

Содержание

Об авторах	33
Благодарности	33
Введение	34
Авторы и читатели — одна команда	35
Краткий обзор книги	36
Часть I. Введение в C# и платформу .NET	36
Часть II. Основы программирования на C#	36
Часть III. Объектно-ориентированное программирование на C#	37
Часть IV. Дополнительные конструкции программирования на C#	37
Часть V. Программирование с использованием сборок .NET	39
Часть VI. Введение в библиотеки базовых классов .NET	40
Часть VII. Windows Presentation Foundation	41
Часть VIII. ASP.NET	43
Загружаемые приложения	44
Исходный код примеров	44
От издательства	44
Часть I. Введение в C# и платформу .NET	45
Глава 1. Философия .NET	46
Начальное знакомство с платформой .NET	46
Некоторые основные преимущества платформы .NET	47
Введение в строительные блоки платформы .NET (CLR, CTS и CLS)	47
Роль библиотек базовых классов	48
Что привносит язык C#	48
Сравнение управляемого и неуправляемого кода	51
Другие языки программирования, ориентированные на .NET	51
Жизнь в многоязычном мире	52
Обзор сборок .NET	52
Роль языка CIL	54
Роль метаданных типов .NET	56
Роль манифеста сборки	57
Понятие общей системы типов (CTS)	58
Типы классов CTS	58
Типы интерфейсов CTS	59
Типы структур CTS	59
Типы перечислений CTS	60
Типы делегатов CTS	60
Члены типов CTS	61
Встроенные типы данных CTS	61
Понятие общезыковой спецификации (CLS)	62
Обеспечение совместимости с CLS	63
Понятие общезыковой исполняющей среды (CLR)	64
Различия между сборками, пространствами имен и типами	64
Роль корневого пространства имен Microsoft	68
Доступ к пространству имен программным образом	69
Ссылка на внешние сборки	70
Исследование сборки с помощью ildasm.exe	71

Просмотр кода CIL	72
Просмотр метаданных типов	72
Просмотр метаданных сборки (манифеста)	72
Независимая от платформы природа .NET	72
Проект Mono	74
Microsoft .NET Core	75
Резюме	75
Глава 2. Создание приложений на языке C#	76
Построение приложений C# в среде Windows	76
Семейство IDE-сред Visual Studio Express	77
IDE-среда Visual Studio Community	84
IDE-среда Visual Studio 2015 Professional	88
Система документации .NET Framework	89
Построение приложений .NET за пределами среды Windows	91
Роль Xamarin Studio	91
Резюме	93
Часть II. Основы программирования на C#	95
Глава 3. Главные конструкции программирования на C#: часть I	96
Структура простой программы C#	96
Вариации метода Main()	98
Указание кода ошибки приложения	98
Обработка аргументов командной строки	100
Указание аргументов командной строки в Visual Studio	101
Интересное отступление от темы: некоторые дополнительные члены	
класса System.Environment	101
Класс System.Console	103
Базовый ввод-вывод с помощью класса Console	104
Форматирование консольного вывода	105
Форматирование числовых данных	106
Форматирование числовых данных за рамками консольных приложений	107
Системные типы данных и соответствующие ключевые слова C#	108
Объявление и инициализация переменных	109
Внутренние типы данных и операция new	110
Иерархия классов для типов данных	111
Члены числовых типов данных	112
Члены System.Boolean	113
Члены System.Char	113
Разбор значений из строковых данных	114
Типы System.DateTime и System.TimeSpan	114
Сборка System.Numerics.dll	115
Работа со строковыми данными	116
Базовые манипуляции строками	117
Конкатенация строк	118
Управляющие последовательности	118
Определение дословных строк	119
Строки и равенство	120
Строки являются неизменяемыми	120
Тип System.Text.StringBuilder	122
Интерполяция строк	123

8 Содержание

Сужающие и расширяющие преобразования типов данных	124
Ключевое слово <code>checked</code>	126
Настройка проверки переполнения на уровне проекта	128
Ключевое слово <code>unchecked</code>	129
Понятие неявно типизированных локальных переменных	129
Ограничения неявно типизированных переменных	131
Неявно типизированные данные являются строго типизированными	132
Полезность неявно типизированных локальных переменных	132
Итерационные конструкции C#	133
Цикл <code>for</code>	134
Цикл <code>foreach</code>	134
Циклы <code>while</code> и <code>do/while</code>	135
Конструкции принятия решений и операции отношения/равенства	136
Оператор <code>if/else</code>	136
Операции отношения и равенства	136
Условные операции	137
Оператор <code>switch</code>	137
Резюме	139
Глава 4. Главные конструкции программирования на C#: часть II	140
Методы и модификаторы параметров	140
Стандартное поведение передачи параметров по значению	141
Модификатор <code>out</code>	142
Модификатор <code>ref</code>	143
Модификатор <code>params</code>	144
Определение необязательных параметров	146
Вызов методов с использованием именованных параметров	147
Понятие перегрузки методов	148
Понятие массивов C#	150
Синтаксис инициализации массивов C#	151
Неявно типизированные локальные массивы	152
Определение массива элементов типа <code>object</code>	152
Работа с многомерными массивами	153
Использование массивов в качестве аргументов и возвращаемых значений	154
Базовый класс <code>System.Array</code>	155
Тип <code>enum</code>	157
Управление хранилищем, лежащим в основе перечисления	158
Объявление переменных типа перечисления	158
Тип <code>System.Enum</code>	159
Динамическое извлечение пар “имя-значение” перечисления	160
Понятие структуры (как типа значения)	162
Создание переменных типа структур	163
Типы значений и ссылочные типы	164
Типы значений, ссылочные типы и операция присваивания	165
Типы значений, содержащие ссылочные типы	167
Передача ссылочных типов по значению	168
Передача ссылочных типов по ссылке	169
Заключительные детали относительно типов значений и ссылочных типов	170
Понятие типов C#, допускающих <code>null</code>	171
Работа с типами, допускающими <code>null</code>	173
Операция объединения с <code>null</code>	174
<code>null</code> -условная операция	174
Резюме	175

Часть III. Объектно-ориентированное программирование на C#	177
Глава 5. Инкапсуляция	178
Знакомство с типом класса C#	178
Размещение объектов с помощью ключевого слова <code>new</code>	180
Понятие конструкторов	181
Роль стандартного конструктора	181
Определение специальных конструкторов	182
Еще раз о стандартном конструкторе	183
Роль ключевого слова <code>this</code>	184
Построение цепочки вызовов конструкторов с использованием <code>this</code>	186
Изучение потока управления конструкторов	188
Еще раз о необязательных аргументах	190
Понятие ключевого слова <code>static</code>	191
Определение статических полей данных	192
Определение статических методов	193
Определение статических конструкторов	194
Определение статических классов	197
Импортирование статических членов с применением ключевого слова <code>using</code> языка C#	198
Основные принципы объектно-ориентированного программирования	199
Роль инкапсуляции	199
Роль наследования	199
Роль полиморфизма	201
Модификаторы доступа C#	202
Стандартные модификаторы доступа	202
Модификаторы доступа и вложенные типы	204
Первый принцип ООП: службы инкапсуляции C#	204
Инкапсуляция с использованием традиционных методов доступа и изменения	205
Инкапсуляция с использованием свойств .NET	207
Использование свойств внутри определения класса	210
Свойства, которые допускают только чтение и только запись	211
Еще раз о ключевом слове <code>static</code> : определение статических свойств	212
Понятие автоматических свойств	213
Взаимодействие с автоматическими свойствами	214
Автоматические свойства и стандартные значения	215
Инициализация автоматических свойств	216
Понятие синтаксиса инициализации объектов	217
Вызов специальных конструкторов с помощью синтаксиса инициализации	218
Инициализация данных с помощью синтаксиса инициализации	219
Работа с данными константных полей	220
Понятие полей, допускающих только чтение	222
Статические поля, допускающие только чтение	222
Понятие частичных классов	223
Сценарии использования для частичных классов?	224
Резюме	224
Глава 6. Наследование и полиморфизм	225
Базовый механизм наследования	225
Указание родительского класса для существующего класса	226
Замечание относительно множества базовых классов	228
Ключевое слово <code>sealed</code>	228

