

Stage pratique
5 jours

VBN

Visual Basic.NET, développer des applications avec Visual Studio 2005

Dates des sessions

Paris

10 décembre 2007
21 janvier 2008
18 février 2008
10 mars 2008
31 mars 2008
21 avril 2008
19 mai 2008
16 juin 2008

Lyon

17 décembre 2007
7 avril 2008
9 juin 2008

Aix

17 décembre 2007
21 avril 2008
23 juin 2008

Nantes

17 décembre 2007
7 avril 2008
23 juin 2008

Rennes

17 décembre 2007
7 avril 2008
23 juin 2008

Lille

22 octobre 2007

Bruxelles

22 octobre 2007
3 décembre 2007
19 mai 2008

Prix 2007

1990 euros HT

OBJECTIFS

Cette formation vous apprendra tout d'abord à connaître l'architecture du Framework 2.0 et à exploiter les classes de base, communes aux différents types d'applications .NET (Windows, Web, DLL, etc.), avec le langage VB.NET. Vous apprendrez ensuite à créer vos classes/objets spécifiques et à tirer parti des mécanismes (encapsulation, héritage et polymorphisme) inhérents à la programmation orientée objet (POO) en .NET.

[Introduction à la plate-forme .NET](#)

[Syntaxe de base : données, expressions et instructions](#)

[Programmation Orientée Objet](#)

[Classes et objets en VB.NET](#)

[Interfaces](#)

[Gestion des exceptions](#)

[Les assembly](#)

[Objet et classes de base du framework .NET](#)

[Gestion des entrées/sorties](#)

[Applications développées avec le framework .NET](#)

Participants

Ce cours s'adresse aux développeurs désirant acquérir les techniques et les réflexes pour développer tous types d'applications VB.NET.

Pré-requis

Expérience dans le développement d'applications avec Visual Basic, Delphi ou en langage C. La connaissance de Visual Basic est appréciable mais n'est pas nécessaire pour suivre cette formation. Le stage réf UNE est également fortement recommandé pour les développeurs qui ne sont pas familiarisés avec la POO.

Travaux pratiques

50 % d'ateliers sont prévus dans ce cours avec le Framework .NET 2.0 et Visual Studio 2005. Dans la première partie du stage, les aspects syntaxiques du langage VB .NET sont illustrés par des exemples simples, permettant de manipuler les classes de base (nombres, chaîne de caractères, formatage, manipulation du système de fichiers, de fichiers texte, etc.). La deuxième partie concerne la conception d'objets spécifiques, puis la création d'une DLL réutilisable, avec son déploiement. Le stage se termine par des démonstrations rapides sur les accès aux données (ADO .NET) et les applications de type Windows et Web (ASP .NET).

CONTENU

[Introduction à la plate-forme .NET](#)

- Principe et architecture de l'environnement .NET.
- Composants du framework .NET : langage, CLR, CLI.
- Architecture de distribution d'objet dans .NET, exécution de composants et d'objets à distance.
- Structure d'un programme VB.NET : notion d'espace de noms, exemple d'entrée/sortie.
- Présentation du framework .NET.
- Outils et environnement de développement.
- Langage de MSIL : principe du langage intermédiaire, principe de la compilation JIT (Just In Time Compiler).

Travaux pratiques

Exemple de programme VB.NET minimum. Exécution en mode géré. Utilisation de l'environnement de développement Visual Studio.Net pour l'écriture du premier programme.

[Syntaxe de base : données, expressions et instructions](#)

- Valeurs, opérateurs, manipulation de variables.
- Types de données : Common Type System.
- Données complexes : tables. Gestion des tables, création et manipulation.
- Instructions de contrôle de flux : boucles, test, contrôle de flux.
- Fonctions.
- Déroulement d'un programme, notion d'exception.
- Petites nouveautés : Continue, IsNot, Using, visibilité des propriétés, support des types non signés, mot-clef Global, TryCast.
- Opérateurs IsTrue et IsFalse.
- Génération de la documentation.

Travaux pratiques

Ecriture de programmes en VB.NET mettant en oeuvre des algorithmes classiques.

