

# Opleiding : Kotlin Multiplatform, multiplatformapplicaties ontwikkelen

*Praktijkcursus - 3d - 21u00 - Ref. LKD*

*Prijs : 1650 € V.B.*

NEW

Na afloop van de cursus zijn de deelnemers in staat om de ontwikkelings- en onderhoudskosten aanzienlijk te verlagen, de time-to-market van hun applicaties te versnellen en effectief te communiceren met native teams bij het integreren van KMP-modules in bestaande projecten. Dit trainingsprogramma is gericht op werknemers in professionele sectoren die onder de OPCO Atlas vallen.

## Pedagogische doelstellingen

Aan het einde van de training is de deelnemer in staat om:

- ✓ De concepten van Kotlin Multiplatform begrijpen
- ✓ Een ontwikkelomgeving opzetten
- ✓ De architectuur van een multiplatform mobiele applicatie ontwerpen
- ✓ Een vloeiende, krachtige gebruikersinterface bouwen
- ✓ Testen uitvoeren om de kwaliteit en betrouwbaarheid van applicaties te garanderen

## Doelgroep

Voor OPCO Atlas leden: ontwikkelaars, architecten.

## Voorafgaande vereisten

Kennis van programmeertalen (Java, C#, C++, JavaScript, Python, enz.).

## Praktische modaliteiten

### Leer methodes

Om het leerproces te optimaliseren, kunnen op verzoek van de deelnemer e-learningmodules worden aangeboden voor en na de klassikale sessie of virtuele les.

### DEELNEMERS

Voor OPCO Atlas leden:  
ontwikkelaars, architecten.

### VOORAFGAANDE VEREISTEN

Kennis van programmeertalen (Java, C#, C++, JavaScript, Python, enz.).

### VAARDIGHEDEN VAN DE CURSUSLEIDER

De deskundigen die de cursus leiden zijn specialisten op het betreffende vakgebied. Zij werden geselecteerd door onze pedagogische teams zowel om hun vakkennis als hun pedagogische vaardigheden voor elke cursus die zij geven. Zij hebben minstens vijf tot tien jaar ervaring in hun vakgebied en oefenen of oefenden verantwoordelijke bedrijfsfuncties uit.

### BEOORDELINGSMODALITEITEN

De cursusleider beoordeelt de pedagogische vooruitgang van de deelnemer gedurende de gehele cursus aan de hand van meerkeuzevragen, praktijksituaties, praktische opdrachten, ...  
De deelnemer legt ook van tevoren en naderhand een test af ter bevestiging van de verworven kennis.

# Opleidingsprogramma

## 1 Algoritmen, redeneren vóór ontwerpen - Vooropleiding digitale

### leerinhoud

- Inleiding tot algoritmen.
- Basisinstructies in pseudocode.

### Digitale activiteiten

Deze e-learningcursus leert je na te denken voordat je een programma ontwerpt door je kennis te laten maken met de basisprincipes van algoritmiek. Deelnemers bestuderen met name de fundamentele instructies in pseudocode.

## 2 Inleiding tot Kotlin Multiplatform

- Traditionele multiplatformontwikkelingsproblemen.
- Evolutie van KMM naar KMP: geschiedenis en JetBrains visie.
- Vergelijking met andere oplossingen (React Native, Flutter, Xamarin).
- Use cases en succesverhalen (Netflix, VMware, Philips).

### Praktisch werk

Groepsbrainstormen : Wat zijn de huidige uitdagingen in mobiele ontwikkeling? Vergelijkende analyse van multiplatformoplossingen in subgroepen.

## 3 Architectuur en fundamentele concepten

- Gelaagde architectuur: gemeenschappelijk, Android, iOS, web.
- Verwacht/werkelijk mechanisme: declaraties en implementaties.
- Strategieën voor het delen van codes: wat wel en wat niet delen.

### Praktisch werk

Creatie van de eerste eenvoudige expect/actual statements.

