Programme de Formation

Java : Maîtriser les principes fondamentaux de la programmation

Description

Formation pratique pour apprendre les bases du langage Java, comprendre la programmation orientée objet et développer des applications simples et robustes.

Points Forts

Approche pratique avec exercices pour vous permettre de maîtriser les concepts fondamentaux de Java, la programmation orientée objet et les bonnes pratiques de développement, tout en renforçant votre autonomie et votre capacité à coder efficacement.

Public visé

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

Objectifs pédagogiques

- Connaître les concepts de base du langage JAVA et maîtriser sa syntaxe
- Utiliser les bibliothèques et API
- Comprendre les concepts de la programmation orientée Objet en Java,
- Créer une application Java
- Gérer les erreurs de code et utiliser les outils de débogage
- Appréhender les nouveautés Java

Organisation

Durée : 35 heures

Mode d'organisation : Mixte

Contenu pédagogique

Description

Jour 1 : Introduction à Java et concepts de base

Maîtriser un environnement de développement intégré pour programmer en Java

- Historique de Java
- JVM, JRE et JDK
- Code source et bytecode
- Environnements de développement
- Premier programme
- Exemples de travaux pratiques : Les participants réalisent un exercice consistant à : écrire, compiler et exécuter un programme Hello World dans un environnement de développement.

Connaître les concepts de base du langage JAVA et maîtriser sa syntaxe

- Les variables

- Les types simples
- Les opérateurs
- Les conditionnelles
- Les expressions conditionnelles ternaires
- Les expressions conditionnelles switch
- Les boucles
- Les tableaux
- Les chaînes de caractères et ses méthodes
- Exemples de travaux pratiques : les participants doivent réaliser un exercice consistant à implémenter l'algorithme d'Euclide.

Jour 2 : Programmation orientée objet en Java

Comprendre les concepts de la programmation orientée Objet en Java (1/2)

- Classe et objets
- Attributs
- Constructeurs
- Méthodes
- Accesseurs
- Les méthodes de la classe Object
- Les Enums
- Exemples de travaux pratiques : les participants doivent écrire une classe décrivant un point en 2D avec un nom et des coordonnées x et y. Ils doivent ajouter les méthodes « translater (Δx, Δy) et distance » avec un autre point. Redéfinir la comparaison et le hashage des objets points en tenant compte des 3 attributs.

Comprendre les concepts de la programmation orientée Objet en Java (2/2)

- Association de classes
- Héritage de classes
- Classe abstraite
- Interface
- Polymorphisme
- Gestion des packages
- Exemples de travaux pratiques : Les participant réalisent un cas pratique et doivent compléter le modèle de données géométriques avec : la classe PointPondéré ajoutant la notion de pointla classe abstraite Figure (nom et méthode translater), les classes Cercle et Polygone et l'interface Mesurable2D (périmètre et surface).

Jour 3 : Manipuler bibliothèques, API et collections

Utiliser les bibliothèques et API (1/3)

- Les collections
- Interfaces et implémentations
- La généricité
- Le pattern Iterateur et la boucle for « each »
- L'autoboxing
- Exemple de travaux pratiques : les participants doivent écrire un programme manipulant une collection de points 2D puis de figures.

Appréhender les nouveautés Java

- Les streams et le map/reduce
- Les interfaces fonctionnelles
- Les fonctions anonymes
- Les collecteurs
- Les comparateurs
- Exemples de travaux pratiques : les participant reprennent l'exercice de la séquence précédente et doivent : reprendre les collections de points 2D et de figures et collecter des statistiques usuelles

comme minimum, maximum, somme, moyenne. Filtrer la collection suivant un nom de figure ou des coordonnées. Écrire un comparateur de points 2D.

Jour 4 : Gérer fichiers et bases de données

Utiliser les bibliothèques et API (2/3)

- Les entrées/sorties
- Les différents types de flux
- Le système de fichier
- Écrire/lire dans un fichier texte
- Écrire/lire dans un fichier objet
- Exemples de travaux pratiques : Réalisation d'un TP consistant à sauvegarder une liste de points 2D dans un fichier texte puis dans un fichier objet.

Utiliser les bibliothèques et API (3/3)

- Accès aux Base de Données Relationnelles
- API JDBC
- Driver JDBC
- Requêtes de lecture et mise à jour
- Requêtes préparées
- Exemples de travaux pratiques : Réalisation d'un TP consistant à sauvegarder une liste de points 2D dans un fichier texte puis dans un fichier objet (suite de la séquence précédente).

Jour 5 : Créer des applications et tester le code

Créer une application Java

- IHM avec Swing et Awt
- Fenêtre et composants
- Gestionnaires de placement
- Événements d'action, de fenêtre, souris et clavier
- Exemples de travaux pratiques : les participants doivent construire une interface homme machine pour gérer une liste de points 2D.

Gérer les erreurs de code et utiliser les outils de débogage

- Tests unitaires avec JUnit
- Avant et après chaque test
- Avant et après un lot de tests
- Tests et exceptions
- Tests sur les collections
- Tests paramétrés
- Gestion dans l'IDE
- Gestion avec Maven ou Gradle
- Exemples de travaux pratiques : les participants doivent écrire les tests unitaires des formules géométriques des différentes figures. Tester la construction de figures bien formées (polygone d'au moins 3 points, cercle de rayon strictement positif).

Prérequis

- Connaître les principes de la programmation orientée objet
- Disposer d'une expérience sur un langage de programmation dans le développement d'applications.
- 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