Programme de Formation

Java : Approfondissez vos compétences et développez des applications complexes

Description

Approfondissez Java, maîtrisez les concepts avancés et développez des applications modulaires et performantes avec des pratiques professionnelles et évaluables.

Points Forts

Formation immersive combinant théorie, exercices pratiques et projets concrets pour appliquer les concepts avancés de Java et renforcer vos compétences en conception, architecture et optimisation.

Public visé

- Développeurs
- Chargés de développement d'applications informatiques
- Chefs de projets proches du développement

Objectifs pédagogiques

- Maîtriser les aspects avancés du langage Java
- Comprendre comment appliquer les principaux Frameworks et librairies Java
- Comprendre le modèle de sécurité et le chargement des classes Java
- Savoir programmer en Java dans le contexte des bases de données relationnelles
- Savoir tester, debugger et optimiser ses applications
- Appréhender les nouveautés Java

Organisation

Durée : 28 heures

Mode d'organisation : Mixte

Contenu pédagogique

Description

Jour 1 : Comprendre la JVM et fiabiliser ses applications Java

Savoir tester, déboguer et optimiser ses applications (1/2)

- Historique de Java SE
- JVM, JRE et JDK
- Édition Oracle ou OpenJDK et licences respectives
- Code source et bytecode
- Package et modules
- Shell Java
- IDE, tests et débogueur
- Exemples de travaux pratiques : les participants réalisent un TP permettant de découvrir du Shell Java pour tester des expressions simples. Ils débuggent une application échouant à des tests unitaires.

Savoir tester, déboguer et optimiser ses applications (2/2)

- Gestion de la mémoire
- Pile et tas
- Le ramasse miette
- Référence Forte
- Référence Faible
- Référence fantôme
- Monitoring
- Console Java
- Java Mission Control (version Oracle uniquement)
- Exemples de travaux pratiques : les participants utilisent des références faibles dans un outil chargeant/déchargeant un graphe d'objets. Ils réalisent un Monitoring de l'application avec Java Console (jconsole) et Java Mission Control (jmc).

Jour 2 : Langage Java avancé

Maîtriser les aspects avancés du langage Java (1/6)

- Autoboxing des types primitifs
- Les varargs
- Les énumérations complexes
- Les données temporelles depuis Java 8
- Création, formatage, parsing des données temporelles
- Fuseaux horaires et paramètres locaux
- Exemples de travaux pratiques : les participants écrivent des types énumérés complexes pour gérer des familles, couleurs et valeurs d'un jeu de carte.

Maîtriser les aspects avancés du langage Java (2/6)

- Interface fonctionnelle
- Méthodes « default » et « static » des interfaces
- Les fonctions anonymes
- Référence de méthode objet
- Référence de méthode sur un objet particulier
- Référence de constructeur
- Référence, héritage et liaison tardive
- Référence et surcharges
- Référence et autoboxing
- Référence et conversions implicites sur les nombres

Quizz : les participants réalisent un quizz permettant de relier des fonctions anonymes, des méthodes et constructeurs à des types fonctionnels (inclure méthode sur un objet particulier, méthode objet d'une classe, méthode de classe, méthode héritée, méthode redéfinie, surcharge de méthodes ou constructeurs, l'autoboxing des types primitifs, les conversions implicites des types primitifs). Vérification avec JUnit.

- Exemples de travaux pratiques : les participants doivent écrire un type fonctionnel IntTernaryOperator et une méthode de composition avec un opérateur unaire sur les entiers Prendre 2 fonctions f et g en paramètres et calculer g(f(x,y,z)). Exemple : f(x) → x^2+3y+z et g(x) → $\sqrt{(x^2+1)}$.

Jour 3 : Langage Java avancé

Maîtriser les aspects avancés du langage Java (3/6)

- Inférence de type
- Déclaration de variable avec mot clé var
- Effacement de type
- Types génériques avec lower ou upper bound
- Covariance et contravariance

- Méthode générique
- Exemples de travaux pratiques : les participants doivent écrire des comparateurs de films (titre, année, durée) en utilisant une combinaison de critères, les ordres naturels ascendants ou descendant, et en ignorant la classe. Exploiter chaque comparateur pour trier une liste de films.

Maîtriser les aspects avancés du langage Java (4/6)

- Collections de Java
- Collections simplifiées pour l'écriture des tests
- Itérateurs
- Instruction for "each" vs méthode forEach
- Boîte à outils util.Collections
- Exemples de travaux pratiques : les participants doivent :
  - Trouver le film de durée maximale dans une liste de
  - Reverser une liste de film dans un ensemble ordonné de films par durée
  - Trouver le film de durée maximale dans cet ensemble, comparer les 2

Maîtriser les aspects avancés du langage Java (5/6)

- Les streams d'objets
- Les principes du map/reduce
- Interfaces fonctionnelles dédiées au map/reduce
- Filtrer les données
- Transformation des données
- Collecte des données
- Gestion particulière des types primitifs
- Utilisation des collecteurs prédéfinis
- Écrire un collecteur custom
- Génération de streams
- Combiner des streams
- Exemples de travaux pratiques : les participants reprennent l'exercice de la séquence précédente et doivent :
  - Collecter des statistiques sur la durée de films sur une plage d'années
  - Regrouper des films par réalisateur et collecter des statistiques sur sa carrière (années ou durées)
  - Partitionner des films suivant un critère d'année ou durée

Maîtriser les aspects avancés du langage Java (6/6)

- Les annotations
- Annotations standards et Méta-Annotations
- Rétention des annotations
- Cible des annotations
- Les arguments d'une annotation
- Lecture par réflexion d'une annotation
- Traitement avec javac ou annotation.processing
- Manipulation de bytecode avec Javassist
- Exemples de travaux pratiques : le formateur réalise un focus sur les annotations @Override et @FunctionalInterface.

