

Lambdas and Streams

Audience Course Lambdas and Streams

The course Lambdas and Streams is intended for Java developers who want to learn how to do functional programming with the lambdas and stream libraries that were added to Java 8.

Prerequisites Course Lambdas and Streams

Good knowledge and experience of programming in Java is required, but it is not necessary to have in-depth knowledge of Java 8.

Realization Training Lambdas and Streams

The theory is treated on the basis of presentations and is interspersed with exercises. Demos are used to clarify the theory. Class times are from 9:30 up and to 16.30.

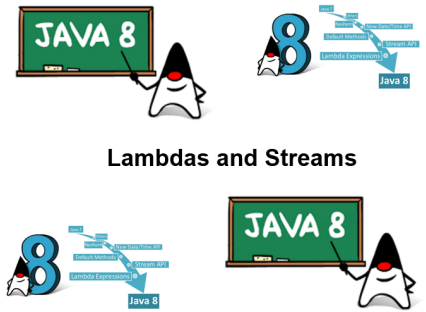
Certificate Lambdas and Streams

Participants will receive a certificate Lambdas and Streams after successful completion of the course.

Duration: 2 days

Price: € 1499

[Open Schedule](#)



Lambdas and Streams

Content Course Lambdas and Streams

In the course Lambdas and Streams participants learn functional programming with the Lambdas and Stream libraries added in Java 8. The functional programming constructs are complementary to object-oriented programming in Java and the two approaches are not mutually exclusive.

Java 8 Review

The course Lambdas and Streams begins with a discussion of how functionality was passed before Java 8. Inner and anonymous classes are covered in this respect. The new default methods of Java 8 interfaces are also treated.

Lambdas Intro

Subsequently Lambdas are introduced. They correspond to interfaces with only one abstract method and are known as functional interfaces.

Method References

Lambdas can be replaced by method references resulting in a more compact notation. The different type of method references are discussed subsequently.

Function Package

The program of the course Lambdas and Stream also includes a discussion of the `java.util.function` package. The many ready-made functional interfaces from that package are covered with their respective parameters, return values and default methods.

Higher Order Functions

Higher order functions to whom functions are passed as parameters or by whom functions are returned as return value, are also treated.

Streams Intro

Next it is time for a discussion of the stream library. It is explained how streams can be created from arrays, lists and other data structures. The streams do not store data, but transform and filter data in a series of sequential operations. The difference between intermediate and terminal stream operations is covered then.

Parallel Streams

Finally attention is paid to parallel streams, which improve performance because they can be executed on multiple CPU cores at the same time. Grouping with streams is discussed as well.

Modules Course Lambdas and Streams

Module 1 : Java 8 Review	Module 2 : Lambdas Intro	Module 3 : Method References
Java 8 Lambdas and Streams Installation and Setup Online References Review of Basic Handlers Anonymous Classes Separate Classes Main Implements Interface Named Inner Classes Anonymous Inner Classes Generic Classes and Methods Common Eclipse Techniques	What are Lambdas? Passing Functions Around Lambdas Interpretation Underlying Advantages Most Basic Form Type Inference Expression for Body Omitting Parens Lambda Alternatives Numerical Integration Timing Utilities	Review @Override @FunctionalInterface Updated Interfaces Method References Type of Method References Constructor References Variable Scoping Lambda Scoping Rules Final Local Variables Button Listeners Concurrent Image Download
Module 4 : Function Package	Module 5 : Higher Order Functions	Module 6 : Streams Intro
Interfaces in java.util.function Lambda Targets IntConsumer DoublePredicate Predicate Interface Refactoring BinaryOperator Interface Consumer Interface Consumer Test Supplier Interface Supplier Usage	Returning Lambdas From Predicate and, or negate, isEqual From Function andThen compose, identity From Consumer andThen Custom Methods Typing Issues	Building Streams Characteristics of Streams Method Types Primitive Streams Converting Streams forEach, map and filter findFirst and findAny toArray and collect Optional Class Lazy evaluation Short Circuit Operations
Module 7 : Stream Operations	Module 8 : Parallel Streams	
limit and skip sorted and distinct noneMatch and allMatch anyMatch and count IntStream DoubleStream LongStream Reduction Operations reduce min and max sum and average	Traditional Loops Stream Approach Parallel versus Concurrent Fork and Join Parallel Reduction No Global Data Associative Operation Performance Comparison Infinite Streams generate, iterate and collect Grouping	