Корректировка диаграмм классов Visual Studio	229
Второй принцип ООП: детали наследования	231
Управление созданием объектов базового класса с помощью ключевого слова <code>base</code>	232
Хранение секретов семейства: ключевое слово <code>protected</code>	234
Добавление запечатанного класса	235
Реализация модели включения/делегации	236
Определения вложенных типов	237
Третий принцип ООП: поддержка полиморфизма в C#	239
Ключевые слова <code>virtual</code> и <code>override</code>	240
Переопределение виртуальных членов в IDE-среде Visual Studio	242
Запечатывание виртуальных членов	243
Абстрактные классы	243
Полиморфные интерфейсы	244
Соккрытие членов	248
Правила приведения для базовых и производных классов	250
Ключевое слово <code>as</code>	251
Ключевое слово <code>is</code>	253
Главный родительский класс <code>System.Object</code>	253
Переопределение метода <code>System.Object.ToString()</code>	256
Переопределение метода <code>System.Object.Equals()</code>	257
Переопределение метода <code>System.Object.GetHashCode()</code>	258
Тестирование модифицированного класса <code>Person</code>	259
Статические члены класса <code>System.Object</code>	260
Резюме	260
Глава 7. Структурированная обработка исключений	261
Ода ошибкам, дефектам и исключениям	261
Роль обработки исключений .NET	262
Строительные блоки обработки исключений в .NET	263
Базовый класс <code>System.Exception</code>	263
Простейший пример	265
Генерация общего исключения	267
Перехват исключений	268
Конфигурирование состояния исключения	270
Свойство <code>TargetSite</code>	270
Свойство <code>StackTrace</code>	271
Свойство <code>HelpLink</code>	271
Свойство <code>Data</code>	272
Исключения уровня системы (<code>System.SystemException</code>)	274
Исключения уровня приложения (<code>System.ApplicationException</code>)	274
Построение специальных исключений, способ первый	275
Построение специальных исключений, способ второй	277
Построение специальных исключений, способ третий	277
Обработка множества исключений	279
Общие операторы <code>catch</code>	281
Повторная генерация исключений	281
Внутренние исключения	282
Блок <code>finally</code>	283
Фильтры исключений	284
Отладка необработанных исключений с использованием Visual Studio	285
Резюме	286

Глава 8. Работа с интерфейсами	287
Понятие интерфейсных типов	287
Сравнение интерфейсных типов и абстрактных базовых классов	288
Определение специальных интерфейсов	290
Реализация интерфейса	292
Обращение к членам интерфейса на уровне объектов	294
Получение ссылок на интерфейсы: ключевое слово <code>as</code>	295
Получение ссылок на интерфейсы: ключевое слово <code>is</code>	296
Использование интерфейсов в качестве параметров	296
Использование интерфейсов в качестве возвращаемых значений	298
Массивы интерфейсных типов	299
Реализация интерфейсов с использованием Visual Studio	300
Явная реализация интерфейсов	301
Проектирование иерархий интерфейсов	303
Множественное наследование с помощью интерфейсных типов	304
Интерфейсы <code>IEnumerable</code> и <code>IEnumerator</code>	306
Построение методов итератора с использованием ключевого слова <code>yield</code>	309
Построение именованного итератора	310
Интерфейс <code>ICloneable</code>	311
Более сложный пример клонирования	313
Интерфейс <code>IComparable</code>	316
Указание множества порядков сортировки с помощью <code>IComparer</code>	318
Специальные свойства и специальные типы сортировки	320
Резюме	320
Часть IV. Дополнительные конструкции программирования на C#	321
Глава 9. Коллекции и обобщения	322
Побудительные причины создания классов коллекций	322
Пространство имен <code>System.Collections</code>	324
Обзор пространства имен <code>System.Collections.Specialized</code>	326
Проблемы, присущие необобщенным коллекциям	327
Проблема производительности	327
Проблема безопасности к типам	330
Первый взгляд на обобщенные коллекции	333
Роль параметров обобщенных типов	334
Указание параметров типа для обобщенных классов и структур	335
Указание параметров типа для обобщенных членов	336
Указание параметров типов для обобщенных интерфейсов	337
Пространство имен <code>System.Collections.Generic</code>	338
Синтаксис инициализации коллекций	339
Работа с классом <code>List<T></code>	340
Работа с классом <code>Stack<T></code>	342
Работа с классом <code>Queue<T></code>	343
Работа с классом <code>SortedSet<T></code>	344
Работа с классом <code>Dictionary<TKey, TValue></code>	345
Пространство имен <code>System.Collections.ObjectModel</code>	347
Работа с классом <code>ObservableCollection<T></code>	347
Создание специальных обобщенных методов	349
Выведение параметров типа	351
Создание специальных обобщенных структур и классов	352
Ключевое слово <code>default</code> в обобщенном коде	353

12 Содержание

Ограничение параметров типа	354
Примеры использования ключевого слова <code>where</code>	355
Отсутствие ограничений операций	356
Резюме	357
Глава 10. Делегаты, события и лямбда-выражения	358
Понятие типа делегата <code>.NET</code>	358
Определение типа делегата в <code>C#</code>	359
Базовые классы <code>System.MulticastDelegate</code> и <code>System.Delegate</code>	361
Пример простейшего делегата	363
Исследование объекта делегата	364
Отправка уведомлений о состоянии объекта с использованием делегатов	365
Включение группового вызова	368
Удаление целей из списка вызовов делегата	369
Синтаксис групповых преобразований методов	370
Понятие обобщенных делегатов	372
Обобщенные делегаты <code>Action<></code> и <code>Func<></code>	373
Понятие событий <code>C#</code>	375
Ключевое слово <code>event</code>	377
“За кулисами” событий	378
Прослушивание входящих событий	379
Упрощение регистрации событий с использованием <code>Visual Studio</code>	380
Приведение в порядок кода обращения к событиям с использованием <code>null</code> -условной операции <code>C# 6.0</code>	381
Создание специальных аргументов событий	382
Обобщенный делегат <code>EventHandler<T></code>	383
Понятие анонимных методов <code>C#</code>	384
Доступ к локальным переменным	386
Понятие лямбда-выражений	387
Анализ лямбда-выражения	389
Обработка аргументов внутри множества операторов	390
Лямбда-выражения с несколькими параметрами и без параметров	391
Модернизация примера <code>PrimAndProperCarEvents</code> с использованием лямбда-выражений	392
Лямбда-выражения и реализация членов с единственным оператором	393
Резюме	394
Глава 11. Расширенные средства языка <code>C#</code>	395
Понятие методов индексаторов	395
Индексация данных с использованием строковых значений	397
Перегрузка методов индексаторов	398
Многомерные индексаторы	399
Определения индексаторов в интерфейсных типах	400
Понятие перегрузки операций	400
Перегрузка бинарных операций	401
А как насчет операций <code>+=</code> и <code>-=</code> ?	403
Перегрузка унарных операций	404
Перегрузка операций эквивалентности	405
Перегрузка операций сравнения	405
Финальные соображения относительно перегрузки операций	406
Понятие специальных преобразований типов	407
Повторение: числовые преобразования	407
Повторение: преобразования между связанными типами классов	407