<u>Jour 4 : Frameworks, données, sécurité et concurrence</u>

Comprendre le modèle de sécurité et le chargement des classes Java

- Api d'introspection
- La classe Class<T>
- Les autres classes du méta modèle
- Instanciations et appels dynamique

- Restriction de sécurité
- Exploitation dynamique des annotations
- Exemples de travaux pratiques : les participants doivent écrire :
  - (généricité et introspection) un outil générique transformant un Stream d'objets de type T en stream d'objets du sous type U, en ne gardant que les instances du type U.
  - (exploitation dynamique d'annotations) un mini Framework de validation de données par
  - (exploitation dynamique d'annotations) un mini Framework de persistance pour générer automatiquement des ordres SQL.

Comprendre comment appliquer les principaux Frameworks et librairies Java

- nio (NIO et NIO2) vs java.io
- Les flux d'entrées sorties
- Gestion du système de fichiers
- Pattern visiteur
- Sérialisation des objets
- Ressources Autocloseable et try-with-resources
- Exemples de travaux pratiques : Dans le cadre de l'exercice de la séquence précédente, les participants doivent :
  - rechercher une liste de fichiers correspondant à un motif.
  - rechercher dans un fichier texte les lignes correspondant à un motif.
  - écrire un indexeur de fichiers collectant la taille et date de dernière modification des fichiers d'une arborescence de fichiers en utilisant le pattern visiteur.
  - sauvegarder les résultats d'une recherche dans un fichier.

Comprendre comment appliquer les principaux Frameworks et librairies Java

- Données optionnelles
- Interface Optional<T>
- Types primitifs optionnels
- Traitement impératif optionnel
- Transformation d'une donnée optionnelle
- Objets optionnels imbriqués
- Exemples de travaux pratiques : dans le cadre de l'exercice de la séquence précédente, les participants doivent :
  - Faire une transformation d'un objet entité en objet DTO issu d'une recherche au résultat
  - Lier 1 objet film à son réalisateur. Les 2 objets sont issus de 2 recherches au résultat
  - Saisir, stocker et afficher la date de naissance d'un artiste de cinéma en fonction des paramètres locaux de l'utilisateur.
  - Écrire une méthode générique emptiesFirst qui construit un comparateur d'éléments optionnels de type T à partir d'un comparateur de type T, à l'image de nullsFirst de java.util.Comparator.

Savoir programmer en Java dans le contexte des bases de données relationnelles

- Spécification JDBC et drivers JDBC
- ORM, JPA et Hibernate
- Configuration ORM et gestion des sessions
- Mapping d'une classe simple avec une table
- Types temporels, énumérations
- Collections simples
- Mapping d'associations
- CRUD sur les objets
- Requêtes JPQL, HQL ou SQL natif
- Requêtes avec l'API Criteria
- Exemples de travaux pratiques : les participants doivent mapper les entités simples « Film et Artiste » avec les tables correspondantes dans une base de données relationnelle. Mapper les associations « jouer dans » et « réaliser par » entre ces 2 entités. Tester via des tests unitaires le CRUD sur ces 2

entités ainsi que quelques requêtes représentatives en JPQL ou avec l'API criteria (sélection avec prédicat simple, projection, jointure, agrégat, statistiques).

Appréhender les nouveautés Java

- Concurrence et Parallélisme
- Thread
- Méthode synchronized
- Mécanisme de moniteur avec wait et notify
- RentrantLock
- Pool de threads, tâches Future et ForkJoinTask
- Traitements parallèles sur les streams d'objets
- Exemples de travaux pratiques : les participants doivent :
  - Écrire un programme Producteur/Consommateur qui permet en parallèle d'écrire et lire dans un buffer.
  - Paralléliser les calculs de statistiques fait en map/reduce sur une liste de films. Comparer avec la version non parallèle

Prérequis

- Maitriser le langage Java
- Connaitre les concepts de bases de données relationnelles et du langage SQL
- Disposer d'une expérience en programmation Java.
- Un questionnaire de positionnement sera réalisé au préalable pour adapter la formation à vos besoins

Modalités pédagogiques

- Exercices concrets
- Cas pratiques
- Quiz d'évaluation des connaissances

Moyens et supports pédagogiques

- Mise à disposition de postes de travail
- Vidéoprojecteur, tableau blanc, supports numériques
- Documentation pédagogique numérique

Modalités d'évaluation et de suivi

- Positionnement en amont de la formation :
  - Un quiz de consolidation des pré-requis sera administré en amont de la formation
- Suivi « pendant » :
  - Feuilles de présence
  - Exercices pratiques
  - Évaluation « fin de formation »
  - Évaluation des acquis en fin de formation
  - Formulaires d'évaluation de la formation
- Évaluation à froid :
  - Suivi post-formation : Questionnaire de satisfaction à j+30