

Introduction

It has been my experience as a Visual Basic trainer that most people do not have trouble picking up the syntax of the language. What perplexes and frustrates many people are the higher-level concepts of object-oriented programming methodology and design. To compound the problem, most introductory programming books and training classes skim over these concepts or, worse, do not cover them at all. It is my hope that this book fills this void. My goal in writing this book is twofold. First, to provide you with the information needed to understand the fundamentals of programming with Visual Basic. Second and more importantly, to present you with the information required to master the higher-level concepts of object-oriented programming methodology and design.

This book provides you with the information needed to understand how you go about architecting an object-oriented programming solution aimed at solving a business problem. As you work your way through the book, first you will learn how to analyze the business requirements. Next, you will model the objects and relationships involved in the solution design. Finally, you will implement the solution using Visual Basic .NET. Along the way, you will learn the fundamentals of software design, the Unified Modeling Language (UML), object-oriented programming, Visual Basic (VB), and the .NET Framework.

Because this is an introductory book, it is meant to be a starting point for your study of the topics presented. As such, this book is *not* designed to make you an expert in object-oriented programming and UML; nor be an exhaustive discussion of VB and the .NET Framework; nor be an in-depth study of Visual Studio. It takes considerable time and effort to become proficient in any one of these areas. It is my hope that by reading this book, your first experiences in object-oriented programming will be enjoyable, comprehensible, and instill a desire for further study.

Target Audience

The target audience for this book is the beginning VB programmer who wants to gain a foundation in object-oriented programming along with the VB language basics. Programmers transitioning from a procedural-oriented programming model to an object-oriented model will also benefit from this book. In addition, there are many pre-.NET VB programmers who do not have a firm grasp of object-oriented programming. Now is the time to become acquainted with the fundamentals of object-oriented programming before transitioning to the current version of VB and the .NET Framework. Because the experience level of a “beginner” can vary immensely, I have included a primer in Appendix A, which discusses some basic programming tenets. I would suggest you review these concepts if you are new to programming.

Organization of the Book

This book is organized into three parts:

Part 1 delves into object-oriented programming methodology and design—concepts that transcend a particular programming language. The concepts presented are important to the success of an object-oriented programming solution regardless of the implementation language chosen. At the conclusion of this part, a case study walks you through modeling a “real-world” application.

Part 2 looks at how object-oriented programming is implemented in Visual Basic. You will look at creating class structures, creating hierarchies, and implementing interfaces. This part also introduces object interaction and collaboration. You will see how the object-oriented programming topics discussed in Part 1 are transformed into Visual Basic coding constructs.

Part 3 returns to the case study introduced and modeled at the end of Part 1. Using the knowledge gained in Part 2, you will transform the design into a fully functional VB application. This includes designing a graphical user interface, implementing the business logic, and integrating with a relational database to store data. Along the way you will be exposed to the .NET Framework classes used to work with data, and see how to create a Windows-based user interface, a Web-based user interface, and a Web service-based programmatic interface.

Activities and Software Requirements

One of the most important aspects of learning is doing. You cannot learn to ride a bike without jumping on a bike, and you cannot learn to program without “cranking out” code. Any successful training program needs to include both a theory component and a hands-on component. I have included both components throughout this book. It is my hope that you will take these activities seriously and work through them thoroughly and even repeatedly. Contrary to some students’ perception that these activities are “exercises in typing,” this is where the theory becomes concrete and true simulation of the concepts occurs. I also encourage you to play during the activities. Do not be afraid to alter some of the code just to see what happens. Some of the best learning experiences occur when students “color outside the lines.”

You can download the starter files referred to in this book from the Apress Web site at www.apress.com. The UML modeling activities in Part 1 are for someone using Objectteering’s UML Modeler. I chose this program because of its simple user interface and the fact it can be downloaded for free at www.objectteering.com. You do not need a CASE tool to complete these activities; a paper and pencil will work just fine. You can also use another CASE tool such as Visio to complete the activities. The activities in Part 2 require Visual Studio 2005 with Visual Basic installed. I encourage you to install the help files and make ample use of them while completing the activities. The activities in Part 3 require Microsoft SQL Server 2000 or 2005 with the Pubs and Northwind databases installed. Appendix C includes instructions on downloading and installing the sample databases. You can find a trial edition of both Visual Studio 2005 and SQL Server 2005 at www.msdn.microsoft.com.

Note The web addresses mentioned are subject to change without notice. Check the Apress site (www.apress.com) for any updates.
