



Conception et langages

Kotlin - Mise en oeuvre

3 jours (21h00) | ★★★★★ 5/5 | KOTL-FND | Évaluation qualitative de fin de stage |
Formation délivrée en présentiel ou distanciel

Formations Informatique > Langages et développement > Conception et langages

Contenu mis à jour le 18/10/2024. Document téléchargé le 08/12/2024.

Objectifs de formation

A l'issue de cette formation, vous serez capable de :

- Mémoriser les bases du langage
- Reconnaître les différentes fonctionnalités de Kotlin
- Découvrir la librairie standard
- Prendre en main les différents outils disponibles.

Modalités, méthodes et moyens pédagogiques

Formation délivrée en présentiel ou distanciel* (blended-learning, e-learning, classe virtuelle, présentiel à distance).

Le formateur alterne entre méthode** démonstrative, interrogative et active (via des travaux pratiques et/ou des mises en situation).

Variables suivant les formations, les moyens pédagogiques mis en oeuvre sont :

- Ordinateurs Mac ou PC (sauf pour certains cours de l'offre Management), connexion internet fibre, tableau blanc ou paperboard, vidéoprojecteur ou écran tactile interactif (pour le distanciel)
- Environnements de formation installés sur les postes de travail ou en ligne
- Supports de cours et exercices

En cas de formation intra sur site externe à M2i, le client s'assure et s'engage également à avoir toutes les ressources matérielles pédagogiques nécessaires (équipements informatiques...) au bon déroulement de l'action de formation visée conformément aux prérequis indiqués dans le programme de formation communiqué.

* nous consulter pour la faisabilité en distanciel

** ratio variable selon le cours suivi

Prérequis

Notions en programmation orientée objet et connaissances de Java, fortement conseillées.

Public concerné

Développeurs et chefs de projets.

Cette formation :

- Est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- Bénéficie d'un suivi de son exécution par une feuille de présence élargée par demi-journée par les stagiaires et le formateur.

Programme

Jour 1

Introduction

- Pourquoi le Kotlin ?
- Introduction à la JVM (Java Virtual Machine)
- Installation des outils REPL de Kotlin (Read Eval Print Loop)
- La structure d'une application Kotlin
- Kotlin et IntelliJ IDEA
- Les conventions utilisées avec Kotlin

Bases de Kotlin

- Déclaration de variables en Kotlin
- Utilisation de variables "Basic Types" en Kotlin
- Boucles et ranges en Kotlin
- Structures conditionnelles If et When
- Collections en Kotlin
- Packages et imports en Kotlin

Exemple de travaux pratiques (à titre indicatif)

- *Ecriture de divers algorithmes reprenant les principales syntaxes de Kotlin afin de concevoir une application simple*

Les fonctions - Partie 1

- Fonctions en Kotlin
- Paramètres des fonctions en Kotlin
- Fonctions Infix en Kotlin
- Fonctions Anonyme en Kotlin
- Returns et Local Returns en Kotlin
- Tail recursion en Kotlin
- Bonnes et mauvaises pratiques

Exemples de travaux pratiques (à titre indicatif)

- *A partir des projets préalablement générés, ajout et appels de plusieurs fonctions Kotlin*

Classes en Kotlin

- Une classe
- Les attributs
- Méthodes (Fonctions Members)

- Visibilité des membres en Kotlin
- Héritage en Kotlin
- Abstract Classes en Kotlin
- Interface en Kotlin
- Polymorphisme en Kotlin
- Data Classes en Kotlin
- Enum Classes en Kotlin
- Nested Classes en Kotlin
- Sealed Classes en Kotlin
- Bonnes et mauvaises pratiques

Exemple de travaux pratiques (à titre indicatif)

- Création d'un projet classe / objet reprenant tous les principes de l'objet en Kotlin

Jour 2

Les fonctions - Partie 2

- Operator Overloading en Kotlin
- Lambda expression en Kotlin
- Extensions de fonctions en Kotlin
- Extensions de propriétés en Kotlin
- Closures en Kotlin
- Bonnes et mauvaises pratiques

Exemple de travaux pratiques (à titre indicatif)

- Création d'un programme de surcharge des opérateurs principaux pour les classes déjà créées

Délégation

- Concept de délégation en Kotlin
- Délégation de fonctions en Kotlin
- Délégation de propriétés en Kotlin
- Bonnes et mauvaises pratiques

Generics

- Generics en Kotlin
- Generics et invariance en Kotlin
- Covariance en Kotlin
- Contravariance en Kotlin
- Bonnes et mauvaises pratiques

Autres fonctionnalités

- Null safety
- Casting de types en Kotlin
- Tuples
- Deconstructing Values
- Gestion des exceptions
- Déclaration de constantes
- Annotation en Kotlin
- Bonnes et mauvaises pratiques

Exemples de travaux pratiques (à titre indicatif)

- Écriture de code permettant de mettre en lumière l'avantage de la délégation plutôt que l'héritage en Kotlin
- Ajout dans le code de Generics <>
- Ajout de la gestion des exceptions "try...catch" afin de rendre le programme plus solide

Jour 3

Interopérabilité

- Interopérabilité avec Java
- De Kotlin au Java
- Nulls de Java
- Le Kotlin dans Java
- Extensions de fonctions à partir du Java
- Interopérabilité avec les dernières versions de Java
- Java Réflexion avec Kotlin
- Kotlin Réflexion

Exemple de travaux pratiques (à titre indicatif)

- Exemple d'intégration de code Java dans une application Kotlin et inversement

Standard Library

- Kotlin Standard Library et collections dans Kotlin
- Filtering, Mapping et Flatmapping en Kotlin
- Kotlin lazy evaluation

Programmation asynchrone

- Le problème de la programmation asynchrone
- Coroutines en Kotlin et l'implémentation des coroutines
- Async et Await en Kotlin
- Yield en Kotlin
- Reactive extension en Kotlin
- Bonnes et mauvaises pratiques

Exemple de travaux pratiques (à titre indicatif)

- Mise en oeuvre de l'asynchronisme et de la programmation réactive en Kotlin

Kotlin et l'ouverture sur Android

- Développer une application Android sur AndroidStudio en Kotlin

Kotlin et l'IA

- Utiliser l'IA Générative pour concevoir des applications Kotlin
- Intégrer des API d'OpenAI dans Kotlin

Exemples de travaux pratiques (à titre indicatif)

- Création rapide d'une application Android en Kotlin avec Android Studio
- Exemple d'utilisation d'outils d'IA Générative comme ChatGPT pour créer du code Kotlin adapté à l'application

Le contenu de ce programme peut faire l'objet d'adaptation selon les niveaux, prérequis et besoins des apprenants.

Modalités d'évaluation des acquis

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation

Accessibilité de la formation

Le groupe M2i s'engage pour faciliter l'accessibilité de ses formations. Toutes nos formations sont accessibles aux personnes en situation de handicap : les détails de l'accueil des personnes sont consultables sur la page Accueil PSH.

Modalités et délais d'accès à la formation

Les formations M2i sont disponibles selon les modalités proposées sur la page programme. Les inscriptions sont possibles jusqu'à 48 heures ouvrées avant le début de la formation. Dans le cas d'une formation financée par le CPF, ce délai est porté à 11 jours ouvrés.