

Bonnes pratiques .NET et Design Patterns : Travaux pratiques - 4 jours

Créer des applications réussies avec des techniques reconnues

formation 511

- Vous apprendrez à**
- Mettre en œuvre des méthodes éprouvées pour créer des applications Web et client .NET souples, fiables et efficaces
 - Résoudre des problèmes de programmation grâce aux design patterns standard
 - Produire un code sans erreur en utilisant le développement piloté par les tests et Visual Studio 2008
 - Créer des architectures multiniveaux réutilisables et éviter les chevauchements fonctionnels
 - Améliorer la conception de classes grâce aux bonnes pratiques
 - Simplifier et automatiser des tâches répétitives

Objectif Posséder des connaissances des différents langages et bibliothèques .NET n'est malheureusement pas suffisant pour développer des applications robustes. Les pratiques éprouvées et les design patterns, élaborés à partir de l'expérience des experts, vous aident à construire des applications sur des fondations solides. Ce cours vous apporte les compétences dont vous avez besoin pour résoudre les problèmes concrets du développement logiciel et produire des applications rapides et fiables.

À qui s'adresse cette formation Programmeurs, architectes système et tous ceux qui développent des applications .NET. Une expérience de la programmation du niveau de la formation 419, "Programmation en C#", ou de la formation 503, "Programmation avec Visual Basic pour .NET", est supposée acquise.

Travaux pratiques Vous acquerez une expérience pratique de la mise en œuvre des bonnes pratiques .NET et des design patterns. Les exercices, réalisés en VB ou C#, incluent :

- Réconciliation de divers problèmes de traitement avec le pattern Strategy
- Simplification d'un système complexe avec le pattern Façade
- Utilisation de l'Entity Framework de Microsoft pour accéder aux données et les mettre à jour
- Création d'un modèle objet du domaine
- Construction de tests automatisés
- Création d'une interface utilisateur testable avec le pattern Model View Controller
- Capture des tests avec Visual Studio

Bonnes pratiques .NET et Design Patterns : Travaux pratiques - 4 jours

Créer des applications réussies avec des techniques reconnues

formation 511

Introduction

- Création de code grâce aux bonnes pratiques
- Simplification des logiciels grâce aux design patterns

Simplification de la programmation d'applications complexes grâce à des Design Patterns éprouvés

Application d'interfaces simples à des algorithmes complexes

- Unification des interfaces à partir de sous-systèmes pour faciliter l'utilisation
- Mise en œuvre du pattern Façade

Modulation des fonctionnalités par programmation sur les interfaces

- Amélioration de la flexibilité et de la capacité d'adaptation de votre application
- Utilisation du pattern Strategy

Altération dynamique du comportement d'un objet

- Extension des fonctions sans impact sur le code existant
- Invocation du pattern Decorator

Réutilisation et flexibilité

- Définition d'un schéma de base pour un algorithme
- Utilisation du pattern Template Method

Interfaçage de classes incompatibles

- Transformation d'une interface pour ajouter de la valeur à du code existant
- Exploitation du pattern Adapter

Application des techniques de développement piloté par les tests

Automatisation du test par unités

- Réduction des cycles de développement grâce aux tests automatisés
- Amélioration de la qualité grâce aux tests
- Suppression des erreurs grâce aux tests réutilisables

Intégration des tests et de l'écriture du code

- Conception de code en s'aidant de la programmation dirigée par les tests
- Génération immédiate de notifications pour une meilleure qualité du code et des cycles de développement courts
- Organiser, coordonner et effectuer des cas de tests avec Visual Studio

Architecture d'une application multineveau

Conception de l'architecture d'une application

- Architectures en couches pour faciliter la réutilisation, l'évolutivité et la longévité
- Accès aux données à partir de la couche métier
- Découpler l'instanciation des objets grâce au pattern Factory
- Conserver l'identité des objets avec le pattern Identity Map

Programmation des niveaux de l'application

- Structure d'une application "client Web" ou "client riche" avec le pattern Observer
- Gestion des états d'une application avec le pattern State
- Restructurer des tables d'une base de données sans affecter le code de l'application

Modélisation du niveau métier

- Relâchement du lien entre les modèles objets métier et la base de données avec le pattern Domain Model
- Mappage entre des objets métier riches et des tables de bases de données grâce au pattern Data Mapper
- Réduction des hiérarchies d'objets métier avec le pattern Inheritance Mapper
- Exploiter la programmation déclarative dans l'Entity Framework de Microsoft pour implémenter des classes Data Mapping

Mise en œuvre des bonnes pratiques

Organisation et implémentation de la logique métier

- Application des patterns logiques de domaine au niveau intermédiaire
- Extension des types Entité grâce aux classes partielles
- Manipulation de groupes d'entités avec des classes métier

Bonnes pratiques de conception de classes

- Éviter la rigidité grâce au principe ouvert/fermé
- Utilisation efficace de l'Inheritance

Automatisation des tâches répétitives

Simplification du code d'accès aux données

- Diminution du code d'accès aux données grâce aux propriétés de navigation des entités

- Suppression du code de mise à jour de la base avec le suivi des modifications des entités

Amélioration du code de l'application

- Automatisation de la revue de code avec FxCop et Visual Studio Analysis
- Élimination du code dupliqué avec la refactorisation en design patterns