Создание специальных процедур преобразования	408
Дополнительные явные преобразования для типа <code>Square</code>	410
Определение процедур неявного преобразования	411
Понятие расширяющих методов	413
Понятие анонимных типов	417
Определение анонимного типа	418
Анонимные типы, содержащие другие анонимные типы	422
Работа с типами указателей	423
Ключевое слово <code>unsafe</code>	424
Работа с операциями <code>*</code> и <code>&</code>	426
Небезопасная (и безопасная) функция обмена	426
Доступ к полям через указатели (операция <code>-></code>)	427
Ключевое слово <code>stackalloc</code>	428
Закрепление типа посредством ключевого слова <code>fixed</code>	428
Ключевое слово <code>sizeof</code>	429
Резюме	430
Глава 12. LINQ to Objects	431
Программные конструкции, специфичные для LINQ	431
Неявная типизация локальных переменных	432
Синтаксис инициализации объектов и коллекций	433
Лямбда-выражения	433
Расширяющие методы	434
Анонимные типы	435
Роль LINQ	435
Выражения LINQ строго типизированы	436
Основные сборки LINQ	437
Применение запросов LINQ к элементарным массивам	437
Решение без использования LINQ	439
Выполнение рефлексии результирующего набора LINQ	439
LINQ и неявно типизированные локальные переменные	440
LINQ и расширяющие методы	441
Роль отложенного выполнения	442
Роль немедленного выполнения	443
Возвращение результатов запроса LINQ	444
Возвращение результатов LINQ посредством немедленного выполнения	445
Применение запросов LINQ к объектам коллекций	446
Доступ к содержащимся в контейнере подобъектам	446
Применение запросов LINQ к необобщенным коллекциям	447
Фильтрация данных с использованием метода <code>OfType<T>()</code>	448
Исследование операций запросов LINQ	449
Базовый синтаксис выборки	450
Получение подмножества данных	451
Процирование новых типов данных	451
Подсчет количества с использованием класса <code>Enumerable</code>	453
Изменение на противоположный порядка следования элементов в результирующих наборах	453
Выражения сортировки	453
LINQ как лучшее средство построения диаграмм Венна	454
Устранение дубликатов	455
Операции агрегирования LINQ	455
Внутреннее представление операторов запросов LINQ	456
Построение выражений запросов с применением операций запросов	457

14 Содержание

Построение выражений запросов с использованием типа <code>Enumerable</code> и лямбда-выражений	457
Построение выражений запросов с использованием типа <code>Enumerable</code> и анонимных методов	458
Построение выражений запросов с использованием типа <code>Enumerable</code> и низкоуровневых делегатов	459
Резюме	460
Глава 13. Время существования объектов	461
Классы, объекты и ссылки	461
Базовые сведения о времени жизни объектов	463
Код CIL для ключевого слова <code>new</code>	463
Установка объектных ссылок в <code>null</code>	465
Роль корневых элементов приложения	465
Понятие поколений объектов	467
Параллельная сборка мусора до версии <code>.NET 4.0</code>	468
Фоновая сборка мусора в <code>.NET 4.0</code> и последующих версиях	468
Тип <code>System.GC</code>	469
Принудительный запуск сборщика мусора	470
Построение финализируемых объектов	472
Переопределение метода <code>System.Object.Finalize()</code>	473
Подробности процесса финализации	475
Построение освобождаемых объектов	476
Повторное использование ключевого слова <code>using</code> в C#	478
Создание финализируемых и освобождаемых типов	479
Формализованный шаблон освобождения	480
Ленивое создание объектов	482
Настройка процесса создания данных <code>Lazy<></code>	485
Резюме	486
Часть V. Программирование с использованием сборок .NET	487
Глава 14. Построение и конфигурирование библиотек классов	488
Определение специальных пространств имен	488
Разрешение конфликтов имен с помощью полностью заданных имен	490
Разрешение конфликтов имен с помощью псевдонимов	491
Создание вложенных пространств имен	493
Стандартное пространство имен <code>Visual Studio</code>	494
Роль сборок <code>.NET</code>	494
Сборки содействуют многократному использованию кода	495
Сборки устанавливают границы типов	495
Сборки являются единицами, поддерживающими версии	495
Сборки являются самоописательными	495
Сборки являются конфигурируемыми	496
Формат сборки <code>.NET</code>	496
Заголовок файла <code>Windows</code>	496
Заголовок файла <code>CLR</code>	498
Код CIL, метаданные типов и манифест сборки	499
Необязательные ресурсы сборки	500
Построение и потребление специальной библиотеки классов	500
Исследование манифеста	503
Исследование кода CIL	505
Исследование метаданных типов	505

Построение клиентского приложения C#	506
Построение клиентского приложения Visual Basic	508
Межъязыковое наследование в действии	509
Понятие закрытых сборок	510
Удостоверение закрытой сборки	510
Понятие процесса зондирования	510
Конфигурирование закрытых сборок	511
Роль файла App.Config	513
Понятие разделяемых сборок	514
Глобальный кеш сборки	515
Понятие строгих имен	516
Генерация строгих имен в командной строке	518
Генерация строгих имен в Visual Studio	519
Установка строго именованных сборок в GAC	520
Потребление разделяемой сборки	521
Изучение манифеста SharedCarLibClient	523
Конфигурирование разделяемых сборок	523
Замораживание текущей версии разделяемой сборки	524
Построение разделяемой сборки версии 2.0.0.0	524
Динамическое перенаправление на специфичные версии разделяемой сборки	526
Понятие сборок политик издателя	528
Отключение политики издателя	529
Элемент <codeBase>	529
Пространство имен System.Configuration	531
Документация по схеме конфигурационного файла	532
Резюме	533
Глава 15. Рефлексия типов, позднее связывание и программирование на основе атрибутов	534
Потребность в метаданных типов	534
Просмотр (частичных) метаданных для перечисления EngineState	535
Просмотр (частичных) метаданных для типа Car	536
Исследование блока TypeRef	537
Документирование определяемой сборки	538
Документирование ссылаемых сборок	538
Документирование строковых литералов	538
Понятие рефлексии	539
Класс System.Type	540
Получение информации о типе с помощью System.Object.GetType()	540
Получение информации о типе с помощью typeof()	541
Получение информации о типе с помощью System.Type.GetType()	541
Построение специального средства для просмотра метаданных	542
Рефлексия методов	542
Рефлексия полей и свойств	543
Рефлексия реализованных интерфейсов	543
Отображение разнообразных дополнительных деталей	544
Реализация метода Main()	544
Рефлексия обобщенных типов	546
Рефлексия параметров и возвращаемых значений методов	546
Динамически загружаемые сборки	547
Рефлексия разделяемых сборок	549
Позднее связывание	551