Programmation Orientée Objet

- Classes et objets : modélisation du monde réel par objet.
- Notions de champ, méthode et propriété.
- Héritage.
- Polymorphisme.
- Implémentation d'interfaces multiples.
- Représentation du modèle objet.

Classes et objets en VB.NET

- Définition des classes. Définition des objets.
- Définition du contenu de la classe : méthodes et attributs.
- Surchage des méthodes. Surchage des opérateurs. Incidences sur les transtypages.
- Cycle de vie des objets : constructeur, destructeur.
- Typage en VB.NET : gestion des types et conversion de types. Interopérabilité entre langages.
- Propriétés : définition des propriétés, définition de propriétés « read only ».
- Visibilité des membres d'une classe : propriétés et méthodes. Utilisation des espaces de noms.
- Dérivation et héritage des classes : principe de dérivation, contrôle d'accès lors de la dérivation. Héritage des propriétés.
- Classe de base System.Object.
- Manipulation des attributs : principe des méta-données. Attributs de classe, de méthode, de champ. Les attributs et l'héritage.
- Manipulation et gestion des tableaux. Gestion de la mémoire. Utilisation du garbage collector.
- Indexeur et structure foreach : principe et mise en oeuvre.
- Classes partielles.
- Gestion personnalisée des événements.
- Classes génériques.
- Types annulables.
- Utilisation des délégués pour la gestion des événements.

Travaux pratiques

Ecriture de classes de base. Manipulation de méthodes et propriétés dans les classes. Dérivation de classe. Utilisation des attributs dans les classes VB.NET. Utilisation des indexeurs et tableaux.

Interfaces

- Principe des interfaces.
- Déclaration et mise en oeuvre d'une interface.
- Les interfaces et l'héritage.
- Principe héritage multiple au travers des interfaces.

Travaux pratiques

Ecriture de programmes mettant en place des interfaces.

Gestion des exceptions

- Principe de la gestion des exceptions.
- Utilisation des exceptions pour le traitement centralisé des erreurs.
- Utilisation des exceptions pour simplifier la gestion des ressources.
- Structure de gestion des exceptions interlangages : déclenchement d'exception dans un langage, interception et traitement dans un autre. Exemple entre VB.NET et C#.

Travaux pratiques

Gestion des exceptions en VB.NET. Exemple de gestion des exceptions interlangages.

Les assembly

- Notion d'assembly. Création d'un assembly. Organisation d'un projet au travers des assembly.
- Assembly privé et assembly partagé : signature, rôle du GAC (Global Assembly Cache), mise en place dans le GAC.
- Assembly à signature retardée.
- Assembly et support du multilingage.

Travaux pratiques

Organisation du développement avec des assembly. Création d'assembly public. Utilisation de sn (Strong Name) pour la signature, mise en place dans le GAC.

Objet et classes de base du framework .NET

- Qu'est-ce qu'un framework ? Principe.
- Le CTS, classes de basiques (Int32, String...), correspondance en C#.
- Classes incontournables : Math, Random...
- Traitement des dates et des durées. Traitement des chaînes avec StringBuilder et les expressions régulières.
- Les différents types de collections, de dictionnaires et de tables de hachage.
- Les collections génériques.

Travaux pratiques

Utilisation des classes de base. Utilisation des types en tant que classe de l'environnement .NET.

Gestion des entrées/sorties

- Hiérarchie des classes.
- FileStream et StreamReader/StreamWriter.
- Manipulation du système de fichier.
- Entrées/Sorties asynchrones.

Travaux pratiques

Utilisation des classes de .NET pour faire des entrées/sorties.

Applications développées avec le framework .NET

- Eléments essentiels du framework : interface utilisateur avec les Windows Form, ADO.NET, ASP.NET, services Web.
- Développement multicible : périphériques .NET, cibles de développement.
- Architecture des applications avec les Web Services.
- Exemple de programmes utilisant les composants essentiels du .NET.
- Autres domaines de .NET : intégration des Web services.

Travaux pratiques

Exemple d'application avec une interface graphique. Utilisation des Windows.Forms.