Book Review:

C# for Programmers (Second Edition)

Harvey M. DEITEL and Paul J. DEITEL
Prentice Hall, 2006
ISBN 0131345915

C# was designed specifically for the .NET platform with the purpose –among others– of facilitating the migration of applications to the new Microsoft platform. This language was developed by A. Helsjberg and S. Wiltamuth, and has its roots in C, C++, and Java.

C# for Programmers (Second Edition) focuses on explaining the characteristics of this language through the development of desktop applications –and in some cases, web forms– with the step-by-step technique, developing study cases such as:

- Basic application for an automatic teller machine (ATM)
- Payroll application
- Guestbook application

Concepts are developed in 26 chapters supported by 9 appendixes included in one CD.

Chapters 1 and 2 present basic concepts of object-oriented programming (classes, objects, attributes and behavior). A brief history of Windows, the Internet and C# by means of a simple visual application using pre-existing controls is included, and an introduction to Visual C# 2005 Express Edition's IDE is made.

In Chapter 3, the basic characteristics of C# for the development of a console application by using the CSC compiler are presented.

Chapter 4 deals in detail with the topics of objects, classes, attributes, and methods. The reader learns to define instance variables (private) and properties (public) to set and get these data. UML is presented for the creation of class diagrams.

In Chapters 5 and 6, control structures *if, if… else, switch, while, do… while, for, foreach, break, continue* are presented. Simple algorithms are used to help understand them.

In Chapter 7, the differences between *static* methods are presented using the Math class of .NET (that provides *static* methods) to solve calculus problems. The definition of constant values is presented with *const* and *enum*. *Ref* and *out* are presented for methods to use arguments as reference.

In Chapters 8, 24, 25, and 26, different data structures, arrays (static); lists, queues, stacks, and trees (dynamic); generics (used with different types); and the structure class that defines .NET, are presented.

In Chapters 9, 10, and 11, object-oriented concepts are described in detail in relation to legacy, polymorphism, operator interfaces and overload, and the use of *this, readonly, internal, static* and *sealed* classes.

In Chapter 12, the use of exceptions to manage application errors is explained. The reader learns to use the *try… match… finally* block. The *Exception* hierarchy of .NET, including *ApplicationException* (which allows user-defined exceptions derivation) and *SystemException* is presented.

In Chapters 13 and 14, the concepts of graphic user interface (GUI) using event-oriented programming in Windows Forms are presented. The reader learns to understand IDE-generated code. The components used are, among others: *Button, Label, RadioButton, CheckBox, TextBox, Panel, ListView, ListBox, ComboBox,* and *DateTimePicker.*

In Chapter 15, process management is described by creating and running *Thread* objects, and results are displayed on a GUI. Examples for synchronizing concurrent processes (*Monitor* class) with traditional producer/consumer algorithms are presented.

In Chapter 16, strings and normal expressions defined in the .NET hierarchy are described. The *StringBuilder* class is used to show how to handle a string. The *Regex* and *Match* classes are used to show how to search for patterns, relate, and split chains.

In Chapter 17, an introduction to .NET drawing tools (graphics and multimedia) is made, and Windows Media Player is incorporated to an application for adding animated characters as well as to Microsoft Agent for adding speech synthesis and recognition capabilities.

In Chapter 18, the reader learns to work with persistent data and files. The *File* (file handling) and *Directory* (directory handling) classes are used.

In Chapter 19, XML is presented using .NET (DOM) classes for data recovery and modification. A method for validating XML documents through DTD is also presented.

In Chapter 20, the relational data model is described, and the reader will learn how to use classes to handle databases with the ADO.NET offline model, as well as to create XML documents based on relational data. An address book application is built.

Chapters 21 and 22 deal with ASP-NET and Web services with Visual Web Developer 2005 Express. Client, server and business logic levels are presented. The relations between ASPX files and code are shown. The reader learns to work with sessions (*HttpSessionState*) to maintain user status across different pages. Web services allow methods from one machine to call methods from other machine (using XML, http and SOAP).

In Chapter 23, applications that allow working with sockets (flows and datagrams) implementing clients and servers are presented. An example is provided by means of the *WebBrowser* control to add Web browsing to any application.

This book is useful for those readers who want to get a deeper knowledge of the new .NET technology by means of specific, step-by-step examples.

Hugo Ramón
hramon@lidi.info.unlp.edu.ar