

# Содержание

<b>Об авторах</b>	34
<b>Введение</b>	36
Авторы и читатели — одна команда	36
Краткий обзор книги	36
Часть I. Введение в C# и платформу .NET	37
Часть II. Основы программирования на C#	37
Часть III. Объектно-ориентированное программирование на C#	38
Часть IV. Дополнительные конструкции программирования на C#	38
Часть V. Программирование с использованием сборок .NET	40
Часть VI. Введение в библиотеки базовых классов .NET	41
Часть VII. Windows Presentation Foundation	42
Часть VIII. ASP.NET	43
Часть IX. .NET Core	44
Загружаемые приложения	45
Исходный код примеров	45
Ждем ваших отзывов!	46
<b>Часть I. Введение в C# и платформу .NET</b>	47
<b>Глава 1. Философия .NET</b>	48
Начальное знакомство с платформой .NET	48
Некоторые основные преимущества платформы .NET	49
Введение в строительные блоки платформы .NET (CLR, CTS и CLS)	50
Роль библиотек базовых классов	50
Что привносит язык C#	51
Сравнение управляемого и неуправляемого кода	54
Другие языки программирования, ориентированные на .NET	54
Жизнь в многоязычном мире	55
Обзор сборок .NET	55
Роль языка CIL	57
Роль метаданных типов .NET	59
Роль манифеста сборки	60
Понятие общей системы типов (CTS)	61
Типы классов CTS	61
Типы интерфейсов CTS	62
Типы структур CTS	62
Типы перечислений CTS	63
Типы делегатов CTS	63
Члены типов CTS	64
Встроенные типы данных CTS	64
Понятие общезыковой спецификации (CLS)	65
Обеспечение совместимости с CLS	66
Понятие общезыковой исполняющей среды (CLR)	67
Различия между сборками, пространствами имен и типами	68
Роль корневого пространства имен Microsoft	71
Доступ к пространству имен программным образом	72
Ссылка на внешние сборки	73

Исследование сборки с помощью <code>ildasm.exe</code>	74
Просмотр кода CIL	75
Просмотр метаданных типов	75
Просмотр метаданных сборки (манифеста)	76
Независимая от платформы природа .NET	77
Проект Mono	78
Xamarin	79
Microsoft .NET Core	79
Резюме	80
<b>Глава 2. Создание приложений на языке C#</b>	81
Построение приложений .NET в среде Windows	81
Установка Visual Studio 2017	82
Испытание Visual Studio 2017	84
Visual Studio 2017 Professional	95
Visual Studio 2017 Enterprise	95
Система документации .NET Framework	95
Построение приложений .NET под управлением операционной системы, отличающейся от Windows	96
Резюме	96
<b>Часть II. Основы программирования на C#</b>	97
<b>Глава 3. Главные конструкции программирования на C#: часть I</b>	98
Структура простой программы C#	98
Вариации метода <code>Main()</code>	100
Указание кода ошибки приложения	101
Обработка аргументов командной строки	102
Указание аргументов командной строки в Visual Studio	103
Интересное отступление от темы: некоторые дополнительные члены класса <code>System.Environment</code>	104
Класс <code>System.Console</code>	106
Базовый ввод-вывод с помощью класса <code>Console</code>	106
Форматирование консольного вывода	107
Форматирование числовых данных	108
Форматирование числовых данных за рамками консольных приложений	109
Системные типы данных и соответствующие ключевые слова C#	110
Объявление и инициализация переменных	111
Внутренние типы данных и операция <code>new</code>	113
Иерархия классов для типов данных	114
Члены числовых типов данных	115
Члены <code>System.Boolean</code>	116
Члены <code>System.Char</code>	116
Разбор значений из строковых данных	116
Типы <code>System.DateTime</code> и <code>System.TimeSpan</code>	118
Сборка <code>System.Numerics.dll</code>	118
Разделители групп цифр (нововведение)	120
Двоичные литералы (нововведение)	120
Работа со строковыми данными	120
Базовые манипуляции строками	121
Конкатенация строк	122

