



PROGRAMMATION JAVA : LES BONNES PRATIQUES

— Public	Développeurs, architectes et toute personne impliquée dans des projets Java et souhaitant étendre ses compétences en programmation Java
— Durée	4 jours - 28 heures
— Pré-requis	Des connaissances de Java sont supposées acquises.
— Objectifs	Ce cours vous apprend à résoudre les problèmes concrets de développement de logiciels et à fournir des applications rapides et fiables. Vous apprendrez à tirer parti des fonctions avancées du langage Java et des API, à maîtriser la programmation concurrente avec les processus légers, à optimiser votre productivité grâce aux outils d'automatisation et enfin à sécuriser et améliorer les performances de vos applications.
— Méthodes pédagogiques	Pour bien préparer la formation, le stagiaire remplit une évaluation de positionnement et fixe ses objectifs à travers un questionnaire. La formation est délivrée en présentiel ou distanciel (e-learning, classe virtuelle, présentiel et à distance). Le formateur alterne entre méthodes démonstratives, interrogatives et actives (via des travaux pratiques et/ou des mises en situation). La validation des acquis peut se faire via des études de cas, des quiz et/ou une certification. 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 Audit Conseil Formation.
— Moyens techniques	1 poste de travail complet par personne De nombreux exercices d'application Mise en place d'ateliers pratiques Remise d'un support de cours Passage de certification(s) dans le cadre du CPF Remise d'une attestation de stage
— Modalité d'évaluation des acquis	Evaluation des besoins et objectifs en pré et post formation Evaluation technique des connaissances en pré et post formation Evaluation générale du stage
— Délai d'accès	L'inscription à cette formation est possible jusqu'à 5 jours ouvrés avant le début de la session
— Accessibilité handicapés	Au centre d'affaires ELITE partenaire d'ACF à 20 m. Guide d'accessibilité à l'accueil.

PROGRAMMATION EFFICACE EN JAVA

- Objectifs des bonnes pratiques
- Identifier les caractéristiques clés d'un logiciel de haute qualité

SIMPLIFIER LA GÉNÉRATION ET LE DÉPLOIEMENT DES PROJETS

- Automatisation du processus de génération en utilisant Ant
- Contrôle et configuration de la journalisation

MISE EN PLACE DU DÉVELOPPEMENT GUIDÉ PAR LES TESTS

- Constituer et maintenir les tests JUnit
- Validation des résultats des applications avec les tests fonctionnels
- Automatisation des tests sur l'intégralité du projet
- Tests de composants encapsulés tels que les servlets

RECOMMANDATIONS DES EXPERTS

- Équilibrer extensibilité et maintenabilité
- Bonnes pratiques pour la gestion des exceptions
- Limiter les problèmes de chargement de classe

CONTRÔLE DES TYPES

- Élimination des erreurs d'exécution grâce aux types génériques
- Limitation des valeurs de paramètre avec la canonicalisation

MISE EN PLACE DE L'ENCAPSULATION

- Fournir des macros méthodes avec le design pattern Memento et simplifier l'adaptation aux interfaces

CRÉER DES FRAMEWORKS FLEXIBLES

- Élargir l'applicabilité avec l'introspection
- Simplifier l'introspection avec les JavaBeans et les annotations

REFACTORISATION ET DESIGN PATTERNS

- Simplification du code source avec la refactorisation
- Conception d'interfaces pour une meilleure flexibilité logicielle
- Design patterns orientés objet clés
- Patron de méthode
- Stratégie
- Singleton
- Composite
- Factory
- Inversion de contrôle

AUTOMATISATION DES CONTRÔLES QUALITÉ DU CODE

- Normes applicables à l'intégralité du projet
- Suppression des erreurs de codage courantes
- Identification précoce des erreurs de conception

RÉGLAGES POUR UN MAXIMUM DE PERFORMANCES

- Outils d'analyse des performance, évaluation des temps de réponse, réalisation des tests de charge et stress
- Identification de goulots d'étranglement
- Techniques de gestion des problèmes de performances courants de Java
- Utilisation du ramasse-miettes
- Choix des paramètres adaptés pour la JVM et le container
- Évaluation des besoins de NIO et JNI
- Réorganiser les boucles pour améliorer les temps de réponse
- Traitement des données en flux continu pour diminuer les dépassements de mémoire

UTILISATION EFFICACE DE L'API COLLECTIONS

- Éviter les fuites de mémoire grâce aux références faibles
- Choix des meilleures classes collection

PARALLÉLISATION POUR UN MEILLEUR TEMPS DE RÉPONSE

- Écriture de code fiable et réentrant
- Éviter les pièges du multitâche: recouvrement et interblocages

SÉCURISATION D'UNE APPLICATION MULTITÂCHE

- Synchronisation des processus légers
- Techniques de partage de données entre les threads
- Conséquences de la synchronisation sur les performances

MISE EN PLACE DE CONTRAINTES DE SÉCURITÉ

- Codage sécurisé en Java
- Restrictions d'accès aux ressources protégées
- Établissement de règles de sécurité
- Application de la sécurité basée sur les rôles
- Authentification des utilisateurs dans des applications web

ÉTENDRE LES FONCTIONNALITÉS D'UNE APPLICATION

- Limiter l'impact des modifications avec Proxy Adapter
- Inversion de contrôle (IoC) par les Beans Factories
- Injection de comportement avec les aspects
- Doter une application de capacités de scripting

NOUS CONTACTER

Siège social

16, ALLÉE FRANÇOIS VILLON
38130 ÉCHIROLLES

Téléphone

04 76 23 20 50 - 06 81 73 19 35

Suivez-nous sur les réseaux sociaux, rejoignez la communauté !



ACF Audit Conseil Formation



@ACF_Formation

Dernière mise à jour : 03/04/2020