Date de publication : 05/09/2025

Formation : Design Patterns, mise en œuvre

Formation pratique - 5j - 35h00 - Réf. DES
Prix : 2500 € H.T.

4,9 / 5

BEST

Cette formation vous formera au design des applications et aux pratiques de conception modernes telles que le développement guidé par les tests et le refactoring. Les nombreux cas pratiques vous apprendront à créer des applications évolutives et réutilisables en prenant en compte les principaux patterns de conception.

Objectifs pédagogiques

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

- Comprendre les principes fondamentaux de la conception Objet
- Appliquer les règles fondamentales de découpage d'une application en package
- Appliquer les principes de construction des classes d'une application
- Mettre en pratique le développement piloté par les tests
- Mettre en œuvre les principaux Design Patterns

Public concerné

Concepteurs, développeurs, architectes ou chefs de projet.

Prérequis

Connaissance d'un langage Objet.

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

Méthodes et moyens pédagogiques

Travaux pratiques

Les ateliers réalisés par les participants seront effectués avec le langage de leur choix (Python, Java, C++, C# ou VB, .Net).

PARTICIPANTS

Concepteurs, développeurs, architectes ou chefs de projet.

PRÉREQUIS

Connaissance d'un langage Objet.

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 Présentation du design

- Rappel des fondamentaux de la POO et d'UML.
- Les apports d'UML pour la conception.
- Les enjeux de la conception.
- L'utilisation de l'héritage. Avantages et inconvénients.

2 Principes fondamentaux en conception Objet

- Les principes d'ouverture/fermeture (OCP) et de substitution de Liskov (LSP).
- Concept de polymorphisme, de couplage faible et de forte cohésion.
- L'impact de la conception objet sur les projets.

Travaux pratiques

Découpage des responsabilités entre les classes.

3 Principes de construction des classes

- La gestion des dépendances avec l'inversion de dépendance (DIP).
- La réduction de la complexité apparente par la séparation des interfaces (ISP).
- La répartition des responsabilités avec le GRASP.

4 Principes d'organisation en packages

- Le package : une unité de conception livraison/réutilisation (REP) et la réutilisation commune (CRP).
- Le découpage des packages. Le CCP.
- L'organisation entre packages.

Travaux pratiques

Construction et hiérarchisation des packages.

5 Développements pilotés par les tests

- Approche Test Driven Development (TDD) versus approche Model Driven Engineering (MDE).
- Écriture des cas et de suites de tests.

Travaux pratiques

Attribution des responsabilités aux composants logiciels via l'approche TDD.

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.

6 Principes des Design Patterns

- Les Design Patterns pour réutiliser l'expérience.
- Périmètre, avantages et limites des Design Patterns.
- Répondre à des problèmes récurrents.
- Les patterns fondateurs de Gamma et GoF : les patterns de création, de comportement, de structure.
- La refactorisation. Pourquoi refactoriser ?
- Modification de la présentation du code et de l'algorithmique des classes. Refonte de la conception.

Travaux pratiques
Exemple de conception, refactorisation et programmation avec des patterns GoF.

7 Architecture logicielle et patterns architecturaux

- Des exigences à l'architecture.
- Styles architecturaux.
- Patterns de distribution (Client Serveur Style, Data Bus Pattern, Blackboard, Repository).
- Patterns de conception de systèmes (MVC, architecture en couches, Plug-in Style, Pipeline).

8 Processus de développement

- Concevoir dans un processus itératif et incrémental.
- Le manifeste Agile. XP, Scrum.

Parcours certifiants associés

Pour aller plus loin et renforcer votre employabilité, découvrez les parcours certifiants qui contiennent cette formation :

- Parcours certifiant Concevoir et développer une application informatique en Python - Réf. ZCT

Dates et lieux

CLASSE À DISTANCE
2026 : 4 mai, 18 mai, 29 juin, 7 sep., 21 sep., 19 oct., 16 nov.

PARIS LA DÉFENSE
2026 : 18 mai, 29 juin, 7 sep., 16 nov.

LILLE
2026 : 29 juin, 16 nov.