
Programme de Formation

Programmation objet : Concevez et structurez des applications fiables et évolutives

Description

Développez vos compétences en programmation orientée objet : concevez, structurez et implémentez des applications évolutives avec UML, classes et design patterns.

Points Forts

Formation structurante alternant apports méthodologiques et exercices pratiques pour maîtriser les concepts clés de la programmation objet, concevoir des architectures cohérentes et produire un code lisible, réutilisable et maintenable.



Public visé

- Toute personne souhaitant évoluer vers les technologies du développement objet
- Développeurs
- Analystes programmeurs
- Chefs de projet techniques



Objectifs pédagogiques

- Comprendre les principes et les spécificités de la programmation orientée objet
- Passer d'une approche fonctionnelle à une approche Objet
- Découvrir l'impact de l'IA dans la programmation orientée Objet
- Mettre en œuvre un projet simple intégrant la programmation orientée objet et l'IA

Organisation

Durée : 21 heures

Mode d'organisation : Mixte

Contenu pédagogique



Description

Jour 1 : Découverte de la programmation orientée objet

Identifier les différents paradigmes de développement

- Définition et présentation des principaux paradigmes de programmation :
 - Programmation déclarative
 - Programmation procédurale
 - Programmation orientée objet
- Aperçu d'un langage et syntaxe pour chaque paradigme présenté
- Identifier qu'un langage de programmation peut être multiparadigme
- Exemples de travaux pratiques : présenter et décrire un paradigme de programmation, décrire les usages et montrer un exemple de code.

Comprendre les principes et les spécificités de la programmation orientée objet

- Présentation des évolutions et besoins du génie logiciel



- Aperçu des différences entre la programmation procédurale et la programmation objet
- L'apport de la programmation orientée objet pour un projet logiciel :
- Définition et illustration des principes de modélisation pour les phases de spécifications fonctionnelles ou techniques
- Définition et illustration des principes d'encapsulation, de couplage faible et cohésion forte pour l'organisation, la réutilisation et l'évolutivité du code.
- Exemples de travaux pratiques : présentation de schémas et tableaux comparatifs pour différencier les approches procédurales et objets

Découvrir l'impact de l'IA dans la programmation orientée Objet

- Mise en évidence des propriétés des objets avec des exemples
- Attributs (couleur, modèle,)
- Comportements (se déplacer, rouler,)
- Notion de type (catégoriser les objets, les regrouper en familles,)
- Notion de classe et d'instance (dans un programme, un objet est une instance de classe)
- Utiliser l'approche cognitive pour décrire des objets en s'appuyant sur l'IA
- Exemples de travaux pratiques : A partir d'une image, réaliser un brainstorming pour déterminer les caractéristiques d'un objet (attributs, comportements, type). Répéter l'exercice avec plusieurs images afin que la démarche soit de plus en plus fluide.

Jour 2 : Conception et modélisation

Passer d'une approche fonctionnelle à une approche Objet

- Initiation à la conception objet pour comprendre les bases de la modélisation : comment penser objet à partir d'un besoin fonctionnel ?
- Traduire un besoin fonctionnel simple en approche objet à l'aide d'un scénario (identifier les objets, leurs relations et leurs propriétés)
- *Exemples de travaux pratiques : Grâce à un scénario proposé sous forme de texte ou de schéma, déduire les objets (classes) et leurs relations nécessaires.*

Modéliser une application à l'aide d'UML (introduction)

- Présentation de UML et des différentes familles de diagrammes
- Présentation simplifiée du diagramme de classe
- Rappel sur les enjeux de la modélisation
- Exemple de passage de la conception au code : les générateurs de code
- Présentation et comparaison de plusieurs logiciels de modélisation
- Mise en pratique pour représenter des classes simples avec un logiciel de modélisation UML
- Exemples de travaux pratiques : A partir d'un énoncé, les participants modélisent des classes sur un diagramme de classe UML grâce à un logiciel

Maîtriser le langage et le vocabulaire lié à la programmation orientée objet

- Synthèse du vocabulaire de la programmation orienté objet et définitions.
- Terminologies à maîtriser : Les concepts de base : classe, objet, type, attribut, méthode, propriétés et membres, encapsulation
- Les relations : héritage, polymorphisme, surcharge, redéfinition, association, agrégation,
- Les concepts liés à la programmation : accesseurs, constructeur, destructeur, interface, signature
 - Représentation concrète des terminologies ou concepts Pour chaque définition abordée sur la séquence précédente, réalisation d'une description plus détaillée avec :
 - Une description ou scénario fonctionnel
 - Une représentation UML
 - Une représentation en code
- Quiz : Répondre à un questionnaire pour valider la compréhension des concepts et définitions, puis participer aux échanges lors de la phase de correction.
 - Exemples de travaux pratiques : Après avoir rappelé les principes de relations entre les classes, un exemple permet de modéliser et coder des associations, agrégations et composition pour comprendre l'intérêt de ces relations et leurs différences.

Jour 3 : Mise en oeuvre d'un projet orienté objet

Expliquer l'utilité des Framework dans une approche Objet

- Définition des Framework
- Les Framework, exemple type de réutilisabilité
- Conception d'application en utilisant des Framework
- Exploiter les objets d'une application dans un Framework choisi
- Importance de la conception applicative, l'approche Framework
- Exemples de travaux pratiques : Identification des principes de la POO à travers un projet existant qui se base sur un Framework

Mettre en œuvre des Design Patterns (introduction)

- Les Design Pattern principaux
- Découpage d'une solution en tiers Data, métier et présentation
- Conception des objets d'accès aux données
- Conception des objets métier
- Conception des objets graphiques
- Conception d'une architecture MVC
- Le liant : les interfaces
- Exemples de travaux pratiques : Les participants assistent à un "live coding" pour montrer l'usage de certains designs patterns. Ils peuvent regarder la démonstration et reproduire en même temps sur leur environnement si besoin.

Mettre en œuvre un projet simple intégration la programmation orientée objet et l'IA

- Série d'exercices pour remettre en pratique les concepts objets
- Cas pratique pour créer une application en programmation orientée objet
- Exemples de travaux pratiques : Un énoncé permet de guider les participants pour qu'ils mettent en œuvre les concepts objets à travers différents exercices dont la difficulté est croissante.

★ **Prérequis**

- Connaissances de base en programmation
- Expérience en conception d'applications
- Expérience en développement logiciel
- 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