Класс <code>System.Activator</code>	552
Вызов методов без параметров	553
Вызов методов с параметрами	554
Роль атрибутов <code>.NET</code>	555
Потребители атрибутов	556
Применение атрибутов в <code>C#</code>	556
Сокращенная система обозначения атрибутов <code>C#</code>	558
Указание параметров конструктора для атрибутов	558
Атрибут <code>[Obsolete]</code> в действии	559
Построение специальных атрибутов	559
Применение специальных атрибутов	560
Синтаксис именованных свойств	560
Ограничение использования атрибутов	561
Атрибуты уровня сборки	562
Файл <code>AssemblyInfo.cs</code> , генерируемый Visual Studio	563
Рефлексия атрибутов с использованием раннего связывания	564
Рефлексия атрибутов с использованием позднего связывания	565
Практическое использование рефлексии, позднего связывания и специальных атрибутов	566
Построение расширяемого приложения	567
Построение сборки <code>CommonSnappableTypes.dll</code>	568
Построение оснастки на <code>C#</code>	568
Построение оснастки на Visual Basic	569
Построение расширяемого приложения Windows Forms	569
Резюме	573
Глава 16. Динамические типы и среда DLR	574
Роль ключевого слова <code>dynamic</code> языка <code>C#</code>	574
Вызов членов на динамически объявленных данных	576
Роль сборки <code>Microsoft.CSharp.dll</code>	577
Область применения ключевого слова <code>dynamic</code>	578
Ограничения ключевого слова <code>dynamic</code>	579
Практическое использование ключевого слова <code>dynamic</code>	579
Роль исполняющей среды динамического языка	580
Роль деревьев выражений	581
Роль пространства имен <code>System.Dynamic</code>	581
Динамический поиск в деревьях выражений во время выполнения	582
Упрощение вызовов с поздним связыванием посредством динамических типов	582
Использование ключевого слова <code>dynamic</code> для передачи аргументов	583
Упрощение взаимодействия с COM посредством динамических данных	585
Роль основных сборок взаимодействия	586
Встраивание метаданных взаимодействия	587
Общие сложности взаимодействия с COM	587
Взаимодействие с COM с использованием динамических данных <code>C#</code>	589
Взаимодействие с COM без использования динамических данных <code>C#</code>	592
Резюме	593
Глава 17. Процессы, домены приложений и объектные контексты	594
Роль процесса Windows	594
Роль потоков	595
Взаимодействие с процессами на платформе <code>.NET</code>	597
Перечисление выполняющихся процессов	598
Исследование конкретного процесса	600
Исследование набора потоков процесса	600

Исследование набора модулей процесса	602
Запуск и останов процессов программным образом	603
Управление запуском процесса с использованием класса <code>ProcessStartInfo</code>	604
Домены приложений .NET	605
Класс <code>System.AppDomain</code>	606
Взаимодействие со стандартным доменом приложения	608
Перечисление загруженных сборок	609
Получение уведомлений о загрузке сборок	610
Создание новых доменов приложений	610
Загрузка сборок в специальные домены приложений	612
Выгрузка доменов приложений программным образом	613
Контекстные границы объектов	614
Контекстно-свободные и контекстно-связанные типы	615
Определение контекстно-связанного объекта	616
Исследование контекста объекта	616
Итоговые сведения о процессах, доменах приложений и контекстах	618
Резюме	618
Глава 18. Язык CIL и роль динамических сборок	619
Причины для изучения грамматики языка CIL	619
Директивы, атрибуты и коды операций CIL	620
Роль директив CIL	621
Роль атрибутов CIL	621
Роль кодов операций CIL	621
Разница между кодами операций и их мнемоническими эквивалентами в CIL	622
Заталкивание и выталкивание: основанная на стеке природа CIL	623
Возвратное проектирование	624
Роль меток в коде CIL	627
Взаимодействие с CIL: модификация файла <code>*.il</code>	628
Компиляция кода CIL с помощью <code>ilasm.exe</code>	629
Роль инструмента <code>peverify.exe</code>	630
Директивы и атрибуты CIL	630
Указание ссылок на внешние сборки в CIL	631
Определение текущей сборки в CIL	631
Определение пространств имен в CIL	632
Определение типов классов в CIL	632
Определение и реализация интерфейсов в CIL	633
Определение структур в CIL	634
Определение перечислений в CIL	634
Определение обобщений в CIL	635
Компиляция файла <code>CILTypes.il</code>	636
Соответствия между типами данных в библиотеке базовых классов .NET, C# и CIL	636
Определение членов типов в CIL	637
Определение полей данных в CIL	637
Определение конструкторов типа в CIL	638
Определение свойств в CIL	638
Определение параметров членов	639
Исследование кодов операций CIL	640
Директива <code>.maxstack</code>	642
Объявление локальных переменных в CIL	642
Отображение параметров на локальные переменные в CIL	643
Скрытая ссылка <code>this</code>	643
Представление итерационных конструкций в CIL	644

Построение сборки .NET на CIL	645
Построение сборки CILCars.dll	645
Построение сборки CILCarClient.exe	647
Динамические сборки	649
Исследование пространства имен System.Reflection.Emit	650
Роль типа System.Reflection.Emit.ILGenerator	650
Выпуск динамической сборки	651
Выпуск сборки и набора модулей	653
Роль типа ModuleBuilder	654
Выпуск типа HelloClass и строковой переменной-члена	655
Выпуск конструкторов	656
Выпуск метода SayHello()	657
Использование динамически сгенерированной сборки	657
Резюме	658
Часть VI. Введение в библиотеки базовых классов .NET	659
Глава 19. Многопоточное, параллельное и асинхронное программирование	660
Отношения между процессом, доменом приложения, контекстом и потоком	661
Проблема параллелизма	662
Роль синхронизации потоков	662
Краткий обзор делегатов .NET	663
Асинхронная природа делегатов	665
Методы BeginInvoke() и EndInvoke()	665
Интерфейс System.IAsyncResult	665
Асинхронный вызов метода	666
Синхронизация вызывающего потока	667
Роль делегата AsyncCallback	668
Роль класса AsyncResult	670
Передача и получение специальных данных состояния	671
Пространство имен System.Threading	672
Класс System.Threading.Thread	672
Получение статистики о текущем потоке выполнения	674
Свойство Name	674
Свойство Priority	675
Ручное создание вторичных потоков	676
Работа с делегатом ThreadStart	676
Работа с делегатом ParametrizedThreadStart	678
Класс AutoResetEvent	679
Потоки переднего плана и фоновые потоки	680
Проблемы параллелизма	681
Синхронизация с использованием ключевого слова lock языка C#	683
Синхронизация с использованием типа System.Threading.Monitor	685
Синхронизация с использованием типа System.Threading.Interlocked	686
Синхронизация с использованием атрибута [Synchronization]	687
Программирование с использованием обратных вызовов Timer	688
Пул потоков CLR	689
Параллельное программирование с использованием TPL	691
Пространство имен System.Threading.Tasks	691
Роль класса Parallel	692
Обеспечение параллелизма данных с помощью класса Parallel	692
Доступ к элементам пользовательского интерфейса во вторичных потоках	694
Класс Task	695

