

PROGRAMMATION C# avec le .Net Framework 4.5

Réf : 20483

Durée : 5 jours

Objectifs :

- Disposer de solides bases pour développer des applications en C#
- Comprendre comment tirer parti de la puissance du Framework avec C#
- Mettre en oeuvre la programmation orientée objet (POO) avec C#
- Acquérir les connaissances et compétences nécessaire pour exploiter des données avec LINQ
- Apprendre à crypter et décrypter les données

Connaissances préalables :

- Avoir une expérience en programmation C, C++, JavaScript, Objective-C, Microsoft Visual Basic, ou Java
- Connaissance du concept de la programmation orientée objet

PLAN DE COURS

Vue d'ensemble de la syntaxe C# :

- Vue d'ensemble de l'écriture d'applications en C#
- Types de données, opérateurs et expressions
- Structures de programmation du langage C#

Créer des méthodes, gérer les exceptions et surveiller les applications :

- Créer et appeler des méthodes
- Créer des méthodes surcharges et utiliser les paramètres optionnels et de sortie
- Gestion des exceptions
- Surveillance des applications

Développer le code des applications graphiques :

- Implémenter structures et énumérations
- Organiser les données dans des collections
- Gestion d'évènements

Créer des classes et implémenter des collections à typage sûr :

- Création de classes
- Définition et implémentation d'interfaces
- Implémentation de collections à typage sûr

Créer une hiérarchie de classes par héritage :

- Création de hiérarchie de classes
- Étendre les classes du .Net Framework
- Création de types génériques

Lecture et écriture de données locales :

- Lecture et écriture de fichiers
- Sérialisation et désérialisation de données
- Gestion des entrées/sorties pas les flux

Accès aux bases de données :

- Création et utilisation d'un modèle de données Entité
- Interrogation de données par LINQ
- Mise à jour de données par LINQ

Accès à des données distantes :

- Accès à des données à travers le Web
- Accès à des données dans le Cloud

Concevoir l'interface utilisateur d'une application graphique :

- Utiliser le XAML pour concevoir une interface utilisateur
- Liaison de contrôles aux données
- Définir le style d'une interface utilisateur

Améliorer les performances et la réactivité d'une application :

- Implémentation du multitâche en utilisant tâches et expressions Lambda
- Réaliser des opérations asynchrones
- Synchronisation d'accès concurrents sur les données

Intégration de code non managé :

- Création et utilisation d'objets dynamiques
- Gestion de la durée de vie des objets et contrôle des ressources non managées

Création de types réutilisables et d'assemblées :

- Examiner les métadonnées des objets
- Créer et utiliser des attributs personnalisés
- Génération du code managé
- Versioning, signature et déploiement d'assemblées

Cryptage et décryptage de données :

- Implémentation de cryptage symétrique
- Implémentation de cryptage asymétrique