



Kotlin Avancée

Durée 2 jour(s) (KOTLIN-ADVANCED)

Pour aller plus loin avec Kotlin

Description

Connaitre la syntaxe d'un langage est un point de départ pour commencer, cependant pour aller plus loin dans la maîtrise de ce langage, et l'utiliser efficacement au quotidien il faut aller plus en profondeur dans les spécificités fournies, mais aussi connaître son éco-système.

Objectifs

L'objectif est d'approfondir sa connaissance du langage Kotlin et de son éco-système.

Mettre en oeuvre les connaissances dans des travaux pratiques proches de notre activité quotidienne.

Dans cette formation nous présenterons les fonctions, la gestion des erreurs ainsi que la gestion de la concurrence avec Kotlin.

Nous aborderons aussi les pratiques de tests.

Enfin, nous illustrerons nos nouvelles connaissances, en faisant évoluer une application Java/Springboot vers du Kotlin.

Public

Personnes ayant les bases du langages et désirant aller plus loin

Prérequis

Connaissance de Kotlin et/ou Java/SpringBoot

Répartition

50% Théorie, 50% Pratique

Evaluations des acquis

L'évaluation des acquis de la formation se fera en séance au travers d'ateliers, d'exercices et/ou de travaux pratiques. Dans le cas d'une formation officielle éditeur, veuillez nous consulter afin que nous vous fassions part des modalités d'évaluation.

A l'issue de la formation, vous sera transmis une évaluation à chaud de l'action de formation qui vous permettra de nous faire part de vos retours quant à votre expérience apprenant avec Zenika.

Ressources pédagogiques

Les ressources pédagogiques proviennent de productions des équipes Zenika et/ou de la documentation éditeur dans le cas d'une formation "Officielle". Les documents sont en français ou en anglais.

RQTH et ma formation Zenika

Si vous êtes sujet à un handicap, prenez contact avec nos équipes pour que nous puissions définir ensemble comment nous pourrions aménager la session afin que vous puissiez vivre une expérience en formation inchangée.

Programme

Concept reminder

- Inheritance
- Variable visibility
- Interface and abstract classes
- Data Class, Enum Class, Sealed Class
- Nested Class / Inner Class
- Companion Object
- Null safety
- Annotation
- Reflection
- Type safe builder
- Dynamic type
- Covariance/contravariance

Kotlin functions

- High order function
- Inline functions
- Lambdas / Invocation
- Extension function

Exceptions handling

- Exceptions
- Préconditions
- Différences avec Java

Concurrency

- Threads, Blocking, CPU-bound versus I/O-bound
- Deadlocks & livelocks
- Executors
- Race conditions : Monitors, Locks , concurrent collections, Atomic variables, CountdownLatch, Cyclic Barrier, non-blocking I/O & asynchronous programming

Build / Packaging

- Kotlin-jvm
- Kotlin-native
- Kotlin-js

Tests

- Mockito Kotlin / MockK
- Expecting exceptions
- Matchers
- Test Data
- Inspectors
- Test configurations
- Property testing

Module Spring boot (2h)

- Gestion des properties
- Configuration des beans
- Spring data/jpa
- Spring rest
- Tests d'intégration spring boot