

Introduzione

Scrivere un manuale strutturato in centinaia di pagine è una grande impresa, ma basta procedere gradualmente: “un viaggio di mille miglia inizia con un singolo passo”. Il mio viaggio è la redazione di pagine, il vostro è l'apprendimento di un linguaggio. A qualcuno può sembrare di una difficoltà insormontabile. Sono sicuro, invece, che è possibile compiere questo viaggio insieme, con successo.

Panoramica

Visual Basic .NET è molto più di Visual Basic 7. Microsoft ha rinnovato quasi completamente un linguaggio che milioni di sviluppatori conoscono e amano, Visual Basic (VB). Probabilmente le ragioni che hanno portato a modificare VB così significativamente richiedono un manuale specifico, ma è un argomento che lascio meglio trattare a Microsoft. Alcune cose sono state chiaramente lasciate nelle versioni passate di VB. Queste lacune potrebbero non apparire evidenti agli amanti di VB, ma sono un dato di fatto. Al termine della lettura, le nuove potenzialità e funzionalità di Visual Basic .NET vi saranno evidenti.

Questo è un manuale completo sui fondamenti della programmazione in un linguaggio totalmente orientato agli oggetti, completo di nuovi idiomi e possibilità, come l'ereditarietà e il polimorfismo, la gestione degli eventi, la gestione strutturata delle eccezioni e i servizi Web. Oltre alla presentazione dei fondamenti, il compito di questo manuale è dare istruzioni supplementari e consigli basati sull'esperienza reale, al fine di aiutare a diventare più produttivi con Visual Basic .NET di quanto non lo si sia mai stati con Visual Basic 6.

A chi è destinato questo manuale

Questo libro è stato scritto per aiutare tanto il neofita quanto il programmatore professionista a compiere il passaggio a un linguaggio profondamente rinnovato, fornendo una serie completa di suggerimenti per le esigenze di programmazione.

Questo manuale è adatto a tutti: al programmatore Visual Basic 6 (o versioni precedenti), al programmatore novizio o a chi proviene da un altro ambiente di sviluppo. Si troveranno informazioni che variano dai fondamenti della programmazione a oggetti (OOP) ad argomenti ed esempi che mostrano alcuni dei più avanzati idiomi orientati agli oggetti e alla programmazione avanzata .NET.

Ho con attenzione codificato e organizzato esempi reperibili sui seguenti siti Web: www.sams-publishing.com e www.softconcepts.com. Ho dedicato un certo tempo per incorporare uno degli aspetti migliori dell'Extreme Programming Refactoring (meglio conosciuto come XP). Come spesso accade, ho utilizzato il refactoring (tecnica di riscrittura del codice) al fine di elaborare esempi di codice orientato agli oggetti, semplice, di alta qualità e che sia possibile riutilizzare per risolvere alcuni problemi. Gli esempi di codice non assomiglieranno a codice VB6: assomiglieranno a codice Visual Basic .NET. I miei sforzi sono orientati ad aiutare il passaggio da VB6 e a impiegare Visual Basic .NET al livello più esteso possibile. Ho utilizzato il refactoring di Martin Fowler come riferimento per la riscrittura del codice.

Già dopo la lettura dei primi capitoli sarà possibile costruire con successo applicazioni in Visual Basic .NET. A lettura conclusa, sarà possibile impiegare le funzionalità più potenti di Visual Basic .NET in maniera professionale e completa. Se l'obiettivo è elaborare soluzioni professionali in Visual Basic .NET, questo manuale è sicuramente lo strumento giusto.

Visual Basic .NET è un linguaggio di prima classe

Quando ho assistito all'incontro degli autori Microsoft nel novembre 2000, le persone chiave di Microsoft si riferivano a Visual Basic .NET come a un linguaggio "di prima classe". Questa consapevolezza di Microsoft che c'erano spazi di miglioramento nel vecchio VB era un grande passo avanti. Visual Basic .NET è davvero un linguaggio di prima classe.