## 8 Содержание

Управляющие последовательности	123
Определение дословных строк	123
Строки и равенство	124
Модификация поведения сравнения строк	125
Строки являются неизменяемыми	126
Тип <code>System.Text.StringBuilder</code>	127
Интерполяция строк	128
Сужающие и расширяющие преобразования типов данных	130
Ключевое слово <code>checked</code>	132
Настройка проверки переполнения на уровне проекта	134
Ключевое слово <code>unchecked</code>	134
Понятие неявно типизированных локальных переменных	135
Ограничения неявно типизированных переменных	136
Неявно типизированные данные являются строго типизированными	137
Полезность неявно типизированных локальных переменных	138
Итерационные конструкции C#	139
Цикл <code>for</code>	139
Цикл <code>foreach</code>	140
Циклы <code>while</code> и <code>do/while</code>	141
Конструкции принятия решений и операции отношения/равенства	141
Оператор <code>if/else</code>	142
Операции отношения и равенства	142
Условная операция	143
Логические операции	143
Оператор <code>switch</code>	144
Резюме	149
<b>Глава 4. Главные конструкции программирования на C#: часть II</b>	<b>150</b>
Понятие массивов C#	150
Синтаксис инициализации массивов C#	151
Неявно типизированные локальные массивы	152
Определение массива объектов	153
Работа с многомерными массивами	153
Использование массивов в качестве аргументов и возвращаемых значений	154
Базовый класс <code>System.Array</code>	155
Методы и модификаторы параметров	157
Возвращаемые значения и члены, сжатые до выражений (обновление)	157
Модификаторы параметров для методов	157
Отбрасывание	158
Стандартное поведение передачи параметров по значению	158
Модификатор <code>out</code> (обновление)	159
Модификатор <code>ref</code>	161
Ссылочные локальные переменные и возвращаемые ссылочные значения (нововведения)	162
Модификатор <code>params</code>	163
Определение необязательных параметров	165
Вызов методов с использованием именованных параметров	166
Понятие перегрузки методов	168
Локальные функции (нововведение)	169
Тип <code>enum</code>	170
Управление хранилищем, лежащим в основе перечисления	171

Объявление переменных типа перечисления	172
Тип <code>System.Enum</code>	173
Динамическое выяснение пар “имя-значение” перечисления	174
Понятие структуры (как типа значения)	175
Создание переменных типа структур	177
Типы значений и ссылочные типы	178
Типы значений, ссылочные типы и операция присваивания	179
Типы значений, содержащие ссылочные типы	180
Передача ссылочных типов по значению	182
Передача ссылочных типов по ссылке	183
Заключительные детали относительно типов значений и ссылочных типов	184
Понятие типов <code>C#</code> , допускающих <code>null</code>	185
Работа с типами, допускающими <code>null</code>	186
Операция объединения с <code>null</code>	187
<code>null</code> -условная операция	188
Кортежи (нововведение)	189
Начало работы с кортежами	189
Выведение имен свойств кортежей ( <code>C# 7.1</code> )	190
Кортежи как возвращаемые значения методов	191
Использование отбрасывания с кортежами	191
Деконструирование кортежей	192
Резюме	192

## **Часть III. Объектно-ориентированное программирование на `C#`** 193

### **Глава 5. Инкапсуляция** 194

Знакомство с типом класса <code>C#</code>	194
Размещение объектов с помощью ключевого слова <code>new</code>	196
Понятие конструкторов	197
Роль стандартного конструктора	197
Определение специальных конструкторов	198
Еще раз о стандартном конструкторе	199
Роль ключевого слова <code>this</code>	201
Построение цепочки вызовов конструкторов с использованием <code>this</code>	202
Изучение потока управления конструкторов	205
Еще раз о необязательных аргументах	206
Понятие ключевого слова <code>static</code>	207
Определение статических полей данных	208
Определение статических методов	210
Определение статических конструкторов	211
Определение статических классов	213
Импортирование статических членов с применением ключевого слова <code>using</code> языка <code>C#</code>	214
Основные принципы объектно-ориентированного программирования	215
Роль инкапсуляции	215
Роль наследования	216
Роль полиморфизма	217
Модификаторы доступа <code>C#</code>	219
Стандартные модификаторы доступа	220
Модификаторы доступа и вложенные типы	220

Первый принцип ООП: службы инкапсуляции C#	221
Инкапсуляция с использованием традиционных методов доступа и изменения	222
Инкапсуляция с использованием свойств .NET	224
Использование свойств внутри определения класса	227
Свойства, допускающие только чтение и только запись	228
Еще раз о ключевом слове <code>static</code> : определение статических свойств	229
Понятие автоматических свойств	230
Взаимодействие с автоматическими свойствами	231
Автоматические свойства и стандартные значения	232
Инициализация автоматических свойств	233
Понятие синтаксиса инициализации объектов	234
Вызов специальных конструкторов с помощью синтаксиса инициализации	235
Инициализация данных с помощью синтаксиса инициализации	236
Работа с данными константных полей	237
Понятие полей, допускающих только чтение	239
Статические поля, допускающие только чтение	239
Понятие частичных классов	240
Сценарии использования для частичных классов?	241
Резюме	241
<b>Глава 6. Наследование и полиморфизм</b>	242
Базовый механизм наследования	242
Указание родительского класса для существующего класса	243
Замечание относительно множества базовых классов	245
Ключевое слово <code>sealed</code>	245
Корректировка диаграмм классов Visual Studio	246
Второй принцип ООП: детали наследования	247
Управление созданием объектов базового класса с помощью ключевого слова <code>base</code>	249
Хранение секретов семейства: ключевое слово <code>protected</code>	251
Добавление запечатанного класса	252
Реализация модели включения/делегации	253
Определения вложенных типов	254
Третий принцип ООП: поддержка полиморфизма в C#	256
Ключевые слова <code>virtual</code> и <code>override</code>	256
Переопределение виртуальных членов в IDE-среде Visual Studio	258
Запечатывание виртуальных членов	259
Абстрактные классы	260
Полиморфные интерфейсы	261
Сокрытие членов	266
Правила приведения для базовых и производных классов	268
Ключевое слово <code>as</code>	269
Ключевое слово <code>is</code> (обновление)	271
Еще раз о сопоставлении с образцом (нововведение)	272
Главный родительский класс <code>System.Object</code>	273
Переопределение метода <code>System.Object.ToString()</code>	276
Переопределение метода <code>System.Object.Equals()</code>	276
Переопределение метода <code>System.Object.GetHashCode()</code>	277
Тестирование модифицированного класса <code>Person</code>	278
Статические члены класса <code>System.Object</code>	279
Резюме	279

