

Déroulé de l'action

• Modalités

Session Inter/Intra
En présentiel/Classe virtuelle

• Horaires

9H00-12H30 / 13H30-17H00

• Méthode pédagogique

Alternance exposés théoriques
et exercices pratiques
(80% de pratique)

• Suivi et assistance

Support de cours adapté
au logiciel étudié et
au niveau suivi

Assistance téléphonique
gratuite et illimitée

• Modalité d'évaluation

Passage de la certification TOSA
en fin de formation

Attestation de stage

Emargement quotidien d'une
feuille de présence

• Accessibilité aux personnes handicapées

Pour tout besoin d'adaptation,
retrouver le contact de notre
référént handicap et les
modalités d'accueil sur la page :

[Infos pratiques/Situation de Handicap](#)

Introduction à la Programmation Objet

Objectif : Développeurs souhaitant acquérir des connaissances de base sur la méthode de programmation objet avant l'étude d'un langage orienté Objet.

À l'issue de la formation, le stagiaire sera capable de :

- Comprendre les principes et les spécificités de la conception Objet
- Passer d'une approche fonctionnelle à une approche Objet
- Savoir modéliser une application à l'aide d'UML (introduction)
- Comprendre l'utilité des Framework dans une approche Objet
- Savoir mettre en œuvre des Design Patterns (introduction)
- Mettre en œuvre les concepts objets à travers un programme simple

Public : Développeurs, analystes, chefs de projets souhaitant évoluer vers les technologies du développement objet

Prérequis : Connaissance des concepts de développement de logiciels.

- 3 jours -

Programmation procédurale Vs. programmation orientée objet

- Pourquoi travailler avec des objets
- Dualité données et traitement dans l'approche orientée objet
- Concepts de classe et d'objet
- Les avantages de l'encapsulation
- Modularité du code par ajout de bibliothèques
- Couplage faible Vs. Cohérente forte
- Notion de "Clean Code"

L'approche objet : les concepts de base

- L'instanciation ou la création d'un objet à partir d'une classe
- Utilisation de constructeurs
- Libération des ressources à l'aide des destructeurs
- Les concepts objet : les objectifs du monde Objet, les classes et les objets, les attributs, les méthodes, l'encapsulation, l'instanciation
- Traduction des concepts Objet en langage : les packages et les espaces de noms, les classes, les méthodes et leur visibilité, les attributs et leur visibilité, l'instanciation, l'appel de méthodes et la référence aux variables
- Organisation par package et espace de noms

Héritage et encapsulation

- Comment spécialiser une classe et réutiliser du code
- Un exemple concret pour comprendre l'utilité de l'héritage
- Redéfinir une méthode dans une classe fille avec le polymorphisme
- Notion de classes et de méthodes abstraites

Déroulé de l'action

- **Modalités**

Session Inter/Intra
En présentiel/Classe virtuelle

- **Horaires**

9H00-12H30 /13H30-17H00

- **Méthode pédagogique**

Alternance exposés théoriques
et exercices pratiques
(80% de pratique)

- **Suivi et assistance**

Support de cours adapté
au logiciel étudié et
au niveau suivi
Assistance téléphonique
gratuite et illimitée

- **Modalité d'évaluation**

Passage de la certification TOSA
en fin de formation
Attestation de stage
Emargement quotidien d'une
feuille de présence

- **Accessibilité aux personnes handicapées**

Pour tout besoin d'adaptation,
retrouver le contact de notre
référént handicap et les
modalités d'accueil sur la page :
[Infos pratiques/Situation de Handicap](#)

Introduction à la Programmation Objet

Introduction à UML

- UML un standard bien établi dans le monde industriel
- L'importance de la modélisation dans les projets complexes
- Présentation des différents diagrammes et points de vue
- Présentation des outils de modélisation : Enterprise Architect , Magic Draw, Visual Paradigm

Concevoir le système logiciel à l'aide d'UML

- Un point de vue, des diagrammes !
- Capturer les aspects fonctionnels avec le diagramme de use cases
- Modéliser la structure et l'architecture de votre application avec le diagramme de classes et le diagramme de composants
- Le point de vue dynamique avec les diagrammes d'activités et de séquences

Introduction aux Design Patterns

- Principes des solutions de conception cataloguées
- Méthodologie : définition des besoins techniques, des classes "types" du pattern, des collaborations entre classes
- Présentation des patrons de conception : origine, les 3 familles (création, structuration et comportement), autres patrons
- Présentation des principaux patrons de conception de chaque catégorie
- Documentation d'un patron de conception et présentation des différents diagrammes UML utilisés
- Bonnes pratiques : comment vous aider à choisir le bon patron pour un problème donné

Travaux pratiques

- **Pour chaque point du programme abordé :**
Présentation et démonstration par le formateur, mise en pratique sur les fonctionnalités abordées par le stagiaire avec l'appui du formateur et du groupe, feedbacks du formateur tout au long de l'activité.