Обработка запроса на отмену	696
Обеспечение параллелизма задач с помощью класса <code>Parallel</code>	697
Запросы <code>Parallel LINQ (PLINQ)</code>	700
Создание запроса <code>PLINQ</code>	701
Отмена запроса <code>PLINQ</code>	702
Асинхронные вызовы с помощью ключевого слова <code>async</code>	703
Знакомство с ключевыми словами <code>async</code> и <code>await</code> языка <code>C#</code>	703
Соглашения об именовании асинхронных методов	705
Асинхронные методы, возвращающие <code>void</code>	706
Асинхронные методы с множеством контекстов <code>await</code>	706
Модернизация примера <code>AddWithThreads</code> с использованием <code>asyncn/await</code>	707
Резюме	709
Глава 20. Файловый ввод-вывод и сериализация объектов	710
Исследование пространства имен <code>System.IO</code>	710
Классы <code>Directory (DirectoryInfo)</code> и <code>File (FileInfo)</code>	712
Абстрактный базовый класс <code>FileSystemInfo</code>	712
Работа с типом <code>DirectoryInfo</code>	713
Перечисление файлов с помощью типа <code>DirectoryInfo</code>	714
Создание подкаталогов с помощью типа <code>DirectoryInfo</code>	715
Работа с типом <code>Directory</code>	716
Работа с типом <code>DriveInfo</code>	717
Работа с классом <code>FileInfo</code>	718
Метод <code>FileInfo.Create()</code>	719
Метод <code>FileInfo.Open()</code>	720
Методы <code>FileOpen.OpenRead()</code> и <code>FileInfo.OpenWrite()</code>	721
Метод <code>FileInfo.OpenText()</code>	721
Методы <code>FileInfo.CreateText()</code> и <code>FileInfo.AppendText()</code>	722
Работа с типом <code>File</code>	722
Дополнительные члены <code>File</code>	723
Абстрактный класс <code>Stream</code>	724
Работа с классом <code>FileStream</code>	725
Работа с классами <code>StreamWriter</code> и <code>StreamReader</code>	726
Запись в текстовый файл	727
Чтение из текстового файла	728
Прямое создание объектов типа <code>StreamWriter/StreamReader</code>	729
Работа с классами <code>StringWriter</code> и <code>StringReader</code>	729
Работа с классами <code>BinaryWriter</code> и <code>BinaryReader</code>	731
Программное слежение за файлами	732
Понятие сериализации объектов	734
Роль графов объектов	736
Конфигурирование объектов для сериализации	737
Определение сериализируемых типов	737
Открытые поля, закрытые поля и открытые свойства	738
Выбор формatera сериализации	739
Интерфейсы <code>IFormatter</code> и <code>IRemotingFormatter</code>	739
Точность типов среди форматов	740
Сериализация объектов с использованием <code>BinaryFormatter</code>	741
Десериализация объектов с использованием <code>BinaryFormatter</code>	742
Сериализация объектов с использованием <code>SoapFormatter</code>	743
Сериализация объектов с использованием <code>XmlSerializer</code>	744
Управление генерацией данных XML	745

Сериализация коллекций объектов	746
Настройка процесса сериализации SOAP и двоичной сериализации	748
Углубленный взгляд на сериализацию объектов	749
Настройка сериализации с использованием <code>ISerializable</code>	749
Настройка сериализации с использованием атрибутов	752
Резюме	753
Глава 21. ADO.NET, часть I: подключенный уровень	754
Высокоуровневое определение ADO.NET	754
Три грани ADO.NET	755
Поставщики данных ADO.NET	756
Поставщики данных ADO.NET производства Microsoft	758
Замечания относительно сборки <code>System.Data.OracleClient.dll</code>	759
Получение сторонних поставщиков данных ADO.NET	760
Дополнительные пространства имен ADO.NET	760
Типы из пространства имен <code>System.Data</code>	761
Роль интерфейса <code>IDbConnection</code>	762
Роль интерфейса <code>IDbTransaction</code>	762
Роль интерфейса <code>IDbCommand</code>	762
Роль интерфейсов <code>IDbDataParameter</code> и <code>IDataParameter</code>	763
Роль интерфейсов <code>IDbDataAdapter</code> и <code>IDataAdapter</code>	763
Роль интерфейсов <code>IDataReader</code> и <code>IDataRecord</code>	764
Абстрагирование поставщиков данных с использованием интерфейсов	765
Увеличение гибкости с использованием конфигурационных файлов приложения	767
Создание базы данных <code>AutoLot</code>	768
Создание таблицы <code>Inventory</code>	768
Добавление тестовых записей в таблицу <code>Inventory</code>	771
Создание хранимой процедуры <code>GetPetName()</code>	772
Создание таблиц <code>Customers</code> и <code>Orders</code>	773
Создание отношений между таблицами в Visual Studio	774
Модель фабрики поставщиков данных ADO.NET	775
Полный пример фабрики поставщиков данных	777
Потенциальный недостаток модели фабрики поставщиков данных	779
Элемент <code><connectionStrings></code>	780
Понятие подключенного уровня ADO.NET	781
Работа с объектами подключения	782
Работа с объектами <code>ConnectionStringBuilder</code>	784
Работа с объектами команд	785
Работа с объектами чтения данных	787
Получение множества результирующих наборов с использованием объекта чтения данных	788
Построение многократно используемой библиотеки доступа к данным	789
Добавление логики подключения	790
Добавление логики вставки	791
Добавление логики удаления	792
Добавление логики обновления	792
Добавление логики выборки	793
Работа с параметризованными объектами команд	794
Указание параметров с использованием типа <code>DbParameter</code>	794
Выполнение хранимой процедуры	796
Создание приложения с консольным пользовательским интерфейсом	798
Реализация метода <code>Main()</code>	798
Реализация метода <code>ShowInstructions()</code>	800

Реализация метода ListInventory()	800
Реализация метода DeleteCar()	801
Реализация метода InsertNewCar()	801
Реализация метода UpdateCarPetName()	802
Реализация метода LockUpPetName()	802
Понятие транзакций базы данных	803
Основные члены объекта транзакции ADO.NET	804
Добавление таблицы CreditRisks в базу данных AutoLot	805
Добавление метода транзакции в InventoryDAL	805
Тестирование транзакции базы данных	807
Резюме	808
Глава 22. ADO.NET, часть II: автономный уровень	809
Понятие автономного уровня ADO.NET	810
Роль объектов DataSet	811
Основные свойства класса DataSet	812
Основные методы класса DataSet	812
Построение объекта DataSet	813
Работа с объектами DataColumn	813
Построение объекта DataColumn	815
Включение автоинкрементных полей	815
Добавление объектов DataColumn в DataTable	816
Работа с объектами DataRow	816
Свойство RowState	818
Свойство DataRowVersion	819
Работа с объектами DataTable	820
Вставка объектов DataTable в DataSet	821
Получение данных из объекта DataSet	821
Обработка данных в DataTable с использованием объектов DataTableReader	822
Сериализация объектов DataTable и DataSet в формате XML	823
Сериализация объектов DataTable и DataSet в двоичном формате	824
Привязка объектов DataTable к графическому пользовательскому интерфейсу Windows Forms	825
Заполнение DataTable из обобщенного List<T>	826
Удаление строк из DataTable	828
Выборка строк на основе критерия фильтрации	830
Обновление строк в DataTable	832
Работа с типом DataView	832
Работа с адаптерами данных	834
Простой пример адаптера данных	835
Отображение имен из базы данных на дружественные к пользователю имена	836
Добавление функциональности автономного уровня в AutoLotDAL.dll	837
Определение начального класса	837
Конфигурирование адаптера данных с использованием SqlCommandBuilder	838
Реализация метода GetAllInventory()	839
Реализация метода UpdateInventory()	839
Установка номера версии	840
Тестирование функциональности автономного уровня	840
Объекты DataSet с несколькими таблицами и отношения между данными	841
Подготовка адаптеров данных	842
Построение отношений между таблицами	844
Обновление таблиц базы данных	844
Навигация между связанными таблицами	845

Инструменты Windows Forms для визуального конструирования баз данных	846
Визуальное проектирование элемента управления DataGridView	847
Сгенерированный файл App.config	850
Исследование строго типизированного класса DataSet	851
Исследование строго типизированного класса DataTable	852
Исследование строго типизированного класса DataRow	853
Исследование строго типизированного адаптера данных	853
Завершение приложения Windows Forms	854
Изоляция строго типизированного кода работы с базой данных в библиотеке классов	855
Просмотр сгенерированного кода	855
Выборка данных с помощью сгенерированного кода	856
Вставка данных с помощью сгенерированного кода	857
Удаление данных с помощью сгенерированного кода	858
Вызов хранимой процедуры с помощью сгенерированного кода	859
Программирование с помощью LINQ to DataSet	860
Роль библиотеки расширений DataSet	861
Получение объекта DataTable, совместимого с LINQ	862
Роль расширяющего метода DataRowExtensions.Field<T>()	864
Заполнение новых объектов DataTable из запросов LINQ	864
Резюме	865
Глава 23. ADO.NET, часть III: Entity Framework	866
Роль Entity Framework	867
Роль сущностей	869
Строительные блоки Entity Framework	871
Роль класса DbContext	871
Прием Code First from Database	875
Генерация модели	875
Что мы получили?	877
Изменение стандартных отображений	880
Добавление сгенерированных классов модели	881
Использование классов модели в коде	881
Вставка записи	881
Выборка записей	882
Роль навигационных свойств	885
Удаление записи	888
Обновление записи	889
Обработка изменений в базе данных	889
Сборка AutoLotDAL версии 4	890
Аннотации данных Entity Framework	891
Добавление или обновление классов модели	892
Добавление класса, производного от DbContext	897
Добавление хранилищ	899
Инициализация базы данных	906
Тестирование сборки AutoLotDAL	907
Вывод всех записей из Inventory	908
Добавление записей в Inventory	908
Редактирование записей	909
Использование навигационных свойств	909
Многотабличные действия и неявные транзакции	910
Миграции Entity Framework	912
Обновление модели	912
Тестирование приложения	914
Понятие миграций EF	915