<b>Глава 7. Структурированная обработка исключений</b>	280
Ода ошибкам, дефектам и исключениям	280
Роль обработки исключений .NET	281
Строительные блоки обработки исключений в .NET	282
Базовый класс System.Exception	282
Простейший пример	284
Генерация общего исключения (обновление)	285
Перехват исключений	287
Конфигурирование состояния исключения	288
Свойство TargetSite	288
Свойство StackTrace	289
Свойство HelpLink	290
Свойство Data	291
Исключения уровня системы (System.SystemException)	292
Исключения уровня приложения (System.ApplicationException)	293
Построение специальных исключений, способ первый	293
Построение специальных исключений, способ второй	295
Построение специальных исключений, способ третий	296
Обработка множества исключений	297
Общие операторы catch	300
Повторная генерация исключений	300
Внутренние исключения	301
Блок finally	302
Фильтры исключений	303
Отладка необработанных исключений с использованием Visual Studio	304
Резюме	305
<b>Глава 8. Работа с интерфейсами</b>	306
Понятие интерфейсных типов	306
Сравнение интерфейсных типов и абстрактных базовых классов	307
Определение специальных интерфейсов	309
Реализация интерфейса	311
Обращение к членам интерфейса на уровне объектов	313
Получение ссылок на интерфейсы: ключевое слово as	314
Получение ссылок на интерфейсы: ключевое слово is (обновление)	314
Использование интерфейсов в качестве параметров	315
Использование интерфейсов в качестве возвращаемых значений	317
Массивы интерфейсных типов	318
Реализация интерфейсов с использованием Visual Studio	319
Явная реализация интерфейсов	320
Проектирование иерархий интерфейсов	322
Множественное наследование с помощью интерфейсных типов	324
Интерфейсы IEnumerable и IEnumerator	326
Построение итераторных методов с использованием ключевого слова yield	328
Построение именованного итератора	330
Интерфейс ICloneable	331
Более сложный пример клонирования	333
Интерфейс IComparable	335
Указание множества порядков сортировки с помощью Comparer	338
Специальные свойства и специальные типы сортировки	339
Резюме	340

<b>Часть IV. Дополнительные конструкции программирования на C#</b>	341
<b>Глава 9. Коллекции и обобщения</b>	342
Побудительные причины создания классов коллекций	342
Пространство имен <code>System.Collections</code>	344
Обзор пространства имен <code>System.Collections.Specialized</code>	346
Проблемы, присущие необобщенным коллекциям	347
Проблема производительности	347
Проблема безопасности к типам	350
Первый взгляд на обобщенные коллекции	353
Роль параметров обобщенных типов	354
Указание параметров типа для обобщенных классов и структур	355
Указание параметров типа для обобщенных членов	356
Указание параметров типов для обобщенных интерфейсов	357
Пространство имен <code>System.Collections.Generic</code>	358
Синтаксис инициализации коллекций	360
Работа с классом <code>List&lt;T&gt;</code>	361
Работа с классом <code>Stack&lt;T&gt;</code>	362
Работа с классом <code>Queue&lt;T&gt;</code>	363
Работа с классом <code>SortedSet&lt;T&gt;</code>	364
Работа с классом <code>Dictionary&lt;TKey, TValue&gt;</code>	366
Пространство имен <code>System.Collections.ObjectModel</code>	367
Работа с классом <code>ObservableCollection&lt;T&gt;</code>	367
Создание специальных обобщенных методов	369
Выведение параметров типа	371
Создание специальных обобщенных структур и классов	372
Ключевое слово <code>default</code> в обобщенном коде	373
Ограничение параметров типа	374
Примеры использования ключевого слова <code>where</code>	375
Отсутствие ограничений операций	377
Резюме	378
<b>Глава 10. Делегаты, события и лямбда-выражения</b>	379
Понятие типа делегата <code>.NET</code>	379
Определение типа делегата в C#	380
Базовые классы <code>System.MulticastDelegate</code> и <code>System.Delegate</code>	383
Пример простейшего делегата	384
Исследование объекта делегата	385
Отправка уведомлений о состоянии объекта с использованием делегатов	387
Включение группового вызова	389
Удаление целей из списка вызовов делегата	391
Синтаксис групповых преобразований методов	392
Понятие обобщенных делегатов	393
Обобщенные делегаты <code>Action&lt;&gt;</code> и <code>Func&lt;&gt;</code>	394
Понятие событий C#	396
Ключевое слово <code>event</code>	398
“За кулисами” событий	399
Прослушивание входящих событий	400
Упрощение регистрации событий с использованием Visual Studio	401
Приведение в порядок кода обращения к событиям с использованием <code>null</code> -условной операции C# 6.0	403
Создание специальных аргументов событий	403

