

Formation .NET: Conversion des applications VB en VB .NET (VBNETMIGFR, 4 jours)

Description

Le cours Convertir des applications VB en VB .NET (formation .NET) couvre tous les aspects du processus de migration d'une version antérieure de VB vers une application .NET v4. Nous commençons par un aperçu des principales différences fonctionnelles entre les applications VB traditionnelles et les applications .NET v4. La discussion comprend tous les éléments de code de base ainsi qu'une exploration complète du refactoring en tant qu'art et science. Le cours présente des solutions pour les principales odeurs de code et aborde les concepts essentiels de hiérarchies de classes et de polymorphisme. Après avoir maîtrisé les outils de base de la migration, notre attention se porte sur les considérations de migration associées à l'utilisation des modèles d'entités .NET, d'ASP .NET MVC et de WCF. Le cours se termine par une discussion des principales différences dans les domaines de la configuration et du déploiement.

Tarifs

- Tarification: \$3,750/person
- Rabais de 10% lorsque vous inscrivez 3 personnes.

Plan de cours

Versionnage et capacités de .NET

Exploration de l'historique des versions de .NET : de la v1 à la v4

Comparaison de version à version : Capacités du code VB .NET

Comparaison entre les versions : Accès aux bases de données

Comparaison entre les versions : XML

Comparaison entre les versions : Services Web

Comparaison entre les versions : Programmation Web

Comparaison entre les versions : Programmation Windows

Comparaison entre les versions : Sécurité

Comparaison entre les versions : Documentation

Comparaison entre les versions : Compilation et déploiement

Une remise à niveau de la programmation orientée objet

Meilleures pratiques pour la création de solutions et de projets

À propos de la gestion des espaces de noms et des assemblages

Configuration de l'environnement pour la documentation automatique

Meilleures pratiques pour la création de classes

Assurer la cohésion de la classe

À propos des classes de collection .NET

Utiliser l'héritage de manière significative

Meilleures pratiques en matière d'héritage

Utiliser le polymorphisme de manière significative

Meilleures pratiques en matière de polymorphisme

La nature et l'utilisation des classes et méthodes abstraites

La nature et l'utilisation des interfaces

La nature et l'utilisation des génériques

L'art du refactoring

Qu'est-ce que le refactoring ?

Une introduction au développement de Test Drive

Le processus de refactoring de A à Z

Présentation des odeurs de code

Les odeurs courantes du code et comment les corriger

Identifier et résoudre les codes en double

Identifier et résoudre les grandes classes et fonctions

Identifier et résoudre l'obsession primitive

Identifier et résoudre les longues listes de paramètres

Identifier et résoudre les changements divergents

Tout mettre en place

Migrer le code procédural

Migration des déclarations de variables

Migration de l'affectation des variables : Utilisation de += et d'autres expressions similaires

Migration des déclarations de fonctions et de procédures

Note sur les paramètres de fonction et les valeurs de retour

À propos des paramètres anonymes et facultatifs

Migration des tableaux et des collections

Migration des expressions booléennes : AndAlso et OrElse

Migration des expressions conditionnelles

Migration des déclarations répétées

Autres considérations relatives à la migration

Migration des classes

Migration des déclarations et des attributs de classe

Migration des membres de la classe

Migration des constructeurs et finaliseurs de classe

Migration des méthodes de classe

Notes de refactoring : Créer, supprimer et exploser des classes existantes

Migration des hiérarchies de classes

Un regard sur l'héritage : Migration des sous-classes

À propos de la création d'une nouvelle hiérarchie de classes : Quand choisir cette voie

Ajout de méthodes de la classe de base : Quand est-ce justifié ?

Un regard sur les interfaces : Migration des interfaces et de leurs implémentations

A propos de la création d'une nouvelle interface : Quand choisir cette voie

Mise en œuvre d'une nouvelle interface : Comment le faire.

Migration de l'accès aux données

Aperçu des technologies d'accès aux données de .NET v4

Comparaison entre les modèles d'entités .NET et les implémentations ORM courantes

Mise en œuvre d'un modèle d'entité .NET

Utiliser le modèle d'entité .NET dans votre code

Migration du code d'accès aux données vers une implémentation de modèle d'entité

Meilleures pratiques et erreurs courantes

Migration des applications Web

Aperçu des technologies d'application Web .NET v4

À propos d'ASP.NET MVC : Technologie et concepts

Création d'une application Web simple basée sur ASP .NET MVC 3

Migration des applications Web ASP .NET vers ASP .NET MVC : un aperçu

Migration de l'interface : HTML

Migration de la logique : de Code Behind à ASP.NET MVC

Intégration des implémentations de modèles d'entités .NET

Meilleures pratiques et erreurs courantes

Migration des services

Aperçu des technologies de services Web .NET v4

À propos de WCF : technologie et concepts

Création d'un service simple basé sur WCF

Migration des services Web vers WCF : une vue d'ensemble

Migration de la classe de service et de ses comportements

À propos des canaux et des comportements

Migration du client de service Web : Appeler le service Web

Intégration des implémentations de modèles d'entités .NET

Meilleures pratiques et erreurs courantes

Migration de la configuration

Aperçu des différences de configuration

Configuration des applications avec web.config

Ajout et utilisation des paramètres de configuration

Travailler avec des jeux de configuration chiffrés

Sécurisation de l'application

Autres considérations relatives à la migration

Préparation du déploiement

Une note spéciale sur le déploiement des applications ASP.NET MVC

Exécution d'un déploiement

Vérification du déploiement