

.NET, maîtriser MVVM pour la création d'applications WPF

Cours Pratique de 3 jours - 21h
Réf : NWP - Prix 2025 : 1 910 HT

Cette formation présente le design pattern MVVM et ses avantages, notamment la testabilité. Au cours de TP successifs en C# sous Visual Studio, vous procéderez (notamment en utilisant MS MVVM Toolkit) à la transformation progressive d'une application WPF simple en application WPF respectant l'architecture MVVM.

OBJECTIFS PÉDAGOGIQUES

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

- Comprendre les règles et les concepts d'architecture applicatives
- Comprendre l'intérêt de l'architecture MVVM avec WPF
- Savoir créer et intégrer des ViewModels aux vues
- Mettre en oeuvre des tests unitaires sur les ViewModels
- Installer et maîtriser MS MVVM Toolkit

MÉTHODES PÉDAGOGIQUES

Alternance entre présentation magistrale, illustrations de cas concrets et travaux pratiques.

TRAVAUX PRATIQUES

Ateliers permettant de mettre MVVM en œuvre, à partir d'une application WPF liée à des données sans architecture MVVM.

LE PROGRAMME

dernière mise à jour : 08/2024

1) Rappels des fondamentaux des applications WPF

- Les concepts fondamentaux du XAML.
- Le DataBinding.
- Techniques de validations avec affichage des erreurs.
- Intérêt des commandes versus événements routés.
- Simplifier les commandes avec la classe RelayCommand.

Travaux pratiques : Analyse d'une application WPF typique liée à des données avec validations, sans architecture MVVM.

2) Intérêt de l'architecture MVVM

- Règles de conception SOLID et SOC.
- Modèles de conception courants (Design Patterns, Modèle en couches, MVC, MVP, Repository et Locator).
- L'architecture MVVM : découpage et interactions entre les différentes parties.

Travaux pratiques : Analyse des étapes de transformation d'une application WPF simple liée à des données avec du code-behind vers une architecture MVVM.

3) Refactoring d'une application type

- Analyse d'une application type réalisée en code-behind (sans MVVM), avec liaisons de données.
- Organisation et création des ViewModels.
- Intégration des ViewModels aux vues de manière déclarative et par code.

PARTICIPANTS

Développeurs .NET souhaitant comprendre l'intérêt du pattern MVVM.

PRÉREQUIS

Bonnes connaissances C# avec le framework 4 ou équivalentes à celles apportées par le cours ".NET, création d'applications WPF" (réf. NPF).

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

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.

- Intégration des commandes aux ViewModels et aux vues, avec gestion de la disponibilité de leur exécution et paramétrage.

Travaux pratiques : Remplacement progressif du code-behind des vues de l'application par des ViewModels dérivés d'une classe BaseViewModel, avec intégration des différents types de commandes.

4) Outils et patterns liés à MVVM

- Analyse des problématiques de couplage entre classes concrètes.
- Principe des patterns DI (Injection de Dépendances) et du Locator.
- Principe et mise en oeuvre de tests unitaires.
- Le Mocking pour faciliter les tests.

Travaux pratiques : Réalisation d'une classe ViewModelLocator pour gérer les ViewModels et les services avec injection de dépendances. Création d'un projet de tests unitaires sur les ViewModels.

5) Mise en oeuvre de MS MVVM Toolkit

- Aperçu comparatif des outils et des principaux Frameworks MVVM disponibles.
- MS MVVM Toolkit : présentation des fonctionnalités et installation.
- Principe et mise en oeuvre de l'IOC avec le système d'injection de dépendances.
- Intérêt et mise en oeuvre de la classe Messenger pour faire communiquer les ViewModels.

Travaux pratiques : Amélioration de l'application avec MS MVVM Toolkit grâce aux outils présentés (ObservableObject, Messenger et IServiceProvider).

LES DATES

CLASSE À DISTANCE
2025 : 08 oct.

PARIS
2025 : 01 oct.