Обобщенный делегат <code>EventHandler&lt;T&gt;</code>	405
Понятие анонимных методов C#	406
Доступ к локальным переменным	407
Понятие лямбда-выражений	408
Анализ лямбда-выражения	411
Обработка аргументов внутри множества операторов	412
Лямбда-выражения с несколькими параметрами и без параметров	413
Модернизация примера <code>CarEvents</code> с использованием лямбда-выражений	414
Лямбда-выражения и члены, сжатые до выражений (обновление)	415
Резюме	416
<b>Глава 11. Расширенные средства языка C#</b>	<b>417</b>
Понятие индексаторных методов	417
Индексация данных с использованием строковых значений	419
Перегрузка методов индексаторов	420
Многомерные индексаторы	421
Определения индексаторов в интерфейсных типах	422
Понятие перегрузки операций	422
Перегрузка бинарных операций	423
А как насчет операций <code>+=</code> и <code>--</code> ?	425
Перегрузка унарных операций	425
Перегрузка операций эквивалентности	426
Перегрузка операций сравнения	427
Финальные соображения относительно перегрузки операций	428
Понятие специальных преобразований типов	428
Повторение: числовые преобразования	428
Повторение: преобразования между связанными типами классов	428
Создание специальных процедур преобразования	429
Дополнительные явные преобразования для типа <code>Square</code>	432
Определение процедур неявного преобразования	432
Понятие расширяющих методов	434
Понятие анонимных типов	438
Определение анонимного типа	439
Анонимные типы, содержащие другие анонимные типы	443
Работа с типами указателей	444
Ключевое слово <code>unsafe</code>	445
Работа с операциями <code>*</code> и <code>&amp;</code>	447
Небезопасная (и безопасная) функция обмена	447
Доступ к полям через указатели (операция <code>-&gt;</code> )	448
Ключевое слово <code>stackalloc</code>	449
Закрепление типа посредством ключевого слова <code>fixed</code>	449
Ключевое слово <code>sizeof</code>	450
Резюме	451
<b>Глава 12. LINQ to Objects</b>	<b>452</b>
Программные конструкции, специфичные для LINQ	452
Неявная типизация локальных переменных	453
Синтаксис инициализации объектов и коллекций	454
Лямбда-выражения	454
Расширяющие методы	455
Анонимные типы	456

## 14 Содержание

Роль LINQ	456
Выражения LINQ строго типизированы	457
Основные сборки LINQ	458
Применение запросов LINQ к элементарным массивам	458
Решение с использованием расширяющих методов	460
Решение без использования LINQ	460
Выполнение рефлексии результирующего набора LINQ	461
LINQ и неявно типизированные локальные переменные	462
LINQ и расширяющие методы	463
Роль отложенного выполнения	464
Роль немедленного выполнения	465
Возвращение результатов запроса LINQ	466
Возвращение результатов LINQ посредством немедленного выполнения	467
Применение запросов LINQ к объектам коллекций	468
Доступ к содержащимся в контейнере подобъектам	469
Применение запросов LINQ к необобщенным коллекциям	470
Фильтрация данных с использованием метода OfType<T> ( )	470
Исследование операций запросов LINQ	471
Базовый синтаксис выборки	472
Получение подмножества данных	473
Проецирование новых типов данных	474
Подсчет количества с использованием класса Enumerable	475
Изменение порядка следования элементов в результирующих наборах на противоположный	475
Выражения сортировки	476
LINQ как лучшее средство построения диаграмм Венна	476
Устранение дубликатов	477
Операции агрегирования LINQ	478
Внутреннее представление операторов запросов LINQ	478
Построение выражений запросов с применением операций запросов	479
Построение выражений запросов с использованием типа Enumerable и лямбда-выражений	479
Построение выражений запросов с использованием типа Enumerable и анонимных методов	481
Построение выражений запросов с использованием типа Enumerable и низкоуровневых делегатов	481
Резюме	482
<b>Глава 13. Время существования объектов</b>	<b>483</b>
Классы, объекты и ссылки	483
Базовые сведения о времени жизни объектов	484
Код CIL для ключевого слова new	485
Установка объектных ссылок в null	486
Роль корневых элементов приложения	487
Понятие поколений объектов	489
Параллельная сборка мусора до версии .NET 4.0	490
Фоновая сборка мусора в .NET 4.0 и последующих версиях	490
Тип System.GC	491
Принудительный запуск сборщика мусора	492
Построение финализируемых объектов	494
Переопределение метода System.Object.Finalize()	495