Создание миграции базового уровня	915
Начальное заполнение базы данных	918
Возвращение к тесту транзакций	920
Параллелизм	920
Корректировка классов хранилищ	921
Тестирование параллелизма	921
Перехват	922
Интерфейс IDbCommandInterceptor	922
Добавление перехвата в AutoLotDAL	923
Регистрация перехватчика	924
Добавление перехватчика DatabaseLogger	924
События ObjectMaterialized и SavingChanges	925
Доступ к объектному контексту	925
Событие ObjectMaterialized	925
Событие SavingChanges	925
Развертывание базы данных в системе SQL Server Express	926
Резюме	927
Глава 24. Введение в LINQ to XML	928
История о двух API-интерфейсах XML	928
Интерфейс LINQ to XML как лучшая модель DOM	930
Синтаксис литералов Visual Basic как лучший способ работы с LINQ to XML	930
Члены пространства имен System.Xml.Linq	932
Осевые методы LINQ to XML	933
Странность класса XElement (и XElement)	934
Работа с XElement и XDocument	935
Генерация документов из массивов и контейнеров	937
Загрузка и разбор содержимого XML	938
Манипулирование документом XML в памяти	939
Построение пользовательского интерфейса для приложения LINQ to XML	939
Импортирование файла Inventory.xml	940
Определение вспомогательного класса LINQ to XML	940
Связывание пользовательского интерфейса и вспомогательного класса	941
Резюме	942
Глава 25. Введение в Windows Communication Foundation	943
Собрание API-интерфейсов распределенных вычислений	943
Роль DCOM	944
Роль служб COM+/Enterprise Services	945
Роль MSMQ	946
Роль .NET Remoting	946
Роль веб-служб XML	947
Роль WCF	948
Обзор функциональных возможностей WCF	949
Обзор архитектуры, ориентированной на службы	949
WCF: итоги	950
Исследование основных сборок WCF	951
Шаблоны проектов WCF в Visual Studio	952
Шаблон проекта WCF Service для веб-сайта	953
Базовая структура приложения WCF	953
Адрес, привязка и контракт в WCF	955
Понятие контрактов WCF	955
Понятие привязок WCF	956

Понятие адресов WCF	959
Построение службы WCF	960
Атрибут [ServiceContract]	961
Атрибут [OperationContract]	962
Типы служб как контракты операций	962
Хостинг службы WCF	963
Установка ABC внутри файла App.config	964
Кодирование с использованием типа ServiceHost	964
Указание базовых адресов	965
Подробный анализ типа ServiceHost	966
Подробный анализ элемента <system.serviceModel>	968
Включение обмена метаданными	968
Построение клиентского приложения WCF	971
Генерация кода прокси с использованием svcutil.exe	972
Генерация кода прокси в Visual Studio	973
Конфигурирование привязки на основе TCP	974
Упрощение конфигурационных настроек	976
Использование стандартных конечных точек	976
Открытие доступа к одиночной службе WCF, использующей множество привязок	977
Изменение параметров привязки WCF	978
Использование конфигурации стандартного поведения MEX	980
Обновление клиентского прокси и выбор привязки	981
Использование шаблона проекта WCF Service Library	982
Построение простой математической службы	982
Тестирование службы WCF с помощью WcfTestClient.exe	983
Изменение конфигурационных файлов с использованием SvcConfigEditor.exe	984
Хостинг службы WCF внутри Windows-службы	985
Указание ABC в коде	985
Включение MEX	987
Создание программы установки для Windows-службы	988
Установка Windows-службы	988
Асинхронный вызов службы из клиента	989
Проектирование контрактов данных WCF	991
Использование веб-ориентированного шаблона проекта WCF Service	992
Реализация контракта службы	994
Роль файла *.svc	995
Содержимое файла Web.config	995
Тестирование службы	996
Резюме	996
Часть VII. Windows Presentation Foundation	997
Глава 26. Введение в Windows Presentation Foundation и XAML	998
Мотивация, лежащая в основе WPF	999
Унификация несходных API-интерфейсов	999
Обеспечение разделения ответственности через XAML	1000
Обеспечение оптимизированной модели визуализации	1001
Упрощение программирования сложных пользовательских интерфейсов	1001
Различные варианты приложений WPF	1002
Традиционные настольные приложения	1002
Приложения WPF на основе навигации	1003
Приложения XBP	1004
Отношения между WPF и Silverlight	1005

Исследованиеборок WPF	1006
Роль класса Application	1008
Роль класса Window	1009
Роль класса System.Windows.Controls.ContentControl	1010
Роль класса System.Windows.Controls.Control	1011
Роль класса System.Windows.FrameworkElement	1011
Роль класса System.Windows.UIElement	1012
Роль класса System.Windows.Media.Visual	1013
Роль класса System.Windows.DependencyObject	1013
Роль класса System.Windows.Threading.DispatcherObject	1013
Построение приложения WPF без XAML	1013
Создание строго типизированного класса окна	1015
Создание простого пользовательского интерфейса	1016
Взаимодействие с данными уровня приложения	1017
Обработка закрытия объекта Window	1018
Перехват событий мыши	1019
Перехват событий клавиатуры	1020
Построение приложения WPF с использованием только XAML	1021
Определение объекта Window в XAML	1022
Определение объекта Application в XAML	1024
Обработка файлов XAML с использованием msbuild.exe	1024
Трансформация разметки в сборку .NET	1026
Отображение разметки XAML окна на код C#	1026
Роль BAML	1028
Отображение разметки XAML приложения на код C#	1029
Итоговые замечания о процессе трансформирования XAML в сборку	1029
Синтаксис XAML для WPF	1030
Введение в KaXaml	1030
Пространства имен XML и “ключевые слова” XAML	1031
Управление видимостью классов и переменных-членов	1033
Элементы XAML, атрибуты XAML и преобразователи типов	1034
Понятие синтаксиса “свойство-элемент” в XAML	1035
Понятие присоединяемых свойств XAML	1036
Понятие расширений разметки XAML	1037
Построение приложений WPF с использованием файлов отделенного кода	1039
Добавление файла кода для класса MainWindow	1039
Добавление файла кода для класса MyApp	1040
Обработка файлов кода с помощью msbuild.exe	1040
Построение приложений WPF с использованием Visual Studio	1041
Шаблоны проектов WPF	1041
Панель инструментов и визуальный конструктор/редактор XAML	1043
Установка свойств с использованием окна Properties	1043
Обработка событий с использованием окна Properties	1044
Обработка событий в редакторе XAML	1045
Окно Document Outline	1046
Просмотр автоматически сгенерированных файлов кода	1046
Построение специального редактора XAML с помощью Visual Studio	1047
Проектирование графического пользовательского интерфейса окна	1047
Реализация события Loaded	1048
Реализация события Click объекта Button	1049
Реализация события Closed	1050
Тестирование приложения	1051
Изучение документации WPF	1052
Резюме	1052

