Visual Basic .NET Class Design Handbook: Coding Effective Classes

This book takes a top-down look at exactly what makes up a class in .NET. It begins by defining a type and exploring how classes relate to the .NET type framework. It then examines what makes up types: type members. The majority of the book is devoted to looking at the different mechanisms VB .NET provides for defining type members (methods, constructors, properties, and events) and examining how types combine to make up assemblies.

Summary of Contents

Introduction

Chapter 1: Defining Types

Chapter 2: Type Members

Chapter 3: Methods

Chapter 4: Constructors and the Object Lifecycle

Chapter 5: Properties

Chapter 6: Events and Delegates

Chapter 7: Inheritance and Polymorphism

Chapter 8: Code Organization and Metadata

Appendix A: Support, Errata, and forums.apress.com

Index

Visual Basic .NET Class Design Handbook: Coding Effective Classes

Damon Allison
Andy Olsen
James Speer of Charteris plc

APress Media, LLC

Visual Basic .NET Class Design Handbook: Coding Effective Classes

Copyright © 2003 by Damon Allison, Andy Olson, and James Speer

Originally published by Springer-Verlag New York, Inc. in 2003

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN 978-1-59059-275-5 ISBN 978-1-4302-0780-1 (eBook)

DOI 10.1007/978-1-4302-0780-1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at <http://www.apress.com> in the Downloads section.

Credits

Additional Material

Steven Sartain

Commissioning Editor

James Hart

Editorial Board

Dan Appleman

Craig Berry

Gary Cornell

Tony Davis

Steven Rycroft

Julian Skinner

Martin Streicher

Jim Sumser

Karen Watterson

Gavin Wray

John Zukowski

Managing Editor

Jan Kolasinski

Project Manager

Beckie Stones

Technical Editors

Benjamin Hickman

Andrew Polshaw

Cover

Kurt Krames

Technical Reviewers

Richard Bonneau

Damien Foggon

Mark Horner

Anthony Naylor

Phil Powers-DeGeorge

David Schultz

Erick Sgarbi

Imar Spaanjaars

Production & Layout

Sarah Hall

Natalie O'Donnell

Index

Adrian Axinte

Andrew Criddle

Proofreader

Chris Smith

About the Authors

Andy Olsen

Andy is a freelance consultant engaged in training, consultancy, and development work in Microsoft .NET and related technologies. Andy studied Physics at Southampton University in England, and began his professional life as a C developer. As the 1990s came and went, Andy migrated into C++, Visual Basic, Java, and OO Analysis and Design using UML. Andy has been using Microsoft development tools and technologies since 1987, and has fond memories and many tall stories to tell of times gone by.

Andy now lives by the sea in Swansea, with his wife Jayne, and their children Emily and Thomas. Andy is a keen football and rugby supporter, and enjoys running and skiing (badly). You can reach Andy at andyo@olsensoft.com.

Damon Allison

Damon is an IT consultant in Minneapolis, MN focusing on implementing Microsoft technologies. Arguably, Damon has a life outside of programming. He enjoys playing golf and has high hopes someday he'll be good at it.

Dad, I miss you, pal.

James Speer of Charteris plc

James Speer has worked in software development since 1987. Beginning his career with BCPL and C++, James has more recently specialized in distributed middle-tier development using Visual Basic, XML, MSMQ, and SQL Server. James is currently employed by Charteris plc (www.charteris.com) as a Senior Developer providing prescriptive guidance, mentoring, and development expertise for Charteris clients.

Thanks to Lucy for the smiles, Kate for the coffee and to the beautiful game of Football for giving me a life outside of work.

Steven Sartain

Steven Sartain is a database applications developer. He has worked on a number of large .NET applications over the past fourteen months concentrating his efforts in controls development. He is currently working on a product called Visual Genie; designed to integrate tightly into the .NET IDE, Visual Genie will remove the coding burden of developing n-tiered applications. When not spending long hours lonely with his computer, he's spending time with his kids, or enjoying racing around country roads in his car.

I would like to thank my two daughters Nikita and Jade for understanding their fatherless periods whilst working on the text; my wife, Donna, for throwing food in at me to keep me alive; and finally my friend Paul Nichols, for putting up with me.

VB.NET

Class Design

Handbook

11

Table of Contents

Introduction	1
Who Is This Book For?	1
What Does This Book Cover?	2
What Doesn't It Cover?	2
What Will You Learn?	2
What Do You Need?	3
Chapter 1: Defining Types	7
Types	8
The .NET Type System	9
Value Types and Reference Types	10
Primitive Types	13
Viewing the Output from the Compiler	16
User-Defined Value Types (Structures)	19
Defining and Using Value Types	19
Using Inheritance with Value Types	21
Boxing and Unboxing Value Instances	24
Enumerations	25
Class Types	26
Defining Classes in Visual Basic .NET	27
Modules	29
Defining Visibility for Classes	29
Nested Classes	31
Creating and Disposing of Class Objects	31
Delegates	34
Declaring Delegates	35
Invoking a Method Using a Delegate Object	35
Binding Delegates to Methods	36