## 4 Configuratie van de omgeving

- Systeemvereisten per platform (Windows, macOS, Linux).
- Tools installeren: Android Studio, Xcode, KMP-plugins.
- Gradle-configuratie en projectstructuur.

### Praktisch werk

Begeleide installatie op werkstations met collectieve verificatie.

## 5 Eerste KMP-project

- Een KMP project maken met de IntelliJ wizard.
- Mapstructuur: commonMain, androidMain, iosMain.
- Eerste gedeelde klasse en zijn specifieke implementaties.
- Bouwen en uitvoeren op de verschillende platforms.

### Praktisch werk

Ontwikkeling in tweetallen: maken van een "Hallo KMP" app met weergave van het huidige platform. Presentatie van projecten en feedback.

## PEDAGOGISCHE EN TECHNISCHE MIDDELEN

- De gebruikte pedagogische middelen en cursusmethoden zijn voornamelijk: audiovisuele hulpmiddelen, documentatie en cursusmateriaal, praktische oefeningen en correcties van de oefeningen voor praktijkstages, casestudies of reële voorbeelden voor de seminars.
- Na afloop van de stages of seminars verstrekt ORSYS de deelnemers een evaluatievragenlijst over de cursus die vervolgens door onze pedagogische teams wordt geanalyseerd.
- Na afloop van de cursus wordt een presentielijst per halve dag verstrekt, evenals een verklaring van de afronding van de cursus indien de stagiair alle sessies heeft bijgewoond.

## TOEGANGSMODALITEITEN EN TERMIJNEN

De inschrijving dient 24 uur voor aanvang van de cursus plaatsgevonden te hebben.

## TOEGANKELIJKHEID VOOR MINDERVALIDEN

Is voor u speciale toegankelijkheid vereist? Neem contact op met mevr. FOSSE, contactpersoon voor mindervaliden, via het adres psh-accueil@ORSYS.fr om uw verzoek en de haalbaarheid daarvan zo goed mogelijk te bestuderen.

## 6 Beheer van afhankelijkheden

- Gemeenschappelijke versus specifieke afhankelijkheden.
- ecosysteem van KMP-bibliotheek: kotlinox, Ktor, SQLDelight.
- Geavanceerde Gradle-configuratie voor sourceSets.

### Praktisch werk

Algemene afhankelijkheden toevoegen en testen (kotlinox-coroutines, kotlinox-serialisatie).

## 7 Patronen en best practices

- Organisatie van pakketten en modules.
- Aanbevolen patronen: Repository, UseCase, ViewModel.
- Platformoverkoepelende foutafhandeling met verzegelde klassen.

### Praktisch werk

Refactoring van het eerder gemaakte project volgens de patronen die in de subgroepen zijn gepresenteerd

## 8 MVVM-architectuur en Repository-ontwerppatroon

- Schone architectuur voor gebruik op meerdere platforms.
- Implementatie van het Repository patroon met gemeenschappelijke interfaces.
- ViewModels gedeeld met StateFlow en coroutines.
- Afhankelijkheidsinjectie met Koin.

### Praktisch werk

Ontwikkeling van een gebruikersmanager met een compleet Repository Design Pattern.

## 9 Gegevensbeheer en API's

- HTTP-client met Ktor: configuratie en gebruik.
- JSON serialisatie met kotlinox.serialization.
- Lokale opslag met SQLDelight: installatie en query's.
- Cache- en synchronisatiestrategieën.

### Praktisch werk

Creatie van een REST API-client voor het ophalen van weergegevens. - Implementatie van lokale caching met SQLDelight.

## 10 Statusbeheer en reactiesnelheid

- StateFlow en SharedFlow: concepten en verschillen.
- UI-statusbeheer: Laden, Succes, Fout.
- Combinatie van stroming en ritssluiting.

### Praktisch werk

Implementatie van een algemene statusmanager voor de weerapplicatie.

