

Contents

Acknowledgments	v
Introduction	xv
Chapter 1: Generics 101	1
Why Generics?	1
Enter Generics	7
Hello Generics	10
A More Conceptual View	12
Terminology	14
Type Parameters	14
Open Types	15
Constructed Types	15
Type Arguments	15
Open and Closed Constructed Types	15
Generic Methods	16
Type Instantiation	16
Arity	17
Generic Types	17
Bringing It All Together	17
Summary	17
Chapter 2: Valuing Type Safety	19
Motivation	19
Least Common Denominator Programming	20
A Basic Example	21
Applying Generics	26
Casting Consequences	29
Interface Type Safety	31
Scratching the Surface	32
Safety vs. Clarity	32
Summary	33
Chapter 3: Generics \neq Templates	35
Shared Concepts	35
Run-Time vs. Compile-Time	36

Contents

Compile-Time Instantiation (Templates)	36
Run-Time Instantiation (Generics)	37
Lost in Translation	38
The Brouhaha	42
Code Bloat	42
Assemblies and Type Equivalence	42
Templates Extras	43
Template Specialization	43
Non-Type Parameters	44
Type Parameter Inheritance	44
Cross-Language Support	44
Debugging	44
Mix and Match Nirvana	44
Summary	45
Chapter 4: Generic Classes	47
<hr/>	
Parameterizing Types	47
Type Parameters	49
Overloaded Types	50
Static Constructors	51
Inheritance	52
Protected Members	55
Fields	56
Static Fields	57
Constructed Fields	59
Methods	60
Overloading Methods	61
Overriding Methods	63
Arrays of Type Parameters	65
Operator Overloading	66
Nested Classes	67
Consuming Generic Classes	69
Accessibility	70
The Default Keyword	73
System.Nullable<T>	74
Accessing Type Info	76
Indexers, Properties, and Events	77
Generic Structs	80
Generic Interfaces	81
Summary	82

Chapter 5: Generic Methods	83
The Basics	83
A Deeper Look	85
Apply Constraints	87
Type Parameter Names	88
Overloading Generic Methods	88
Uniqueness with Generic Classes	90
Constraints and Uniqueness	91
Overriding Generic Methods	92
Type Inference	95
Generic Methods and Delegates	96
Type-Safe Database Access Example	97
Summary	98
Chapter 6: Generic Delegates	99
Delegate Basics	99
Adding Generics to the Equation	102
Event Handling	105
Generic Delegates with Generic Methods	106
Delegates in the Framework	106
Action<T>	106
Comparison<T>	107
Converter<T, U>	107
Predicate<T>	107
Choosing Your Delegates	107
Type Coercion	107
Applying Constraints	108
Delegates and Anonymous Methods	109
Summary	109
Chapter 7: Generic Constraints	111
Overview	111
Constraint Types	115
Interface Constraints	115
Class Constraints	116
Generic Types as Constraints	117
Constructor Constraints	118
Boxing and Constraints	119

Contents

Using Multiple Constraints	122
Ambiguous Constraints	123
Ambiguity When Mixing Classes and Interfaces	124
Generic Delegate and Method Constraints	125
Inheritance and Constraints	126
Are Generics Generic?	127
Summary	128
Chapter 8: BCL Generics	129
Motivation	129
The Big Picture	130
Generic Collection Interfaces	130
Classes	132
System.Collections.ObjectModel	133
Enumerators	133
Using Delegates	134
Generic Collection Classes	134
Collection<T>	134
Comparer<T>	147
Dictionary<TKey, TValue>	149
EqualityComparer<T>	157
KeyedCollection<TKey, TItem>	159
LinkedList<T>	160
LinkedListNode<T>	161
List<T>	162
Queue<T>	175
ReadOnlyCollection<T>	177
SortedDictionary<TKey, TValue>	179
SortedList<TKey, TValue>	182
Stack<T>	183
BindingList<T>	184
Testing Equality	184
Null Collection Elements	185
More to Come	185
Summary	185
Chapter 9: Reflection, Serialization, and Remoting	187
Reflection	187
Working with Open and Closed Types	188
Extending System.Type	188
Determining if a Type Is Generic	188