Глава 27. Программирование с использованием элементов управления WPF	1053
Обзор основных элементов управления WPF	1053
Элементы управления Ink API	1054
Элементы управления документов WPF	1054
Общие диалоговые окна WPF	1055
Подробные сведения находятся в документации	1055
Краткий обзор визуального конструктора WPF в Visual Studio	1056
Работа с элементами управления WPF в Visual Studio	1056
Работа с окном Document Outline	1057
Управление компоновкой содержимого с использованием панелей	1058
Позиционирование содержимого внутри панелей Canvas	1060
Позиционирование содержимого внутри панелей WrapPanel	1062
Позиционирование содержимого внутри панелей StackPanel	1063
Позиционирование содержимого внутри панелей Grid	1064
Позиционирование содержимого внутри панелей DockPanel	1066
Включение прокрутки в типах панелей	1067
Конфигурирование панелей с использованием визуальных конструкторов Visual Studio	1068
Построение окна с использованием вложенных панелей	1070
Построение системы меню	1071
Построение панели инструментов	1073
Построение строки состояния	1074
Завершение проектирования пользовательского интерфейса	1074
Реализация обработчиков событий MouseEnter/MouseLeave	1075
Реализация логики проверки правописания	1076
Понятие команд WPF	1076
Внутренние объекты команд	1077
Подключение команд к свойству Command	1078
Подключение команд к произвольным действиям	1078
Работа с командами Open и Save	1080
Понятие маршрутизируемых событий	1082
Роль пузырьковых маршрутизируемых событий	1083
Продолжение или прекращение пузырькового распространения	1084
Роль туннельных маршрутизируемых событий	1084
Более глубокий взгляд на API-интерфейсы и элементы управления WPF	1086
Работа с элементом управления TabControl	1086
Построение вкладки Ink API	1088
Проектирование панели инструментов	1088
Элемент управления RadioButton	1090
Обработка событий для вкладки Ink API	1091
Элемент управления InkCanvas	1092
Элемент управления ComboBox	1094
Сохранение, загрузка и очистка данных InkCanvas	1096
Введение в интерфейс Documents API	1097
Блочные элементы и встроенные элементы	1097
Диспетчеры компоновки документа	1097
Построение вкладки Documents	1098
Наполнение FlowDocument с помощью кода	1099
Включение аннотаций и “клеяких” заметок	1101
Сохранение и загрузка потокового документа	1102
Введение в модель привязки данных WPF	1103
Построение вкладки Data Binding	1104

Установка привязки данных с использованием Visual Studio	1104
Свойство DataContext	1105
Преобразование данных с использованием IValueConverter	1106
Установка привязок данных в коде	1108
Построение вкладки DataGrid	1108
Роль свойств зависимости	1109
Исследование существующего свойства зависимости	1111
Важные замечания относительно оболочек свойств CLR	1114
Построение специального свойства зависимости	1115
Добавление процедуры проверки достоверности данных	1118
Реагирование на изменение свойства	1118
Резюме	1119
Глава 28. Службы визуализации графики WPF	1120
Службы графической визуализации WPF	1120
Варианты графической визуализации WPF	1121
Визуализация графических данных с использованием фигур	1122
Добавление прямоугольников, эллипсов и линий на поверхность Canvas	1124
Удаление прямоугольников, эллипсов и линий с поверхности Canvas	1127
Работа с элементами Polyline и Polygon	1128
Работа с элементом Path	1129
Кисти и перья WPF	1132
Конфигурирование кистей с использованием Visual Studio	1132
Конфигурирование кистей в коде	1134
Конфигурирование перьев	1135
Применение графических трансформаций	1135
Первый взгляд на трансформации	1136
Трансформация данных Canvas	1137
Работа с редактором трансформаций Visual Studio	1140
Построение начальной компоновки	1140
Применение трансформаций на этапе проектирования	1141
Трансформация холста в коде	1142
Визуализация графических данных с использованием рисунков и геометрий	1143
Построение кисти DrawingBrush с использованием геометрий	1144
Рисование с помощью DrawingBrush	1145
Включение типов Drawing в DrawingImage	1146
Работа с векторными изображениями	1147
Преобразование файла с векторной графикой в XAML	1147
Импортирование графических данных в проект WPF	1149
Взаимодействие с объектами изображения	1150
Визуализация графических данных с использованием визуального уровня	1150
Базовый класс Visual и производные дочерние классы	1151
Первый взгляд на класс DrawingVisual	1151
Визуализация графических данных в специальном диспетчере компоновки	1153
Реагирование на операции проверки попадания	1155
Резюме	1157
Глава 29. Ресурсы, анимация, стили и шаблоны WPF	1158
Система ресурсов WPF	1158
Работа с двоичными ресурсами	1159
Программная загрузка изображения	1161
Работа с объектными (логическими) ресурсами	1163
Роль свойства Resources	1164

Определение ресурсов уровня окна	1164
Расширение разметки {StaticResource}	1166
Расширение разметки {DynamicResource}	1167
Ресурсы уровня приложения	1167
Определение объединенных словарей ресурсов	1169
Определение сборки, включающей только ресурсы	1170
Службы анимации WPF	1171
Роль классов анимации	1172
Свойства To, From и By	1173
Роль базового класса Timeline	1173
Реализация анимации в коде C#	1174
Управление темпом анимации	1175
Запуск в обратном порядке и циклическое выполнение анимации	1176
Реализация анимации в разметке XAML	1176
Роль раскадровок	1177
Роль триггеров событий	1178
Анимация с использованием дискретных ключевых кадров	1178
Роль стилей WPF	1179
Определение и применение стиля	1180
Переопределение настроек стиля	1181
Ограничение применения стиля с помощью TargetType	1181
Автоматическое применение стиля с помощью TargetType	1182
Создание подклассов существующих стилей	1183
Определение стилей с триггерами	1183
Определение стилей с несколькими триггерами	1184
Анимированные стили	1184
Применение стилей в коде	1185
Логические деревья, визуальные деревья и стандартные шаблоны	1186
Программное инспектирование логического дерева	1187
Программное инспектирование визуального дерева	1189
Программное инспектирование стандартного шаблона элемента управления	1189
Построение шаблона элемента управления с помощью инфраструктуры триггеров	1192
Шаблоны как ресурсы	1193
Встраивание визуальных подсказок с использованием триггеров	1195
Роль расширения разметки {TemplateBinding}	1196
Роль класса ContentPresenter	1197
Встраивание шаблонов в стили	1198
Резюме	1199
Глава 30. Уведомления, команды, проверка достоверности и MVVM	1200
Введение в шаблон MVVM	1201
Модель	1201
Представление	1201
Модель представления	1201
Анемичные модели или модели представлений	1202
Система уведомлений привязки WPF	1202
Наблюдаемые модели и коллекции	1203
Добавление привязок и данных	1204
Изменение данных об автомобиле в коде	1205
Наблюдаемые модели	1206
Наблюдаемые коллекции	1209
Проверка достоверности	1217
Модификация примера для демонстрации проверки достоверности	1217

