

Kotlin Programmeren

Doelgroep Cursus Kotlin Programmeren

De cursus **Kotlin** Programmeren is bedoeld voor developers die willen leren hoe Kotlin zich onderscheidt van Java en willen leren programmeren in Kotlin.

Voorkennis training Kotlin Programmeren

Om aan deze cursus deel te nemen is ervaring met programmeren vereist. Kennis van programmeren in Java is bevorderlijk voor de begripsvorming.

Uitvoering cursus Kotlin Programmeren

De theorie wordt behandeld aan de hand van presentaties. Illustratieve demo's worden gebruikt om de behandelde concepten te verduidelijken. Er is voldoende gelegenheid om te oefenen en afwisseling van theorie en praktijk. De cursustijden zijn van 9.30 tot 16.30.

Certificering cursus Kotlin Programmeren

De deelnemers krijgen na het goed doorlopen van de cursus een officieel certificaat Kotlin Programmeren.

Duur: 3 dagen

Prijs: € 1999

[Open Rooster](#)

Inhoud Cursus Kotlin Programmeren

In de cursus Kotlin Programmeren leren de deelnemers de object georiënteerde en functionele taal Kotlin te gebruiken voor software ontwikkeling. Kotlin is een moderne statisch getypeerde taal die volledig compatibel is met Java en draait op de Java Virtual Machine (JVM). Kotlin heeft moderne taal kenmerken zoals type inferences, null safety, coroutines en unchecked exception die in Java inbreken. Bovendien is Kotlin code is vergelijking met **Java** veel compacter.

Kotlin Intro

De cursus Kotlin Programmeren gaat van start met een overzicht van een aantal belangrijke kenmerken van Kotlin. Aandacht wordt besteed aan de syntax vereenvoudigingen in de Kotlin taal ten opzichte van Java. Ook wordt besproken dat Kotlin code niet alleen naar Java kan worden gecompileerd, maar ook naar JavaScript of native code.

Language Syntax

Vervolgens wordt ingegaan op de taal syntax met type inference, mutable en immutable variabele declaraties, if en when expressions, ranges, loops en iterators.

Classes en Objects

Wat betreft object georiënteerd programmeren komen class initialization, primary en secondary constructors final en open classes, abstract classes en interfaces aan de orde. En ook wordt ingegaan op data classes waarbij methodes als equals, toString en hashCode automatisch worden gegenereerd.

Functies

Kotlin ondersteunt ook functioneel programmeren en onderdeel van het programma van de cursus zijn dan ook lambda functies, higher order function, het doorgeven van functies als parameter en het retourneren van functies. Ook wordt aandacht besteed aan extension methods, destructuring declarations, nested functies en het uitpakken van parameters met de spread operator.

Collections en Generics

Eveneens wordt het Collection Framework in Kotlin behandeld dat ondersteuning biedt voor mutable en immutable collection en sequences met lazy evaluation. Ook geparametriseerde types met generics passeren dan de revue. En er wordt ingegaan op delegation in Kotlin met lazy en observable properties.

Coroutines

Tenslotte wordt aandacht besteed aan het gebruik van coroutines in Kotlin die beschouwd kunnen worden als licht gewicht threads en die uitstekend geschikt zijn voor asynchrone afhandeling.

Modules Cursus Kotlin Programmeren

Module 1 : Kotlin Intro	Module 2 : Language Syntax	Module 3 : Classes and Objects
What is Kotlin? Variables Type Inference Kotlin Characteristics Null Handling Safe Call Operator Properties Custom Accessors Kotlin Exceptions Kotlin versus Java Interoperability Run as ECMAScript Potential Downsides	Packages and Imports Default Imports Basic Types Boxing Explicit Conversions Characters Arrays If and When Expressions Loops and Iterators Ranges Jumps and Labels Elvis Operator !! Operator	Kotlin Class Initialization Property Settings Inheritance Calling Base Constructors Secondary Constructors Visibility Modifiers Abstract Classes and Interfaces Nested and Inner Classes Data Classes Destructuring Declarations Sealed Classes Kotlin Objects Companion Objects
Module 4 : Functions	Module 5 : Collections	Module 6 : Generics
Function Scope Local Functions Extension Functions Static Resolvment Extension Properties Recursive Functions Kotlin Tail Recursion Higher Order Functions Lambda Expressions Closures Infix Functions Operator Functions Scope Functions	Collection Types Immutable Collections Mutable Collections Collection Hierarchy Iterators Ranges and Progressions Sequences Common Operations Write Operations Transformations Filtering Plus and Minus Grouping	Generic Classes Generic Functions Type Inference PECS Principle out Keyword in Keyword Type Projections Subtype to Supertype Variances Covariance Contravariance Star Projections Generic Constraints
Module 7 : Delegation	Module 8 : Interoperability	Module 9 : Coroutines
Delegation Design Pattern by Keyword Inheritance Alternative Delegated Properties Lazy Properties Property as Input Return Type as Lazy Observable Properties Standard Delegates Properties in Map Local Delegated Properties Delegate Requirements Translation Rules	Calling Java from Kotlin Calling Static Methods Using Java Collection Reserved Words in Kotlin Calling Kotlin in Java Calling Kotlin Functions Calling Extension Functions Using Mutable Collections Immutable Collections Files with JVM Annotation Functions with JVM Annotation Calling Kotlin Class Calling Kotlin Singleton	What are Coroutines? Concurrency Pattern Light-weight Threads Coroutine Scope launch Method Blocking versus non-Blocking Structured Concurrency Scope Builder Global Coroutines Cancelation and Timeout Coroutine Context Channels Asynchronous Flow