Creating Open and Closed Types with Reflection	189
Converting Closed Types to Open Types	194
Examining Parameters and Arguments	195
Reflection and Generic Inheritance	198
Reflecting on Generic Methods	199
Obfuscation Reminder	201
Serialization	202
Serialization Basics	202
Custom Serialization	204
Serialization with Web Services	208
Remoting	211
Summary	215
Chapter 10: Generics Guidelines	217
<hr/>	
An Evolving List	217
Defining Guidelines	217
Organization	218
Identifying Generic Opportunities	218
Item 1: Use Generic Collections	218
Item 2: Replace Objects with Type Parameters	219
Item 3: Replace System.Type with Type Parameters	219
Item 4: Use Type Parameters for Ref Types (C# Only)	220
Item 5: Genericize Types That Vary Only by a Data Type	221
Balancing Readability with Expressiveness	227
Item 6: Use Expressive, Consistent Type Parameter Names	228
Item 7: Use Aliasing for Complex or Frequently Used Types	231
Item 8: Don't Use Constructed Types as Type Arguments	232
Item 9: Don't Use Too Many Type Parameters	233
Item 10: Prefer Type Inference with Generic Methods	233
Item 11: Don't Mix Generic and Non-Generic Static Methods	234
Using BCL Generic Types	235
Item 12: Custom Collections Should Extend Collection<T>	235
Item 13: Use the Least Specialized Interface in Your APIs	235
Item 14: Enable "for each" Iteration with IEnumerable<T>	236
Applying Constraints	236
Item 15: Select the Least Restrictive Constraints	236
Item 16: Don't Impose Hidden Constraints	237
Item 17: Avoid Multiple Constraint Ambiguity	237
Item 18: Provide Parameterless Constructors	238

Contents

The Kitchen Sink	239
Item 19: Use Static Data Members with Caution	239
Item 20: Use Interfaces in Lieu of Classes	240
Item 21: Use Comparer<T> for All Type Comparisons	241
Item 22: Use Nullable<T> for Optional Values	241
Item 23: Use EventHandler<T> for All Events	241
Summary	242
Chapter 11: Under the Hood	243
Overview	243
Assumptions	244
High-Level Goals	244
Validate at Declaration	244
Simplicity	244
Platform Conformity	245
Language Agnostic Generics	245
No Boxing	245
No Meta-Programming	245
IL Representation of Generic Types	246
Specialization and Sharing	248
Code Specialization	248
Code Sharing	248
The .NET Hybrid Model	248
Just-in-Time Specialization	249
Exact Run-Time Types	254
Support for Polymorphic Recursion	256
NGen and Generic Types	257
Performance	259
Memory Footprint	261
Backward Compatibility	262
Summary	262
Chapter 12: Using Generics with C++	263
Templates or Generics?	263
Blurring the Lines	264
No Limits	264
Generic Classes in C++	264
Consuming Generic Classes	265
Inheritance	266
Nested Classes	267

Methods in Generic Classes	268
Default Types	268
Generic Methods in C++	268
Generic Interfaces in C++	270
Constraints	271
Generic Delegates in C++	273
Mixing Generics and Templates	274
STL.NET	277
Summary	278
Chapter 13: Using Generics with J#	279
<hr/>	
J# Generic Limitations	279
Migrating from Java Generics	280
Consuming Generic Types	280
Leveraging Cross-Language Support	281
Working with Interfaces	283
Complying with Constraints	285
Calling Generic Methods	287
Java Type Arguments	288
Calling Generic Delegates	289
Arrays of Generic Types	289
Summary	290
Chapter 14: Power Collections	291
<hr/>	
A Little History	291
The Big Picture	292
Base Classes	292
Containers	293
Algorithms Class	295
Generic Structures	295
Delegates	296
Class Reference	296
Algorithms	296
Bag<T>	321
BigList<T>	332
BinaryPredicate<T> (Delegate)	336
CollectionBase<T>	336
Deque<T>	337
DictionaryBase<TKey, TValue>	341
ListBase<T>	342

Contents

MultiDictionary<TKey, TValue>	343
MultiDictionaryBase<TKey, TValue>	345
OrderedBag<T>	346
OrderedBag<T>.View	349
OrderedDictionary<TKey, TValue>	350
OrderedDictionary<TKey, TValue>.View	354
OrderedMultiDictionary<TKey, TValue>	354
OrderedMultiDictionary<TKey, TValue>.View	355
OrderedSet<T>	355
OrderedSet<T>.View	358
Pair<TFirst, TSecond >	358
ReadOnlyCollectionBase<T>	358
ReadOnlyDictionaryBase<TKey, TValue>	359
ReadOnlyListBase<T>	360
ReadOnlyMultiDictionary<TKey, TValue>	361
Set<T>	362
Triple<TFirst, TSecond, TThird>	366
Additional Libraries	367
Summary	367
Index	369