Класс Validation	1218
Варианты проверки достоверности	1219
Использование аннотаций данных	1230
Добавление аннотаций данных	1230
Контроль ошибок проверки достоверности на основе аннотаций данных	1231
Настройка свойства ErrorMessage	1232
Создание специальных команд	1234
Реализация интерфейса ICommand	1234
Изменение файла MainWindow.xaml.cs	1235
Изменение файла MainWindow.xaml	1235
Присоединение команды к объекту CommandManager	1236
Тестирование приложения	1237
Добавление оставшихся команд	1237
Полная реализация MVVM	1239
Перемещение источника данных из представления	1239
Перемещение команд в модель представления	1241
Изменение сборки AutoLotDAL для MVVM	1241
Изменение моделей AutoLotDAL	1241
Полный пример MVVM	1244
Использование события ObjectMaterialized вместе с Entity Framework	1245
Резюме	1245
Часть VIII. ASP.NET	1247
Глава 31. Введение в ASP.NET Web Forms	1248
Роль протокола HTTP	1248
Цикл “запрос/ответ” HTTP	1249
HTTP является протоколом без хранения состояния	1249
Веб-приложения и веб-серверы	1250
Роль виртуальных каталогов IIS	1250
Веб-сервер IIS Express	1251
Роль языка HTML	1251
Структура документа HTML	1252
Роль форм HTML	1253
Инструменты визуального конструктора разметки HTML в Visual Studio	1253
Построение формы HTML	1255
Роль сценариев клиентской стороны	1257
Пример сценария клиентской стороны	1258
Обратная отправка веб-серверу	1258
Обратные отправки в Web Forms	1259
Обзор API-интерфейса Web Forms	1260
Основные возможности Web Forms 2.0 и последующих версий	1261
Основные возможности Web Forms 3.5 (и .NET 3.5 SP1) и последующих версий	1262
Основные возможности Web Forms 4.0	1262
Основные возможности Web Forms 4.5 и Web Forms 4.6	1263
Построение однофайловой веб-страницы Web Forms	1264
Добавление ссылки на сборку AutoLotDAL.dll	1265
Проектирование пользовательского интерфейса	1267
Добавление логики доступа к данным	1268
Роль директив ASP.NET	1270
Анализ блока <script>	1270
Анализ объявлений элементов управления ASP.NET	1271

Построение веб-страницы ASP.NET с использованием файлов кода	1272
Ссылка на проект AutoLotDAL	1273
Обновление файла кода	1274
Отладка и трассировка страниц ASP.NET	1274
Сравнение веб-сайтов и веб-приложений ASP.NET	1275
Включение поддержки C# 6 для веб-сайтов ASP.NET	1276
Структура каталогов веб-сайта ASP.NET	1276
Ссылка на сборки	1277
Роль папки App_Code	1278
Цепочка наследования типа Page	1278
Взаимодействие с входящим HTTP-запросом	1279
Получение статистики о браузере	1281
Доступ к входным данным формы	1283
Свойство IsPostBack	1283
Взаимодействие с исходящим HTTP-ответом	1284
Выпуск содержимого HTML	1285
Перенаправление пользователей	1285
Жизненный цикл страницы Web Forms	1286
Роль атрибута AutoEventWireup	1287
Событие Error	1288
Роль файла Web.config	1289
Утилиты администрирования веб-сайтов ASP.NET	1290
Резюме	1290
Глава 32. Веб-элементы управления, мастер-страницы и темы ASP.NET	1291
Природа веб-элементов управления	1291
Обработка события серверной стороны	1292
Свойство AutoPostBack	1293
Базовые классы Control и WebControl	1294
Перечисление содержащихся элементов управления	1295
Динамическое добавление и удаление элементов управления	1297
Взаимодействие с динамически созданными элементами управления	1298
Функциональность базового класса WebControl	1298
Основные категории элементов управления Web Forms	1299
Несколько слов о пространстве имен System.Web.UI.HtmlControls	1301
Документация по веб-элементам управления	1301
Построение веб-сайта Web Forms для работы с автомобилями	1302
Работа с мастер-страницами Web Forms	1302
Определение стандартной страницы содержимого	1308
Проектирование страницы содержимого Inventory.aspx	1310
Добавление сборки AutoLotDAL и инфраструктуры Entity Framework в проект AspNetCarsSite	1311
Наполнение данными элемента управления GridView	1311
Включение редактирования на месте	1312
Включение сортировки и разбиения на страницы	1314
Включение фильтрации	1315
Проектирование страницы содержимого BuildCar.aspx	1317
Роль элементов управления проверки достоверности	1318
Включение поддержки проверки достоверности с помощью кода JavaScript клиентской стороны	1321
Элемент управления RequiredFieldValidator	1321
Элемент управления RegularExpressionValidator	1322

Элемент управления RangeValidator	1322
Элемент управления CompareValidator	1322
Создание итоговой панели проверки достоверности	1323
Определение групп проверки достоверности	1324
Проверка достоверности с помощью аннотаций данных	1325
Работа с темами	1330
Файлы * .skin	1331
Применение тем ко всему сайту	1333
Применение тем на уровне страницы	1333
Свойство SkinID	1333
Назначение тем программным образом	1334
Резюме	1335
Глава 33. Управление состоянием в ASP.NET	1336
Проблема поддержки состояния	1336
Приемы управления состоянием ASP.NET	1338
Роль состояния представления ASP.NET	1339
Демонстрация работы с состоянием представления	1339
Добавление специальных данных состояния представления	1341
Роль файла Global . asax	1342
Глобальный обработчик исключений “последнего шанса”	1343
Базовый класс HttpApplication	1344
Разница между состоянием приложения и состоянием сеанса	1344
Поддержка данных состояния уровня приложения	1345
Модификация данных приложения	1347
Обработка прекращения работы веб-приложения	1348
Работа с кешем приложения	1348
Использование кеширования данных	1349
Модификация файла * . aspx	1351
Поддержка данных сеанса	1353
Дополнительные члены класса HttpSessionState	1355
Cookie-наборы	1356
Создание cookie-наборов	1356
Чтение входящих cookie-данных	1357
Роль элемента <sessionState>	1358
Хранение данных сеанса на сервере состояния сеансов ASP.NET	1358
Хранение информации о сеансах в выделенной базе данных	1359
Введение в API-интерфейс ASP.NET Profile	1360
База данных ASPNETDB . mdf	1360
Определение пользовательского профиля в файле Web . config	1361
Доступ к данным профиля в коде	1362
Группирование данных профиля и сохранение специальных объектов	1364
Резюме	1366
Глава 34. ASP.NET MVC и ASP.NET Web API	1367
Введение в шаблон MVC	1367
Модель	1368
Представление	1368
Контроллер	1368
Почему появилась инфраструктура MVC?	1368
Появление ASP.NET MVC	1369

Построение первого приложения ASP.NET MVC	1370
Мастер создания проекта	1370
Компоненты базового проекта MVC	1372
Модернизация пакетов NuGet до текущих версий	1378
Пробный запуск сайта	1379
Маршрутизация	1380
Шаблоны URL	1380
Создание маршрутов для страниц Contact и About	1381
Перенаправление пользователей с применением маршрутизации	1382
Добавление библиотеки AutoLotDAL	1382
Контроллеры и действия	1383
Добавление контроллера Inventory	1384
Исследование шаблонных представлений	1385
Контроллеры MVC	1386
Представления MVC	1394
Механизм представлений Razor	1394
Компоновки	1397
Частичные представления	1398
Отправка данных представлению	1399
Представление Index	1400
Представление Details	1404
Представление Create	1405
Представление Delete	1407
Представление Edit	1409
Проверка достоверности	1410
Завершение пользовательского интерфейса	1412
Заключительное слово по поводу ASP.NET MVC	1413
Введение в ASP.NET Web API	1414
Добавление проекта Web API	1414
Исследование проекта Web API	1416
Конфигурирование проекта	1417
Замечание относительно JSON	1417
Добавление контроллера	1418
Обновление проекта CarLotMVC для использования CarLotWebAPI	1425
Резюме	1431
Предметный указатель	1433