Date de publication : 27/06/2024

Formation : Conception orientée objet

ingénierie logicielle objet

Formation pratique - 4j - 28h00 - Réf. COB

Prix : 2470 CHF H.T.

Comment aborder la conception orientée Objet ? Comment passer d'une approche fonctionnelle à une approche Objet ? Comment écrire un programme orienté Objet possédant de réelles capacités d'évolutivité et de réutilisabilité ? Ce cours vous propose une maîtrise conceptuelle et pratique de la conception Objet.

Objectifs pédagogiques

À l'issue de la formation, le participant sera en mesure de :

- Comprendre les principes et les spécificités de la conception par objets.
- Passer d'une approche fonctionnelle à une approche Objet.
- Modéliser un logiciel objet à l'aide de la notation UML.
- Traduire le modèle UML dans un langage objet.
- Décrire les approches par frameworks et composants.
- Apprendre à mettre en œuvre des Design Patterns.

Public concerné

Développeurs, chefs de projets souhaitant se former à la conception orientée Objet.

Prérequis

Connaissances de base en conception d'applications et en développement logiciel.

Vérifiez que vous avez les prérequis nécessaires pour profiter pleinement de cette formation en faisant ce test.

Méthodes et moyens pédagogiques

Travaux pratiques

Les points abordés seront illustrés par de nombreux exercices spécialement choisis pour mettre en évidence les principes et les spécificités de la conception par objets, depuis l'analyse initiale jusqu'à l'implémentation dans un langage Objet.

PARTICIPANTS

Développeurs, chefs de projets souhaitant se former à la conception orientée Objet.

PRÉREQUIS

Connaissances de base en conception d'applications et en développement logiciel.

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.

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.

Programme de la formation

1 Qu'attendre de l'approche Objet ?

- Pourquoi utiliser des technologies à objets ?
- Les défis de la nouvelle informatique : modularité (Plug-Ins), réutilisabilité, évolutivité.
- L'utilisation de bibliothèques de composants. Comment l'approche Objet répond à ces défis ?
- Dans quel état d'esprit aborder un problème Objet ?
- Les acquis provenant des autres domaines de l'informatique et des autres disciplines.

2 Les concepts de base de l'approche Objet

- Les objets : une dualité procédure/donnée.
- Les classes comme modèles de structure et de comportement des objets, les instances comme représentants des classes.
- Les méthodes, des procédures définies dans les classes et utilisées par les instances.
- Les interactions entre objets par envois de messages. Comment les messages sont interprétés par les objets ?
- L'héritage. Héritage et typage des variables dans les langages fortement typés (C++, Java).

3 Diagrammes et représentation des objets à l'aide d'UML

- Les principaux diagrammes (diagrammes de classe, diagrammes de séquence) et leur utilisation pour la conception Objet.
- Les outils de notation et représentation des objets : prise en main d'un modeleur du marché.

4 Les grands principes de la conception objets

- Que met-on sous la forme d'un objet ? Principe de réification.
- Critères à appliquer pour décider de ce qui doit être mis sous forme Objet. Les erreurs à éviter.
- Comment structurer un logiciel Objet ? Principe de modularité et de décomposition des domaines.
- Comment structurer un ensemble de classes ? Principe d'abstraction et de classification.
- Comment penser l'interaction entre objets ? Principe d'encapsulation et d'autonomie.
- Analyser des systèmes complexes en termes de communications. La démarche générale. Les erreurs à éviter.
- Critères à appliquer pour disposer de "bonnes" hiérarchies de classes. Les erreurs à éviter.

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 formations pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
• À l'issue de chaque formation 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 participant 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

Pour toute question ou besoin relatif à l'accessibilité, vous pouvez joindre notre équipe PSH par e-mail à l'adresse psh-accueil@orsys.fr.

5 Comment aborder un logiciel Objet ?

- Les principes de développement. Du développement en spirale au développement incrémental.
- Identification des entités du domaine et description des interactions. Réutilisation et évolutivité des programmes.
- Concevoir par objets, ce n'est pas utiliser un outil Objet !
- Les erreurs à éviter.

6 De la conception à l'implémentation

- Traduire les diagrammes de classe UML dans des langages de programmation et dans des bases de données.
- Les principes de mise en œuvre d'applications objet. L'importance du distribué. Modèles clients-serveurs généralisés.
- Les grandes plateformes objets actuels : les technologies .NET de Microsoft et JEE de SUN.
- Comparaison de leurs points forts et de leurs points faibles.
- L'importance du distribué. Bibliothèques de classes. Langages de programmation et d'utilisation de composants.

7 L'approche par frameworks et composants

- Le problème du cycle de vie des logiciels.
- Les problèmes d'évolution et de maintenance nécessitent une approche logicielle permettant l'évolution.
- L'approche par frameworks et composants, qui est fondée sur la pensée Objet, est une réponse à cette nécessité.
- Comment concevoir et réaliser des applications rapidement à partir de frameworks et de composants réutilisables ?
- Comment intégrer des composants logiciels dans un framework existant ? Comment construire des frameworks ?
- Savoir reprendre une application existante pour la transformer en framework et la rendre ainsi évolutive.
- Grandes classes de frameworks. Les modèles de composants actuels.

8 Les Design Patterns

- Comment réutiliser de l'expérience lors de la conception et du développement d'applications objets ?
- Les Design Patterns ou "patrons de conception" comme solutions logicielles issues de problèmes généraux récurrents.
- Les différents types de Design Patterns. Exemple de Design Patterns.
- Avantages et limites des Design Patterns.
- Comment utiliser pratiquement des Design Patterns ? Apprendre à mettre en œuvre des Design Patterns par la pratique.