Stage pratique de 5 jour(s)
Réf : DES

Participants

Concepteurs, développeurs, architectes ou chefs de projet.

Pré-requis

Connaissance d'un langage Objet.

Prix 2019 : 2710€ HT

Dates des sessions

AIX
02 déc. 2019, 30 mar. 2020
29 juin 2020

BORDEAUX
02 déc. 2019, 23 mar. 2020
27 juil. 2020, 28 sep. 2020

LILLE
02 déc. 2019, 16 mar. 2020
20 juil. 2020, 21 sep. 2020

LYON
02 déc. 2019, 06 avr. 2020
20 juil. 2020

NANTES
02 déc. 2019, 02 mar. 2020
06 juil. 2020, 07 sep. 2020

PARIS
21 oct. 2019, 04 nov. 2019
02 déc. 2019, 13 jan. 2020
16 mar. 2020, 11 mai 2020
20 juil. 2020, 21 sep. 2020

SOPHIA-ANTIPOLIS
02 déc. 2019, 02 mar. 2020
06 juil. 2020, 07 sep. 2020

STRASBOURG
02 déc. 2019, 30 mar. 2020
29 juin 2020

TOULOUSE
02 déc. 2019, 23 mar. 2020
27 juil. 2020, 28 sep. 2020

Modalités d'évaluation

L'évaluation des acquis se fait tout au long de la session au travers des multiples exercices à réaliser (50 à 70% du temps).

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.

Design Patterns, mise en œuvre

Ce stage 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 PEDAGOGIQUES

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

Travaux pratiques

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

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).
- Ecriture des cas et de suites de tests.

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

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.

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.

• A 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.

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.