Visual Basic .NET ora include caratteristiche che sono state disponibili per anni ai programmatori C++, Java e Delphi. Chi di noi ha programmato in questi altri linguaggi, ma era innamorato di VB, ha pazientemente aspettato le modifiche che avrebbero permesso di uscire dall'ambito delle DLL per andare oltre i limiti che sicuramente affliggevano VB6. Con l'arrivo di Visual Basic .NET, quelle limitazioni sono superate. Il mio lavoro è mostrare le caratteristiche di quelle limitazioni mentre espongo le nuove peculiarità e funzionalità di Visual Basic .NET.

Visual Basic .NET è un linguaggio di prima classe perché supporta e contiene funzionalità come: ereditarietà, metodi virtuali, metodi condivisi, gestione degli eventi, gestione strutturata delle eccezioni, multithreading e incapsulamento a livello namespace. L'ereditarietà in Visual Basic .NET significa ereditare metodi e proprietà e non interfacce. La gestione degli eventi è implementata attraverso delegate e classi speciali che permettono di scrivere dinamicamente i gestori di evento oltre al supporto dei form. I metodi condivisi, spesso chiamati metodi di classe o metodi statici, supportano il richiamo dei metodi al livello di metaclassa. Sono supportati anche il multithreading, i namespace e i costruttori con parametri. Si noterà che le applicazioni scritte in Visual Basic .NET sono limitate più dall'immaginazione che dal linguaggio stesso o dall'ambiente .NET.

Che cos'è il Common Language Runtime

In chiave marketing, il Common Language Runtime (CLR) viene presentato come una sostituzione di tecnologie pesanti come COM, MTS, VB Runtime DLL e altre ancora. In pratica, il CLR mostra che Microsoft ha fatto quello che noi tutti avremmo voluto fare, cioè sfruttare il corpo di un codice rendendolo riutilizzabile in molti contesti.

Il CLR è un'architettura comune di classi e codice che è condivisa tra tutti i linguaggi sotto la struttura .NET. Questo significa che gran parte del corpo base di un codice costruito da Visual Basic .NET è condiviso tra C# (C sharp), Visual C++ e altri linguaggi per essere ancora implementato per .NET.

Il CLR è essenziale per il gemellaggio tra Visual Basic .NET e altri linguaggi .NET, oltre a rendere le applicazioni Visual Basic .NET molto più semplici ed efficienti.

Perché passare a Visual Basic .NET

Non è facile spiegare in poche righe i motivi per cui passare a Visual Basic .NET. Fra i motivi principali ci sono sicuramente i WebForm, le applicazioni Web, i servizi Web, il CLR, la gestione delle eccezioni strutturata, la programmazione a oggetti e il multithreading.

Al termine della lettura, i vantaggi della migrazione vi saranno molto più chiari di quanto sia possibile spiegare in questo paragrafo. L'importante è capire che i motivi per passare a VB .NET sono reali e tangibili e hanno poco o niente a che fare con il marketing. Se questo manuale è stato costruito correttamente, ogni lettore sarà in grado di fornire una prova chiara di tali vantaggi e potenzialità a tutti i propri interlocutori (manager, colleghi e clienti).

Implicazioni

Le applicazioni scritte nelle versioni precedenti di VB dovranno essere riscritte completamente. Vale la pena di osservare che Microsoft ha incluso uno strumento di aggiornamento. All'avvio di applicazioni VB6 in Visual Basic .NET, parte automaticamente un wizard di aggiornamento. Quanto bene questo lavorerà è da verificare di volta in volta. Eseguita una semplice applicazione "HelloWorld" scritta in VB6, l'installazione guidata ha prodotto molto codice supplementare per supportare la compatibilità. (L'aggiornamento di "HelloWorld" e l'applicazione originale sono disponibili per il download all'indirizzo: www.sampublishing.com).

A parte fornire assistenza sull'aggiornamento da VB6, Microsoft ha già dichiarato la modesta compatibilità all'indietro di Visual Basic .NET, rendendo più concreta la necessità di riscrivere completamente le applicazioni VB6. Il vantaggio di riscrivere è che sarà possibile utilizzare VB .NET su un'applicazione già nota, ricostruendola con una codifica più snella. Lo svantaggio della semplice conversione da VB6 sarà quello di avere un'applicazione appesantita da parti di codice generate dalla conversione stessa.