Подробности процесса финализации	497
Построение освобождаемых объектов	498
Повторное использование ключевого слова <code>using</code> в C#	500
Создание финализируемых и освобождаемых типов	501
Формализованный шаблон освобождения	502
Ленивое создание объектов	504
Настройка процесса создания данных <code>Lazy&lt;&gt;</code>	507
Резюме	508
<b>Часть V. Программирование с использованием сборок .NET</b>	509
<b>Глава 14. Построение и конфигурирование библиотек классов</b>	510
Определение специальных пространств имен	510
Разрешение конфликтов имен с помощью полностью заданных имен	512
Разрешение конфликтов имен с помощью псевдонимов	513
Создание вложенных пространств имен	515
Стандартное пространство имен Visual Studio	516
Роль сборки .NET	516
Сборки содействуют многократному использованию кода	517
Сборки устанавливают границы типов	517
Сборки являются единицами, поддерживающими версии	517
Сборки являются самоописательными	518
Сборки являются конфигурируемыми	518
Формат сборки .NET	518
Заголовок файла Windows	519
Заголовок файла CLR	520
Код CIL, метаданные типов и манифест сборки	521
Дополнительные ресурсы сборки	522
Построение и потребление специальной библиотеки классов	522
Исследование манифеста	525
Исследование кода CIL	527
Исследование метаданных типов	528
Построение клиентского приложения C#	529
Построение клиентского приложения Visual Basic	530
Межязыковое наследование в действии	531
Понятие закрытых сборок	532
Удостоверение закрытой сборки	533
Понятие процесса зондирования	533
Конфигурирование закрытых сборок	534
Роль файла <code>App.Config</code>	535
Понятие разделяемых сборок	537
Глобальный кеш сборок	537
Понятие строгих имен	539
Генерация строгих имен в командной строке	540
Генерация строгих имен в Visual Studio	542
Установка строго именованных сборок в GAC	543
Потребление разделяемой сборки	544
Изучение манифеста <code>SharedCarLibClient</code>	546
Конфигурирование разделяемых сборок	546
Замораживание текущей версии разделяемой сборки	547
Построение разделяемой сборки версии 2.0.0.0	548
Динамическое перенаправление на специфичные версии разделяемой сборки	549

Понятие сборок политик издателя	551
Отключение политики издателя	552
Элемент <codeBase>	552
Пространство имен System.Configuration	554
Документация по схеме конфигурационного файла	555
Резюме	556
<b>Глава 15. Рефлексия типов, позднее связывание и программирование на основе атрибутов</b>	<b>557</b>
Потребность в метаданных типов	557
Просмотр (частичных) метаданных для перечисления EngineState	558
Просмотр (частичных) метаданных для типа Car	559
Исследование блока TypeRef	560
Документирование определяемой сборки	561
Документирование ссылаемых сборок	561
Документирование строковых литералов	562
Понятие рефлексии	562
Класс System.Type	563
Получение информации о типе с помощью System.Object.GetType()	564
Получение информации о типе с помощью typeof()	564
Получение информации о типе с помощью System.Type.GetType()	564
Построение специального средства для просмотра метаданных	565
Рефлексия методов	566
Рефлексия полей и свойств	566
Рефлексия реализованных интерфейсов	567
Отображение разнообразных дополнительных деталей	567
Реализация метода Main()	568
Рефлексия обобщенных типов	569
Рефлексия параметров и возвращаемых значений методов	569
Динамическая загрузка сборок	571
Рефлексия разделяемых сборок	573
Позднее связывание	575
Класс System.Activator	575
Вызов методов без параметров	577
Вызов методов с параметрами	577
Роль атрибутов .NET	578
Потребители атрибутов	579
Применение атрибутов в C#	580
Сокращенная система обозначения атрибутов C#	581
Указание параметров конструктора для атрибутов	581
Атрибут [Obsolete] в действии	582
Построение специальных атрибутов	583
Применение специальных атрибутов	583
Синтаксис именованных свойств	584
Ограничение использования атрибутов	584
Атрибуты уровня сборки	585
Файл AssemblyInfo.cs, генерируемый Visual Studio	586
Рефлексия атрибутов с использованием раннего связывания	587
Рефлексия атрибутов с использованием позднего связывания	588
Практическое использование рефлексии, позднего связывания и специальных атрибутов	589

