Effective SOFTWARE DEVELOPMENT SERIES

Scott Meyers, Consulting Editor

MORE *Effective* C#

Second Edition

COVERS C# 7.0

50 Specific Ways to Improve Your C#

Content Update Program

FREE…See Details Inside

Bill Wagner

More Effective C#

Second Edition

More Effective C#: 50 Specific Ways to Improve Your C#

Table of Contents

Cover

Title Page

Copyright Page

Contents

Introduction

Chapter 1 Working with Data Types

- Item 1: Use Properties Instead of Accessible Data Members
- Item 2: Prefer Implicit Properties for Mutable Data
- Item 3: Prefer Immutability for Value Types
- Item 4: Distinguish Between Value Types and Reference Types
- Item 5: Ensure That 0 Is a Valid State for Value Types
- Item 6: Ensure That Properties Behave Like Data
- Item 7: Limit Type Scope by Using Tuples
- Item 8: Define Local Functions on Anonymous Types
- Item 9: Understand the Relationships Among the Many Different Concepts of Equality
- Item 10: Understand the Pitfalls of GetHashCode()

Chapter 2 API Design

- Item 11: Avoid Conversion Operators in Your APIs
- Item 12: Use Optional Parameters to Minimize Method Overloads
- Item 13: Limit Visibility of Your Types

Pearson

Table of Contents

Item 14: Prefer Defining and Implementing Interfaces to Inheritance

Item 15: Understand How Interface Methods Differ from Virtual Methods

Item 16: Implement the Event Pattern for Notifications

Item 17: Avoid Returning References to Internal Class Objects

Item 18: Prefer Overrides to Event Handlers

Item 19: Avoid Overloading Methods Defined in Base Classes

Item 20: Understand How Events Increase Runtime Coupling Among Objects

Item 21: Declare Only Nonvirtual Events

Item 22: Create Method Groups That Are Clear, Minimal, and Complete

Item 23: Give Partial Classes Partial Methods for Constructors, Mutators, and Event Handlers

Item 24: Avoid ICloneable Because It Limits Your Design Choices

Item 25: Limit Array Parameters to params Arrays

Item 26: Enable Immediate Error Reporting in Iterators and Async Methods Using Local Functions

Chapter 3 Task-Based Asynchronous Programming

Item 27: Use Async Methods for Async Work

Item 28: Never Write async void Methods

Item 29: Avoid Composing Synchronous and Asynchronous Methods

Item 30: Use Async Methods to Avoid Thread Allocations and Context Switches

Item 31: Avoid Marshalling Context Unnecessarily

Item 32: Compose Asynchronous Work Using Task Objects

Item 33: Consider Implementing the Task Cancellation Protocol

Table of Contents

Item 34: Cache Generalized Async Return Types

Chapter 4 Parallel Processing

Item 35: Learn How PLINQ Implements Parallel Algorithms

Item 36: Construct Parallel Algorithms with Exceptions in Mind

Item 37: Use the Thread Pool Instead of Creating Threads

Item 38: Use BackgroundWorker for Cross-Thread Communication

Item 39: Understand Cross-Thread Calls in XAML Environments

Item 40: Use lock() as Your First Choice for Synchronization

Item 41: Use the Smallest Possible Scope for Lock Handles

Item 42: Avoid Calling Unknown Code in Locked Sections

Chapter 5 Dynamic Programming

Item 43: Understand the Pros and Cons of Dynamic Typing

Item 44: Use Dynamic Typing to Leverage the Runtime Type of Generic Type Parameters

Item 45: Use DynamicObject or IDynamicMetaObjectProvider for Data-Driven Dynamic Types

Item 46: Understand How to Use the Expression API

Item 47: Minimize Dynamic Objects in Public APIs

Chapter 6 Participate in the Global C# Community

Item 48: Seek the Best Answer, Not the Most Popular Answer

Item 49: Participate in Specs and Code

Item 50: Consider Automating Practices with Analyzers

Index