

Contents

About the Author	vii
About the Technical Reviewer	ix
Acknowledgments	xi
Introduction	xiii
CHAPTER 1 Testing Your Code	1
Quick Notes About TDD	1
Getting Started with TDD and NUnit	2
Writing Tests Using Contexts and Results	6
Writing Tests for Code Pieces That Have No Tests or Few Tests	11
Writing Tests for Code Pieces That Don't Give Information Back	19
Verifying the Correctness of an Object Instance Without Having Access	26
CHAPTER 2 .NET Runtime- and Framework-Related Solutions	31
Keeping Value Types and Reference Types Straight	31
Using Delegates	41
Versioning Assemblies	43
Loading and Unloading Assemblies Dynamically	47
Loading Assemblies Dynamically	47
Loading and Unloading Assemblies Dynamically	52
Implementing GetHashCode	67
Thinking of .NET Generics as Black Boxes	72
Figuring Out What Generic Methods Do	76
Using the new and class Keywords with .NET Generics	82
CHAPTER 3 Text-Related Solutions	85
Converting a String to an Array and Vice Versa	85
Parsing Numbers from Buffers	89
Processing Plain-Vanilla Numbers in Different Cultures	89
Managing the Culture Information	93

When to Use StringBuilder	97
Finding a Piece of Text Within a Text Buffer	101
Always Implement ToString	104
Using a Disposable Type to Find Multiple Text Pieces and Iterate the Results	106
Making ToString Generate Structured Output	110
CHAPTER 4 C# Coding Solutions	117
What Does the Yield Keyword Really Generate?	117
Using Inheritance Effectively	123
Implementing Interfaces	128
Naming Conventions for a Namespace, a Class, and an Interface	135
Namespaces	135
Class and Interface Identifiers	136
Understanding the Overloaded Return Type and Property	139
Nullable Types: A Null Is Not Always a Null	145
Abstract-Class Bridge-Pattern Variation	148
Nested Private-Class Bridge-Pattern Variation	151
Dealing with Marker Interfaces or Base Classes	153
Editing Text Using the Command Pattern	154
Marker Interfaces and Their Dependencies	156
How Marker Interfaces Dependencies Are Implemented	157
A Null Value Is Not Always a Null State	165
The Essentials of the Factory Pattern	169
The Classical Factory Pattern	170
More Sophisticated Factory Implementations	173
Don't Expose a Class's Internal State	178
Designing Consistent Classes	181
Immutable Types Are Scalable Types	187
Understanding and Using Functors	191
The Comparer Functor	196
The Closure Functor	200
The Predicate Functor	200
The Transformer Functor	202
Functors in Practice	202
Avoiding Parameters That Have No Identity	205
INDEX	209