Построение расширяемого приложения	590
Построение мультипроектного решения <code>ExtendableApp</code>	591
Построение сборки <code>CommonSnappableTypes.dll</code>	591
Добавление проектов в решение	592
Добавление ссылок на проекты	593
Построение оснастки на C#	594
Построение оснастки на Visual Basic	594
Установка запускаемого проекта	594
Установка порядка построения проектов	595
Построение расширяемого консольного приложения	597
Резюме	599
<b>Глава 16. Динамические типы и среда DLR</b>	<b>600</b>
Роль ключевого слова <code>dynamic</code> языка C#	600
Вызов членов на динамически объявленных данных	602
Роль сборки <code>Microsoft.CSharp.dll</code>	603
Область применения ключевого слова <code>dynamic</code>	604
Ограничения ключевого слова <code>dynamic</code>	605
Практическое использование ключевого слова <code>dynamic</code>	605
Роль исполняющей среды динамического языка	606
Роль деревьев выражений	607
Роль пространства имен <code>System.Dynamic</code>	607
Динамический поиск в деревьях выражений во время выполнения	608
Упрощение вызовов с поздним связыванием посредством динамических типов	608
Использование ключевого слова <code>dynamic</code> для передачи аргументов	609
Упрощение взаимодействия с COM посредством динамических данных	612
Роль основных сборок взаимодействия	613
Встраивание метаданных взаимодействия	614
Общие сложности взаимодействия с COM	615
Взаимодействие с COM с использованием динамических данных C#	616
Взаимодействие с COM без использования динамических данных C#	617
Резюме	619
<b>Глава 17. Процессы, домены приложений и объектные контексты</b>	<b>620</b>
Роль процесса Windows	620
Роль потоков	621
Взаимодействие с процессами на платформе .NET	623
Перечисление выполняющихся процессов	625
Исследование конкретного процесса	626
Исследование набора потоков процесса	626
Исследование набора модулей процесса	628
Запуск и останов процессов программным образом	629
Управление запуском процесса с использованием класса <code>ProcessStartInfo</code>	630
Домены приложений .NET	631
Класс <code>System.AppDomain</code>	632
Взаимодействие со стандартным доменом приложения	634
Перечисление загруженных сборок	635
Получение уведомлений о загрузке сборок	636
Создание новых доменов приложений	637
Загрузка сборок в специальные домены приложений	638
Выгрузка доменов приложений программным образом	639

Контекстные границы объектов	640
Контекстно-свободные и контекстно-связанные типы	641
Определение контекстно-связанного объекта	642
Исследование контекста объекта	642
Итоговые сведения о процессах, доменах приложений и контекстах	644
Резюме	644
<b>Глава 18. Язык CIL и роль динамических сборок</b>	645
Причины для изучения грамматики языка CIL	645
Директивы, атрибуты и коды операций CIL	646
Роль директив CIL	647
Роль атрибутов CIL	647
Роль кодов операций CIL	647
Разница между кодами операций и их мнемоническими эквивалентами в CIL	648
Заталкивание и выталкивание: основанная на стеке природа CIL	649
Возвратное проектирование	650
Роль меток в коде CIL	653
Взаимодействие с CIL: модификация файла *.il	654
Компиляция кода CIL с помощью ilasm.exe	655
Роль инструмента peverify.exe	656
Директивы и атрибуты CIL	656
Указание ссылок на внешние сборки в CIL	656
Определение текущей сборки в CIL	657
Определение пространств имен в CIL	658
Определение типов классов в CIL	658
Определение и реализация интерфейсов в CIL	659
Определение структур в CIL	660
Определение перечислений в CIL	660
Определение обобщений в CIL	661
Компиляция файла CILTypes.il	661
Соответствия между типами данных в библиотеке базовых классов .NET, C# и CIL	662
Определение членов типов в CIL	663
Определение полей данных в CIL	663
Определение конструкторов типа в CIL	663
Определение свойств в CIL	664
Определение параметров членов	665
Исследование кодов операций CIL	665
Директива .maxstack	667
Объявление локальных переменных в CIL	668
Отображение параметров на локальные переменные в CIL	668
Скрытая ссылка this	669
Представление итерационных конструкций в CIL	669
Построение сборки .NET на CIL	670
Построение сборки CILCars.dll	671
Построение сборки CILCarClient.exe	673
Динамические сборки	674
Исследование пространства имен System.Reflection.Emit	675
Роль типа System.Reflection.Emit.ILGenerator	676
Выпуск динамической сборки	677
Выпуск сборки и набора модулей	679
Роль типа ModuleBuilder	680

Выпуск типа <code>HelloClass</code> и строковой переменной-члена	681
Выпуск конструкторов	681
Выпуск метода <code>SayHello()</code>	682
Использование динамически сгенерированной сборки	683
Резюме	684