## 11 Gebruikersinterfaces met Compose Multiplatform

- Inleiding tot Compose Multiplatform: basisconcepten.
- UI-componenten: Tekst, Knop, LazyColumn, Kaart.
- Thema's en Material Design-beheer.
- Navigeer tussen schermen met Compose Navigation.
- Reactiviteit met collectAsState.

### Praktisch werk

De interface van de weerapplicatie gemaakt met Compose. Navigatie tussen lijst- en detailschermen geïmplementeerd.

## 12 Native integratie

- Android-integratie: de gedeelde module gebruiken.
- iOS-integratie: Swift/Kotlin framework en bridge.
- Gegevensoverdracht tussen native en gedeelde lagen.

### Praktisch werk

Integratie van KMP ViewModels in Android-activiteiten en iOS ViewControllers.

## 13 Beheer van specifieke platforms

- Specifieke API's: geolocatie, camera, meldingen.
- Abstractie strategieën voor systeem-API's.
- Prestaties en optimalisatie per platform.

### Praktisch werk

Geolocatie toegevoegd aan de weer-app, met specifieke implementaties.

## 14 KMP teststrategie

- Gedeelde eenheidstests : CommonTest en specifiek.
- Mocken met MockK in een multiplatformcontext.
- ViewModels en Repositories testen.
- Integratietests met databases en API's.

### Praktisch werk

Het schrijven van unit tests voor de componenten die de afgelopen dagen zijn ontwikkeld.

## 15 Gebruikersinterface testen

- Tests samenstellen met ComposeTestRule.
- Gebruikersinteractietests en toestandsverificatie.
- Navigatie- en gebruikersflowtests.
- Teststrategieën voor eigen onderdelen.

### Praktisch werk

UI-tests maken voor de schermen van de weerapplicatie.

## 16 Uitrol en CI/CD

- GitHub Acties configuratie voor KMP.
- Geautomatiseerde build voor Android (APK/AAB) en iOS (IPA).
- Implementatie in winkels: Play Store, App Store.
- Beheer van certificaten en handtekeningen.

### Praktisch werk

Een basis CI/CD pijplijn configureren met GitHub Actions.

## 17 Eindproject - E-commercetoepassing

- Productcatalogus met zoekfunctie en filters.
- Persistent winkelmandje.
- Gebruikersauthenticatie.
- Responsieve interface en vloeiende navigatie.
- Unit- en integratietesten.

### Praktisch werk

Architecture et planification du projet. Développement des fonctionnalités core. Tests et finalisations.

## 18 Perspectieven en "verder gaan"

- KMP-ecosysteem: nieuwe bibliotheken en ontwikkelingen.
- Gemeenschap en bronnen: documentatie, forums, conferenties.
- Bedrijfsadoptiestrategieën: geleidelijke migratie, teamtraining.
- KMP Stappenplan: Multiplatform web, desktop, wasm samenstellen.

### Workshop storytelling

Ontwikkeling van een persoonlijk actieplan voor elke deelnemer.

## 19 Kotlin, de essentiële basis - Post-training digitale leerinhoud

- Kennismaking met Kotlin.
- De grondbeginselen van Kotlin.
- De functies.
- Coroutines.
- Objectgeoriënteerd programmeren.
- Android-ontwikkeling.
- Moederlijke ontwikkeling.
- JavaScript-ontwikkeling.
- Server-side ontwikkeling.

### Digitale activiteiten

Deze online cursus behandelt de Kotlin-taal, de grondbeginselen, de verschillende functies, het begrip coroutine, objectgeoriënteerd programmeren met een demonstratie van de ontwikkeling van een toepassing in Android Studio, hoe Kotlin te gebruiken voor native of server-side ontwikkeling en hoe Kotlin JavaScript-code kan genereren.

## Data en plaats

### KLAS OP AFSTAND

2026 : 10 maa., 19 mei, 6 okt., 8 dec.

### PARIS LA DÉFENSE

2026 : 5 mei, 29 sep., 1 dec.

### LILLE

2026 : 10 maa., 19 mei, 6 okt., 8 dec.