



Содержание

<i>Предисловие</i>	17
<i>Предисловие к первому изданию</i>	19
<i>Предисловие</i>	21
<i>Благодарности</i>	25
<i>Об авторах</i>	27
<i>О комментаторах</i>	29
1 Введение	37
1.1. Характеристики правильно спроектированной инфраструктуры	39
1.1.1. Правильно спроектированные инфраструктуры просты	39
1.1.2. Правильно спроектированные инфраструктуры обходятся дорого	40
1.1.3. Правильно спроектированные инфраструктуры полны компромиссов	40
1.1.4. Учет опыта в правильно спроектированных инфраструктурах	41
1.1.5. Правильно спроектированные инфраструктуры проектируются так, чтобы их можно было расширить	41
1.1.6. Правильно спроектированные инфраструктуры интегрированы	42
1.1.7. Правильно спроектированные инфраструктуры непротиворечивы	42
2 Основные принципы разработки инфраструктуры	43
2.1. Прогрессивные инфраструктуры	45
2.2. Фундаментальные принципы разработки инфраструктуры	48
2.2.1. Принцип дизайна, основанного на сценариях	49
2.2.2. Принцип низкого барьера для доступа	54
2.2.3. Принцип самодокументирования объектных моделей	59
2.2.4. Принцип многоуровневой архитектуры	65

3	Рекомендации по обозначениям	69
3.1.	Соглашения о прописных буквах	70
3.1.1.	Правила использования прописных букв в идентификаторах	70
3.1.2.	Набор акронимов прописными буквами	72
3.1.3.	Использование прописных букв в составных словах и общепринятых терминах	74
3.1.4.	Чувствительность к регистру	75
3.2.	Общие соглашения об именах	76
3.2.1.	Выбор слов	77
3.2.2.	Использование сокращений и акронимов	79
3.2.3.	Избегайте названий, специфических для языка	79
3.2.4.	Обозначение новых версий для существующих API	81
3.3.	Названия сборок и DLL	83
3.4.	Названия пространств имен	85
3.4.1.	Пространства имен и конфликты с именами типов	86
3.5.	Названия классов, структур и интерфейсов	88
3.5.1.	Названия параметров универсальных типов	91
3.5.2.	Названия общих типов	92
3.5.3.	Обозначение перечислений	93
3.6.	Имена членов типа	95
3.6.1.	Названия методов	95
3.6.2.	Названия свойств	95
3.6.3.	Названия событий	97
3.6.4.	Обозначение полей	98
3.7.	Обозначение параметров	99
3.7.1.	Обозначение параметров перегружаемых операторов	100
3.8.	Обозначение ресурсов	100
4	Рекомендации по разработке типов	103
4.1.	Типы и пространства имен	105
4.1.1.	Стандартные названия подпространств имен	108
4.2.	Выбор между классом и структурой	110
4.3.	Выбор между классом и интерфейсом	113
4.4.	Дизайн абстрактных классов	119
4.5.	Разработка статических классов	121
4.6.	Проект интерфейса	122
4.7.	Разработка структур	124

8 Оглавление

4.8. Разработка перечислений	126
4.8.1. Проектирование перечислений флажков	132
4.8.2. Добавление значений к перечислениям	136
4.9. Вложенные типы	137
4.10. Типы и метаданные сборки	139
5 Проектирование членов	141
5.1. Общие рекомендации по разработке членов	141
5.1.1. Перегрузка членов	141
5.1.2. Явная реализация членов интерфейса	147
5.1.3. Выбор между свойствами и методами	151
5.2. Разработка свойств	156
5.2.1. Разработка индексированных свойств	158
5.2.2. События, уведомляющие об изменении свойств	159
5.3. Разработка конструкторов	161
5.3.1. Рекомендации для конструктора типов	166
5.4. Разработка событий	168
5.4.1. Разработка обработчика событий пользователем	173
5.5. Разработка полей	174
5.6. Методы расширения	176
5.7. Перегрузка операторов	181
5.7.1. Перегрузка оператора <code>operator==</code>	185
5.7.2. Операторы преобразования	185
5.8. Разработка параметров	187
5.8.1. Выбор между перечислениями и булевыми параметрами	189
5.8.2. Проверка правильности параметров	190
5.8.3. Передача параметров	193
5.8.4. Члены с переменным числом параметров	196
5.8.5. Параметры-указатели	199
6 Проектирование с целью расширяемости	201
6.1. Механизмы расширяемости	201
6.1.1. Негерметичные классы	202
6.1.2. Защищенные члены	203
6.1.3. События и обратные вызовы	204
6.1.4. Виртуальные члены	208
6.1.5. Абстракции (абстрактные типы и интерфейсы)	210

