

OMG Certified UML Professional, préparation à la certification

UML OCUP 2 Foundation

Cours Pratique de 3 jours - 21h

Réf : OCU - Prix 2024 : 2 240CHF HT

Articulée autour d'une étude de cas, cette formation vous permettra d'apprendre et d'utiliser le langage UML de façon efficace et pragmatique dans vos projets de développement logiciel. Elle vous préparera également à la certification UML OCUP 2 Foundation d'OMG (Object Management Group).

OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

Utiliser UML dans les différentes activités d'ingénierie logicielle

Maîtriser la modélisation structurelle

Maîtriser la modélisation comportementale

Maîtriser tous les éléments adressés par la certification OCUP 2 Foundation

MÉTHODES PÉDAGOGIQUES

Les travaux sont réalisés sur papier, sous la forme d'atelier favorisant les échanges au sein du groupe.

ETUDE DE CAS

Elaboration d'une étude de cas complète, permettant une véritable compréhension de l'utilité des diagrammes dans un contexte réel.

CERTIFICATION

Cette formation inclut un voucher vous permettant de passer l'examen de certification "OMG Certified UML Professional" dans un centre Pearson Vue. La langue de passage de la certification est l'Anglais.

LE PROGRAMME

dernière mise à jour : 08/2018

1) Pourquoi modéliser ?

- L'intérêt de la modélisation UML dans le projet de développement, depuis la modélisation métier jusqu'à l'implémentation
- L'organisation de la modélisation à l'aide de la structuration en packages.
- Prise en main de l'étude de cas : comprendre les besoins généraux du client.

2) La modélisation du métier

- Modéliser les processus métier avec le diagramme d'activité.
- Modéliser le domaine avec le diagramme de classes.
- Modéliser les statuts des entités métier avec le diagramme d'état-transitions.

3) La modélisation des exigences

- Modéliser l'environnement externe du système avec les acteurs.
- Définir le comportement du système avec les cas d'utilisation. Diagramme de cas d'utilisation. Diagramme de séquence.
- Les exigences non fonctionnelles à l'aide de l'approche FURPS et la notion de "requirement" UML.

PARTICIPANTS

Architectes logiciel, développeurs, chefs de projet, MOA, AMOA, MOE.

PRÉREQUIS

Expérience requise en développement logiciel, dans un langage orienté Objet de préférence.

COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

4) La modélisation de la conception du système. Vue structurelle

- Définir la plateforme technique - diagramme de déploiement.
- Définir l'architecture du code. Diagramme de packages.
- Bonnes pratiques de séparation des préoccupations et limitation des dépendance - pattern de structuration en couches.
- Modéliser les classes d'un point de vue structurel ? Diagramme de classes.
- Définir les attributs.
- Lier les objets avec les associations entre classes.
- Factoriser du code avec la généralisation : du bon usage de l'héritage.

5) La modélisation de la conception du système. Vue comportementale

- Décrire les interactions internes au système avec le diagramme de séquence.
- Définir les opérations pour répondre aux exigences fonctionnelles.
- Synthétiser le comportement d'un objet avec le diagramme d'état-transitions.

6) Modéliser les composants déployables

- Définir les composants et leurs interfaces - diagramme de composant.
- Définir le déploiement des composants - diagramme de déploiement.

7) Préparation au déroulement pratique de l'examen OCUP 2

- Révision générale.
- Entraînement sur des questions types.

LES DATES

Nous contacter