

Programmation en C# : Fonctions avancées - 4 jours

Rationalisation des applications Entity Framework

formation 973

Ce cours est disponible en ligne et en direct via Learning Tree AnyWare™, notre solution de formation présentielle à distance. Contactez-nous dès maintenant au 01 49 68 53 00 pour plus d'informations ou pour vous inscrire !

- Vous apprendrez à**
- Rationaliser les applications centrées sur les données avec les fonctions avancées de C# et l'Entity Framework
 - Appliquer les expressions lambda et les méthodes d'extension pour un traitement du niveau intermédiaire
 - Exploiter les mots clés de LINQ (Language Integrated Query) pour filtrer et trier les données
 - Transférer la logique complexe avec des délégués génériques **Func<T,R>**
 - Générer des énumérations dynamiques avec **IEnumerable<T>** dans vos propres classes génériques
 - Utiliser LINQ pour simplifier le traitement du XML

Objectif C# est devenu un langage de manipulation des données à part entière. Les fonctionnalités étendues de .NET permettent aux programmeurs de simplifier l'accès aux données, d'augmenter la productivité et d'améliorer les performances globales des applications. Dans cette formation, vous mettez en œuvre les fonctionnalités étendues de C#, y compris LINQ et l'Entity Framework, pour intégrer efficacement les fonctions orientées objet et de manipulation des données.

À qui s'adresse cette formation Aux programmeurs C# expérimentés qui veulent améliorer leur savoir-faire en développement de logiciels grâce aux fonctionnalités étendues du langage, plus particulièrement LINQ, conjointement avec l'Entity Framework. Une expérience de la programmation en C# du niveau de la formation 419, "Programmation en C# : Les fondamentaux", est nécessaire.

- Travaux pratiques** Les exercices, basés sur une étude de cas progressive permettant la mise en œuvre des fonctionnalités étendues du C#, comprennent:
- Application des auto-propriétés implémentées automatiquement et des initialiseurs d'objets
 - Écriture d'expressions lambda et de méthodes d'extension
 - Implémentation du niveau données avec LINQ et l'Entity Framework
 - Utilisation d'objets DataContext et de mots-clés de requêtes LINQ pour accéder à une base de données et la modifier
 - Mappage de classes d'entités avec le Concepteur O/R
 - Mise en place d'énumérations dynamiques avec **yield**
 - Analyse de documents XML avec LINQ
 - Accès aux procédures stockées avec C#/LINQ

Programmation en C# : Fonctions avancées - 4 jours

Rationalisation des applications Entity Framework

formation 973

Ce cours est disponible en ligne et en direct via Learning Tree AnyWare™, notre solution de formation présentielle à distance. Contactez-nous dès maintenant au 01 49 68 53 00 pour plus d'informations ou pour vous inscrire !

Introduction

- Programmation orientée objet : révisions
- Implémentation d'une interface
- Les technologies d'accès aux données
- Collections génériques et non génériques

Exploiter les fonctionnalités du langage C#

Constructions du langage

- Les propriétés implémentées automatiquement
- Les avantages de la déclaration implicite de type

Enrichissement de la logique de programmation

- Instanciation d'objets entités
- Simplification avec les initialiseurs d'objets
- Construction d'objets anonymes

Méthodes d'extension pour la manipulation des données du niveau intermédiaire

Filtre et tri de données avec les expressions lambda

- Construction d'expressions lambda
- Comparaison des délégués et des expressions lambda
- Méthodes d'extension Count, Reverse, Union, Except et autres
- Passage de types et de fonctionnalités aux méthodes

Application des fonctionnalités d'extension

- Flexibilité du code grâce aux délégués
- Paramétrage des délégués et d'expressions lambda
- Simplification du code avec les délégués génériques `Func<T,R>`

Les mots-clés des requêtes LINQ

Syntaxe et sémantique

- Codage de requêtes LINQ
- Tri de données et d'objets
- Filtrage avec `from`, `where`, `orderby` et `group`

Itération sur les collections

- Interface entre C# et LINQ avec `IEnumerable<T>`
- Conversion de `IEnumerable<T>` en `List<T>`

Adaptation de LINQ et de l'Entity Framework

Comparaison entre ADO.NET et LINQ

- Objets DataSet et listes génériques
- SqlDataReader et `IEnumerable<T>`
- Traitement des informations du niveau données

Connexion à une base de données et lecture

- Création d'un ObjectContext
- Rattachement à des bases de données via l'Entity Framework (EF)
- Corrélation entre classes d'entités et tables de données
- Préservation du paradigme orienté objet

L'outil Concepteur Objet/Relationnel

- Mise en correspondance des tables de données et classes d'entités
- Mise en place de relations d'héritage
- Importation de procédures stockées

Mise à jour d'une base de données

- Insertion, modification et suppression de données
- Traitement des erreurs et exceptions
- "Commit" et "roll back" de transactions

Énumérations et classes génériques

Étude d'enum

- Comparer `IEnumerable<T>` et `IEnumerator<T>`
- Génération d'énumérations dynamiques
- Le mot-clé `yield`

Écriture de classes génériques

- Minimiser les classes dupliquées
- Implémentation d'un énumérateur générique rapide

Traitement de données à l'aide de requêtes LINQ

Manipulation des données

- Combinaison et agrégation de données similaires avec `group`
- Mise en œuvre de jointures internes, externes et groupées
- Création de sous-ensembles de données avec la clause `into`

Techniques LINQ avancées

- Création de résultats de requêtes anonymes
- Obtention de vues composites à l'aide de clauses `from`

- Optimisation de requêtes LINQ avec des délégués et des expressions lambda

Application de LINQ à XML

Exploitation de l'espace de nom XML

- Charger du XML dynamiquement via le Web
- Créer et sauvegarder du contenu XElement

Traitement du XML

- Accès au document, aux éléments et aux attributs
- Analyse d'un document XML avec LINQ