6.2. Базовые классы	212
6.3. Герметизация	214
7 Исключения	217
7.1. Вызов исключения	221
7.2. Выбор правильного типа вызываемого исключения	226
7.2.1. Разработка сообщений об ошибках	229
7.2.2. Обработка особых ситуаций	231
7.2.3. Обертывание исключений	235
7.3. Использование стандартных типов исключений	237
7.3.1. Исключения Exception и SystemException	237
7.3.2. Исключение ApplicationException	237
7.3.3. Исключение InvalidOperationException	238
7.3.4. Исключения ArgumentException, ArgumentNullException и ArgumentOutOfRangeException	238
7.3.5. Исключения NullReferenceException, IndexOutOfRangeException и AccessViolationException	239
7.3.6. Исключение StackOverflowException	240
7.3.7. Исключение OutOfMemoryException	240
7.3.8. Исключения ComException, SEHException и ExecutionEngineException	241
7.4. Проектирование пользовательских исключений	241
7.5. Исключения и производительность	243
7.5.1. Шаблон Tester-Doer)	243
7.5.2. Шаблон Try-Parse)	244
8 Рекомендации по использованию	247
8.1. Массивы	247
8.2. Атрибуты	249
8.3. Коллекции	251
8.3.1. Параметры-коллекции	253
8.3.2. Свойства коллекции и возвращаемые значения	254
8.3.3. Выбор между массивами и коллекциями	257
8.3.4. Реализация пользовательских коллекций	258
8.4. Типы DateTime и DateTimeOffset	260
8.5. Интерфейс ICloneable	262
8.6. IComparable<T> и IEquatable<T>	263

10 ■ Оглавление

8.7. IDisposable	265
8.8. Nullable<T>	265
8.9. Объект Object	266
8.9.1. Object.Equals	266
8.9.2. Хеш-функция Object.GetHashCode	268
8.9.3. Метод Object.ToString	269
8.10. Сериализация — преобразование в последовательную форму	271
8.10.1. Выбор подходящей для поддержки технологии преобразования в последовательную форму	272
8.10.2. Поддержка преобразования в последовательную форму в соответствии с контрактом данных	273
8.10.3. Поддержка преобразования в последовательную XML-форму	277
8.10.4. Поддержка преобразования в последовательную форму во время выполнения	278
8.11. Тип Uri	280
8.12. Использование System.Xml	281
8.13. Операторы равенства	282
8.13.1. Операторы равенства на типах значений	283
8.13.2. Операторы равенства на ссылочных типах	283
9 Общие шаблоны проектирования	285
9.1. Составные компоненты	285
9.1.1. Компонентно-ориентированный дизайн	287
9.1.2. Факторизованные типы	289
9.1.3. Рекомендации по объединению составляющих	290
9.2. Асинхронные шаблоны Async	293
9.2.1. Выбор одного из асинхронных шаблонов Async	293
9.2.2. Классический асинхронный шаблон Classic Async	295
9.2.3. Пример базовой реализации классического асинхронного шаблона Classic Async	298
9.2.4. Основанный на событиях асинхронный шаблон Event-Based Async	299
9.2.5. Поддержка out- и ref-параметров	301
9.2.6. Поддержка отмены	301
9.2.7. Поддержка сообщения о продвижении	302
9.2.8. Поддержка приращений — возрастающих результатов	304
9.3. Зависимые свойства	304
9.3.1. Разработка зависимых свойств	305
9.3.2. Разработка прикрепленных зависимых свойств	307

9.3.3. Проверка правильности зависимого свойства	308
9.3.4. Уведомления об изменении зависимого свойства	309
9.3.5. Приведение типа данных значения зависимого свойства	310
9.4. Шаблон освобождения Dispose	311
9.4.1. Основной шаблон освобождения Basic Dispose	313
9.4.2. Финализируемые типы	318
9.5. Фабрики	322
9.6. Поддержка LINQ	325
9.6.1. Краткий обзор LINQ	326
9.6.2. Способы реализации поддержки LINQ	327
9.6.3. Поддержка LINQ через IEnumerable<T>	327
9.6.4. Поддержка LINQ через IQueryable<T>	328
9.6.5. Поддержка LINQ через шаблон запроса Query	329
9.7. Шаблон дополнительных возможностей Optional Feature	331
9.8. Моделирование ковариации	335
9.9. Шаблон Template Method	340
9.10. Тайм-аут (истечение времени ожидания)	341
9.11. Читаемые XAML-типы	343
9.12. И в заключение...	345
А Соглашения о стиле кодирования в C#	347
A.1. Общие соглашения о стиле	348
A.1.1. Использование скобок	348
A.1.2. Использование пробелов	349
A.1.3. Использование отступов	350
A.1.4. Другое	350
A.2. Соглашения об именах	351
A.3. Комментарии	351
A.4. Организация файлов	352
Б Использование FxCop для проверки рекомендаций по разработке инфраструктур	355
Б.1. Что такое FxCop	355
Б.2. Развитие FxCop	356
Б.3. Как это работает	357

12 ■ Оглавление

Б.4. Описание руководства по FxСор	357
Б.4.1. Правила FxСор для рекомендаций по обозначениям	357
Б.4.2. Правила FxСор для разработки типа	366
Б.4.3. Правила FxСор по разработке членов	369
Б.4.4. Правила FxСор для проектирования расширяемости	375
Б.4.5. Правила FxСор для исключений	375
Б.4.6. Правила FxСор по использованию инфраструктуры	378
Б.4.7. Правила FxСор для шаблонов разработки Design Patterns	382
В Пример спецификации API	385
1. Требования	
2. Спецификации API	
2.1. Сценарии	
2.2. Разработка API	
3. Функциональные спецификации	388
Глоссарий	393
Список рекомендуемой литературы	399
Предметный указатель	403