## **Часть VI. Введение в библиотеки базовых классов .NET** 685

### **Глава 19. Многопоточное, параллельное и асинхронное программирование** 686

Отношения между процессом, доменом приложения, контекстом и потоком	687
Проблема параллелизма	688
Роль синхронизации потоков	688
Краткий обзор делегатов .NET	689
Асинхронная природа делегатов	691
Методы <code>BeginInvoke()</code> и <code>EndInvoke()</code>	691
Интерфейс <code>System.IAsyncResult</code>	691
Асинхронный вызов метода	692
Синхронизация вызывающего потока	693
Роль делегата <code>AsyncCallback</code>	694
Роль класса <code>AsyncResult</code>	696
Передача и получение специальных данных состояния	697
Пространство имен <code>System.Threading</code>	698
Класс <code>System.Threading.Thread</code>	699
Получение статистики о текущем потоке выполнения	700
Свойство <code>Name</code>	700
Свойство <code>Priority</code>	701
Ручное создание вторичных потоков	702
Работа с делегатом <code>ThreadStart</code>	702
Работа с делегатом <code>ParametrizedThreadStart</code>	704
Класс <code>AutoResetEvent</code>	705
Потоки переднего плана и фоновые потоки	706
Проблемы параллелизма	707
Синхронизация с использованием ключевого слова <code>lock</code> языка C#	709
Синхронизация с использованием типа <code>System.Threading.Monitor</code>	711
Синхронизация с использованием типа <code>System.Threading.Interlocked</code>	712
Синхронизация с использованием атрибута <code>[Synchronization]</code>	713
Программирование с использованием обратных вызовов <code>Timer</code>	714
Использование автономного отбрасывания	715
Пул потоков CLR	716
Параллельное программирование с использованием TPL	717
Пространство имен <code>System.Threading.Tasks</code>	718
Роль класса <code>Parallel</code>	718
Обеспечение параллелизма данных с помощью класса <code>Parallel</code>	719
Доступ к элементам пользовательского интерфейса во вторичных потоках	722
Класс <code>Task</code>	723
Обработка запроса на отмену	723
Обеспечение параллелизма задач с помощью класса <code>Parallel</code>	725
Запросы <code>Parallel LINQ (PLINQ)</code>	728
Создание запроса <code>PLINQ</code>	729
Отмена запроса <code>PLINQ</code>	729

Асинхронные вызовы с помощью ключевого слова <code>async</code>	731
Знакомство с ключевыми словами <code>async</code> и <code>await</code> языка C#	731
Соглашения об именовании асинхронных методов	733
Асинхронные методы, возвращающие <code>void</code>	733
Асинхронные методы с множеством контекстов <code>await</code>	734
Вызов асинхронных методов из неасинхронных методов	734
Ожидание с помощью <code>await</code> в блоках <code>catch</code> и <code>finally</code>	734
Обобщенные возвращаемые типы в асинхронных методах (нововведение)	735
Локальные функции (нововведение)	735
Итоговые сведения о ключевых словах <code>async</code> и <code>await</code>	736
Резюме	737
<b>Глава 20. Файловый ввод-вывод и сериализация объектов</b>	<b>738</b>
Исследование пространства имен <code>System.IO</code>	738
Классы <code>Directory</code> ( <code>DirectoryInfo</code> ) и <code>File</code> ( <code>FileInfo</code> )	740
Абстрактный базовый класс <code>FileSystemInfo</code>	740
Работа с типом <code>DirectoryInfo</code>	741
Перечисление файлов с помощью типа <code>DirectoryInfo</code>	742
Создание подкаталогов с помощью типа <code>DirectoryInfo</code>	743
Работа с типом <code>Directory</code>	744
Работа с типом <code>DriveInfo</code>	745
Работа с классом <code>FileInfo</code>	746
Метод <code>FileInfo.Create()</code>	747
Метод <code>FileInfo.Open()</code>	747
Методы <code>FileInfo.OpenRead()</code> и <code>FileInfo.OpenWrite()</code>	749
Метод <code>FileInfo.OpenText()</code>	749
Методы <code>FileInfo.CreateText()</code> и <code>FileInfo.AppendText()</code>	749
Работа с типом <code>File</code>	750
Дополнительные члены <code>File</code>	751
Абстрактный класс <code>Stream</code>	752
Работа с классом <code>FileStream</code>	753
Работа с классами <code>StreamWriter</code> и <code>StreamReader</code>	754
Запись в текстовый файл	755
Чтение из текстового файла	756
Прямое создание объектов типа <code>StreamWriter/StreamReader</code>	756
Работа с классами <code>StringWriter</code> и <code>StringReader</code>	757
Работа с классами <code>BinaryWriter</code> и <code>BinaryReader</code>	758
Программное слежение за файлами	760
Понятие сериализации объектов	762
Роль графов объектов	764
Конфигурирование объектов для сериализации	765
Определение сериализируемых типов	765
Открытые поля, закрытые поля и открытые свойства	766
Выбор формatera сериализации	767
Интерфейсы <code>IFormatter</code> и <code>IRemotingFormatter</code>	767
Точность воспроизведения типов среди форматов	768
Сериализация объектов с использованием <code>BinaryFormatter</code>	769
Десериализация объектов с использованием <code>BinaryFormatter</code>	771
Сериализация объектов с использованием <code>SoapFormatter</code>	771
Сериализация объектов с использованием <code>XmlSerializer</code>	772
Управление генерацией данных XML	773