Interfaces	37
Declaring Interfaces	38
Defining a Hierarchy of Interfaces	38
Implementing Interfaces	39
Using Interfaces	40
Summary	41
Chapter 2: Type Members	43
About Type Members	44
Modifying Members	45
Type Member Accessibility	47
Shared Type Members	50
Modules	50
Constants	51
Fields	53
Properties	56
Methods	59
Method Overloading	60
Properties vs. Methods	60
Constructors	61
Events	65
The System.Object Contract	66
ToString()	67
Equals()	68
GetHashCode()	70
Value and Reference Semantics and Mutability	72
Summary	73
Chapter 3: Methods	75
VB.NET Method Syntax	75
Functions	76
Subroutines	76
Method Scope and Visibility	76
Scope Example	77
Arguments and Parameters	78
Parameter Directives – ByRef/ByVal	79
Passing Reference Types versus Value Types	80
Passing Value Types ByVal	80
Passing Value Types ByRef	81
Passing Reference Types ByVal	82

Passing Reference Types ByRef	82
ByRef and ByVal Best Practice	83
Passing Objects versus Passing Structures	84
Passing Strings	89
Passing Arrays	90
Passing Parameter Arrays (ParamArray)	92
Passing Enumerated Values (Enums)	93
Method Overloading	94
Optional Parameters	98
Shared Methods	99
Returning Values	101
Invoking Methods	102
Design Considerations	102
Summary	103
 Chapter 4: Constructors and the Object Lifecycle	 105
Object Lifecycle Management	106
Object Instantiation	108
Object Destruction	110
Kinds of Constructor	113
Default Constructor	113
Constructors and Optional Parameters	115
Overloading Constructors	116
Chaining Constructors	119
Constructors and Inheritance	120
Protected Constructors and MustInherit Classes	124
Private Constructors	126
Shared Constructors	130
Copy Constructors and Object Cloning	132
The ICloneable Interface	133
Deserialization	137
XML Serialization	138
Singleton	140
Factory	142
Summary	145
 Chapter 5: Properties	 149
Properties in Visual Basic .NET	149

Scalar Properties	152
Compiling Properties into MSIL	154
Read/Write, Read-Only, and Write-Only Properties	156
Defining Shared Properties	159
Guidelines for Writing Get Procedures	160
Guidelines for Writing Set Procedures	163
Complete Example of Scalar Properties	166
Indexed Properties	170
Indexed Properties in .NET Framework Classes	170
Defining a Single Indexed Property in a Class	171
Defining a Default Indexed Property	175
Using Default Properties in .NET Framework Classes	175
Defining a Default Property in Our Own Classes	175
Defining Overloaded Indexed Properties	180
Defining Indexed Properties with Multiple Keys	184
Summary	188
 Chapter 6: Events and Delegates	 191
Delegates	192
Creating and Using Simple Delegates	192
Creating and Using Multicast Delegates	200
Creating and Using Asynchronous Delegates	209
Events	209
Event Architecture	210
Publishing Events	212
Specifying the Event Signature Explicitly	212
Specifying the Event Signature implicitly by using a Delegate	220
Defining Shared Events	222
Subscribing to Events	223
Defining Event Handlers Statically	225
Defining Event Handlers Dynamically	236
Summary	245
 Chapter 7: Inheritance and Polymorphism	 247
Inheritance and Polymorphism Concepts	248
Designing Inheritance Hierarchies	248
Defining Class Members	250
Single Inheritance vs. Multiple Inheritance	251
Overriding and Polymorphism	252
Abstract Classes and Abstract Operations	253
Inheritance of Interface	254
Visual Basic .NET and Inheritance	257
The .NET Unified Inheritance Hierarchy	257
Benefits and Implications of a Unified Inheritance Hierarchy	262

Inheritance of Implementation	264
Defining the Superclass	264
Defining Subclasses	271
Inheritance of Interface	278
What Can We Define In An Interface?	287
Implementing Multiple Interfaces	293
Summary	301
 Chapter 8: Code Organization and Metadata	 305
Structuring Applications with Namespaces	306
Setting Namespaces using the Command-Line Compiler	310
Designing and Implementing Namespaces	312
Example of Designing Namespaces	313
Example of Implementing Namespaces	314
Nested Namespaces	321
Understanding Metadata	324
Viewing Metadata in a Single-File Assembly	324
Creating Multi-File Assemblies	330
Example of a Multi-File Assembly	330
Deploying Applications as Assemblies	335
Deploying Single-Assembly Applications	336
Deploying Applications using Private Assemblies	336
Deploying Private Assemblies in the Same Folder	337
Deploying Private Assemblies in a Different Sub-Folder	339
Deploying Shared Assemblies	342
Creating a Shared Assembly as a Strongly Named Assembly	343
Creating an Application that Uses the Shared Assembly	345
Installing Shared Assemblies into the Global Assembly Cache	347
Generating Documentation for an Assembly	350
Summary	352
 Appendix A: Support, Errata, and forums.apress.com	 355
Errata on www.apress.com	356
Customer Support	356
The Peer-to-Peer Forums at forums.apress.com	357
 Index	 361

VB.NET

Class Design

Handbook

Introduction