La necessità di riscrivere applicazioni esistenti può disincentivare l'adozione di Visual Basic .NET. Tuttavia, i vantaggi saranno tangibili sia per il cliente che per il programmatore, ricompensando al meglio lo sforzo.

Contenuto del libro

Il libro prende in considerazione il nuovo linguaggio Visual Basic .NET, Visual Studio IDE, la programmazione Web con WebForm, ADO.NET, i servizi Web, GDI+ e molti altri argomenti. Visual Basic .NET è uno strumento aziendale. Per dimostrare le potenzialità dello strumento nel contesto di Visual Basic .NET, verranno utilizzati modelli (come l'“Automation Extensibility Model”) che permettono di personalizzare la propria esperienza in Visual Studio.

Per aiutare a ottenere il maggiore vantaggio possibile dalla programmazione in un linguaggio orientato agli oggetti, dal Refactoring e dall'Extreme Programming (XP), verrà impiegato codice riutilizzabile nelle applicazioni reali.

Infine, tutti i listati, dal più banale al più complesso, sono disponibili per il download all'indirizzo www.sampublishing.com (gli esempi di codice sono organizzati per capitolo).

Il nuovo linguaggio Visual Basic

Alcune parole chiave e caratteristiche di base di Visual Basic si ritrovano in Visual Basic .NET, ma è qui che la maggior parte delle somiglianze termina. Nel libro presenterò dettagliatamente tutte le novità.

La programmazione a oggetti avanzata

Le classi in Visual Basic 6 supportano l'ereditarietà di interfaccia. L'ereditarietà di interfaccia è ortogonale all'ereditarietà orientata agli oggetti. Con le classi in VB6 vengono definiti i metodi pubblici e gli attributi in una classe e quindi realizzati in un'altra classe. Questo approccio funziona abbastanza bene ma non è così potente come la relazione di superclasse e di sottoclasse dell'ereditarietà orientata agli oggetti.

In Visual Basic .NET si ottengono metodi e attributi dalla superclasse o dalla classe parente e li si ampliano nelle sottoclassi. Il vantaggio è un grado più elevato di riutilizzo del codice. La nomenclatura per OOP in Visual Basic .NET e la grammatica che li supporta sono differenti. In questo testo sono presenti esempi completi di nomenclatura orientata agli oggetti.

Oltre ai termini OOP fondamentali, questo manuale analizza e dimostra l'effetto che revisioni OOP hanno in Visual Basic .NET sulla progettazione delle interfacce, utilizzando namespace, definendo classi, realizzando delegate, adottando l'ereditarietà, il polimorfismo, l'incapsulamento e i metodi condivisi. Tutte le discussioni teoriche sono supportate da esempi concreti.

Dopo aver definito nuovi termini OOP, il libro prosegue esponendo concetti avanzati come la riflessione, gli attributi, il multithreading e la programmazione Web.

Common Language Runtime (CLR)

Saranno descritte molte classi nell'ambito del CLR, tra cui la classe estesa `System.Diagnostics` che garantisce ampie facoltà di verifica e di correzione. Altri aspetti del CLR saranno analizzati all'occorrenza.

Sviluppo client/server

La programmazione client/server è ancora un caposaldo della programmazione in senso lato. Un accenno alla programmazione ADO.NET può essere trovato nel Capitolo 19, che tratta di applicazioni ASP.NET e Web.

La programmazione di database ha bisogno di un'estesa trattazione e richiederà un libro a parte.

Programmazione per il Web

Visual Basic .NET è una scelta eccellente per la programmazione per il Web. ASP.NET utilizza moduli code-behind che possono essere implementati con lo stesso Visual Basic .NET come utilizzato in applicazioni Windows. Il Capitolo 18 tratta i servizi Web e il Capitolo 19 introduce le applicazioni Web e i controlli server ASP.NET. Peraltro, il controllo esteso della programmazione per il Web è un argomento che necessita di un manuale a sé stante.

