/Join parallelism common pool to provide sorting of arrays.
- It is an overloaded method.

Security Enhancements

- Java 8 : TLS 1.1, 1.2 (default) at client side
- Advanced Encryption Standard (AES) and Password-Based Encryption (PBE) algorithms
 - PBESWithSHA256AndAES_128
 - PBESWithSHA512AndAES_256

-

Java IO Improvements

- `Files.list (Path dir)`: Returns a lazily filled stream, each element of which represents a directory entry.
- `Files.lines (Path path)`: Reads all the lines from a stream.
- `Files.find ()`: Returns a stream filled by a path after searching for files in the file tree rooted at a provided beginning file and many more.
- `BufferedReader.lines ()`: Returns a stream containing all of the elements of `BufferedReader`'s lines and much more
- `Buffered`

Java 9

- Modular System
- A New HTTP Client
- Process API
- Try-With-Resources
- Diamond Operator Extension
- Interface Private Method
- JShell Command Line Tool
- JCMD Sub-Commands
- Multi-Resolution Image API
- Variable Handles
- Publish-Subscribe Framework
- Unified JVM Logging
- Immutable Set
- Optional to Stream

Modular System

- One of the big changes or java 9 feature is the Module System.
- Before Java SE 9 versions, we are using Monolithic Jars to develop Java-Based applications.
 - This architecture has lot of limitations and drawbacks. To avoid all these shortcomings, Java SE 9 comes with the Module System.
- Oracle Corp. introduced the following features as part of Jigsaw Project:
 - Modular JDK
 - Modular Java Source Code
 - Modular Run-time Images
 - Encapsulate Java Internal APIs
 - Java Platform Module System

