

## Formation UML 2 - Initiation + Approfondissement

<b>Durée :</b>	5 jours
<b>Public :</b>	Développeurs, chefs de projet
<b>Pré-requis :</b>	Connaissance de la programmation objet
<b>Objectifs :</b>	Acquérir les concepts fondamentaux de l'analyse/conception UML
<b>Sanction :</b>	Attestation de fin de stage mentionnant le résultat des acquis
<b>Taux de retour à l'emploi:</b>	Aucune donnée disponible
<b>Référence:</b>	GéN468-F
<b>Note de satisfaction des participants:</b>	4,64 / 5

### Introduction

Besoin de modélisation : analyse et conception d'un projet informatique  
Présentation du langage : principe, historique et utilité  
Démarches de modélisation : UML et les méthodes d'analyse (Merise, Unified Process)  
Panorama des environnements de modélisation UML  
Terminologie générale : méta-modèle, vue, modèle,...  
Types de diagrammes : de structures, de comportements, d'interactions  
Positionnement des diagrammes dans le cycle de développement

**Atelier pratique : présentation et analyse de plusieurs études de cas (méthode UP)**

### Recueil et analyse des besoins

Diagramme des cas d'utilisation : présentation, fonctionnalités  
Description des éléments du diagramme : acteurs, cas d'utilisation  
Pré/Post conditions et Types de relations  
Méthodologie : identification des acteurs, description des cas d'utilisation, scénarios

**Atelier pratique : analyse d'un cahier des charges d'un projet et construction de diagrammes de cas d'utilisation**

### Rappel des concepts de l'objet

L'objet par rapport aux autres styles de programmation (impératif, procédural)  
Classes, objets et packages  
Méthodes et communication inter-objets  
Agrégation et encapsulation  
Héritage, polymorphisme, classes abstraites et interfaces

**Atelier pratique : application des différents concepts de la programmation orientée objets avec un langage de programmation ou avec une syntaxe algorithmique**

### **Conception globale (architecturale)**

Diagramme de séquence : interactions entre objets au cours du temps. Messages synchrone et asynchrone

Diagramme de communication : rôle des objets, interactions, concurrence de traitements...

Diagramme globale d'interaction

Diagramme de temps : variations au cours du temps

Diagramme de composants : description des modules de l'application et description des dépendances

Diagramme de structure composite : détail de la structure interne d'un composant composé

**Atelier pratique : Exemples multiples d'analyse et de modélisation de la dynamique du système**

### **Conception détaillée**

Diagramme de paquetages : organisation des différentes classes/couches de l'application

Diagramme de classes : représentation statique de la structure interne de l'application

Diagramme d'objets : représentation de l'état du système à un instant donné (expression des exceptions)

Diagramme d'activités : modélisation du flux objet/activité pour la réalisation d'une opération

Diagramme d'états-transitions : détail des transitions affectant l'état d'un objet

Diagramme de déploiement : répartition physique des composants du système

**Atelier pratique : Construction/Application des diagrammes définis à des structures complètes**

### **Concepts avancés**

Etude détaillée des spécifications : UML Infrastructure / Superstructure

Développement piloté par les modèles (MDA - MDD)

Utilisation de patron de conception dans les diagrammes de classes

Apport des frameworks

Imbrication de diagrammes

Outils de conceptions : fonctionnalités, rapport, génération de code (BOUML, ArgoUml,...)

**Atelier pratique : Analyse complète d'un projet à l'aide de la méthode Unified Process**