

Lambdas en Streams

Doelgroep Cursus Lambdas and Streams

De cursus Lambdas and Streams is bedoeld voor Java ontwikkelaars die functioneel willen leren programmeren met de aan Java 8 toegevoegde lambdas en stream libraries.

Voorkennis Cursus Lambdas and Streams

Goede kennis van en ervaring met programmeren in Java is vereist, maar het is niet noodzakelijk om diepgaande kennis van Java 8 te hebben.

Uitvoering Training Lambdas and Streams

De theorie wordt behandeld aan de hand van presentaties en wordt afgewisseld met oefeningen. Demo's worden gebruikt om de theorie te verduidelijken. De lestijden zijn van 9.30 uur tot 16.30 uur.

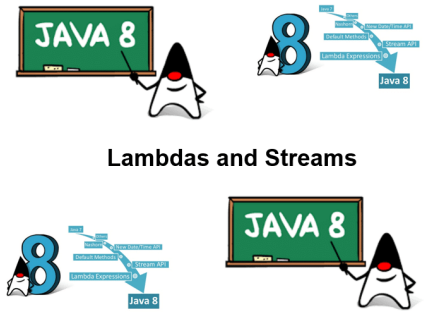
Certificaat Lambdas and Streams

Deelnemers ontvangen na succesvolle afronding van de cursus een certificaat Lambdas and Streams.

Duur: 2 dagen

Prijs: € 1499

[Open Rooster](#)



Inhoud Cursus Lambdas en Streams

In de cursus Lambdas en Streams leren de deelnemers functioneel programmeren met de in Java 8 toegevoegde Lambdas en Stream libraries. De functionele programmeer constructies zijn een aanvulling op object georiënteerd programmeren in Java en de twee benaderingen sluiten elkaar niet uit.

Java 8 Review

De cursus Lambdas en Streams gaat van start met een bespreking van hoe functionaliteit voor Java 8 werd doorgegeven. Hierbij komen inner en anonymous classes ter sprake. Ook wordt ingegaan op de nieuwe default methods van Java 8 interfaces.

Lambdas Intro

Vervolgens worden Lambdas geïntroduceerd die corresponderen met interfaces met maar één abstracte method. Deze interfaces staan bekend als functional interfaces.

Method References

Lambdas kunnen vervangen worden door method references hetgeen een compactere notatie to gevolg heeft. De verschillende type method references worden dan besproken.

Function Package

Op het programma van de cursus Lambdas en Stream staat ook de bespreking van het java.util.function package. Aan de orde komen de vele kant en klare functional interfaces uit die package met hun respectievelijke parameters, return values en default methods.

Higher Order Functions

Ook worden higher order functions besproken aan wie functies als parameter worden doorgegeven dan wel door wie functies als return value terug worden gegeven.

Streams Intro

Vervolgens is het tijd voor een bespreking van de stream library. Uitgelegd wordt hoe streams kunnen worden gecreëerd uit arrays, lists en andere data structuren. De streams slaan geen data op maar transformeren en filteren data in een reeks van achtereenvolgende operaties. Het verschil tussen intermediate en terminal stream operations komt aan de orde.

Parallel Streams

Tenslotte wordt aandacht besteed aan parallel streams waardoor een verbeterde performance kan worden bereikt doordat ze uitgevoerd kunnen worden op meerdere CPU cores tegelijkertijd. Ook komt dan grouping met streams aan de orde.

Modules Cursus Lambdas en 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 Paresn 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	