Сериализация коллекций объектов	775
Настройка процесса сериализации SOAP и двоичной сериализации	776
Углубленный взгляд на сериализацию объектов	776
Настройка сериализации с использованием ISerializable	777
Настройка сериализации с использованием атрибутов	780
Резюме	781
<b>Глава 21. Доступ к данным с помощью ADO.NET</b>	<b>782</b>
Высокоуровневое определение ADO.NET	783
Три грани ADO.NET	783
Поставщики данных ADO.NET	784
Поставщики данных ADO.NET производства Microsoft	785
Получение сторонних поставщиков данных ADO.NET	787
Дополнительные пространства имен ADO.NET	787
Типы из пространства имен System.Data	788
Роль интерфейса IDbConnection	789
Роль интерфейса IDbTransaction	789
Роль интерфейса IDbCommand	790
Роль интерфейсов IDbDataParameter и IDataParameter	790
Роль интерфейсов IDbDataAdapter и IDataAdapter	791
Роль интерфейсов IDataReader и IDataRecord	791
Абстрагирование поставщиков данных с использованием интерфейсов	792
Повышение гибкости с использованием конфигурационных файлов приложения	794
Создание базы данных AutoLot	796
Установка SQL Server 2016 и SQL Server Management Studio	796
Создание таблицы Inventory	796
Добавление тестовых записей в таблицу Inventory	799
Создание хранимой процедуры GetPetName ( )	799
Создание таблиц Customers и Orders	800
Создание отношений между таблицами	801
Модель фабрики поставщиков данных ADO.NET	803
Полный пример фабрики поставщиков данных	804
Потенциальный недостаток модели фабрики поставщиков данных	807
Элемент <connectionStrings>	808
Понятие подключенного уровня ADO.NET	809
Работа с объектами подключений	809
Работа с объектами ConnectionStringBuilder	811
Работа с объектами команд	812
Работа с объектами чтения данных	814
Получение множества результирующих наборов с использованием объекта чтения данных	815
Работа с запросами создания, обновления и удаления	816
Добавление конструкторов	817
Открытие и закрытие подключения	817
Создание модели автомобиля	818
Добавление методов выборки	818
Вставка новой записи об автомобиле	819
Создание строго типизированного метода InsertCar ( )	820
Добавление логики удаления	820
Добавление логики обновления	821
Работа с параметризованными объектами команд	821

Указание параметров с использованием типа DbParameter	822
Выполнение хранимой процедуры	823
Создание консольного клиентского приложения	825
Понятие транзакций базы данных	826
Основные члены объекта транзакции ADO.NET	827
Добавление таблицы CreditRisks в базу данных AutoLot	828
Добавление метода транзакции в InventoryDAL	828
Тестирование транзакции базы данных	830
Выполнение массового копирования с помощью ADO.NET	831
Исследование класса SqlBulkCopy	831
Создание специального класса чтения данных	831
Выполнение массового копирования	833
Тестирование массового копирования	834
Резюме	835
<b>Глава 22. Введение в Entity Framework 6</b>	<b>836</b>
Роль Entity Framework	837
Роль сущностей	838
Строительные блоки Entity Framework	839
Роль класса DbContext	839
Аннотации данных Entity Framework	842
Парадигма Code First из существующей базы данных	843
Генерация модели	843
Что мы получили?	846
Изменение стандартных отображений	849
Добавление возможностей в сгенерированные классы моделей	850
Использование классов моделей в коде	851
Вставка данных	851
Выборка записей	852
Роль навигационных свойств	855
Удаление данных	858
Обновление записи	860
Обработка изменений в базе данных	860
Создание уровня доступа к данным AutoLotDAL	861
Добавление классов моделей	862
Обновление класса, производного от DbContext	862
Инициализация базы данных	863
Тестирование сборки AutoLotDAL	865
Миграции Entity Framework	865
Создание начальной миграции	866
Обновление модели	867
Создание финальной миграции	869
Начальное заполнение базы данных	870
Добавление хранилищ для многократного использования кода	871
Добавление интерфейса IRepo	871
Добавление класса BaseRepo	871
Продолжение тестирования AutoLotDAL	874
Вывод инвентаризационных записей	875
Добавление инвентаризационных записей	875
Редактирование записей	875
Удаление записей	875

Параллелизм	876
Перехват	877
Интерфейс IDbCommandInterceptor	877
Добавление перехвата в AutoLotDAL	878
Регистрация перехватчика	879
Добавление перехватчика DatabaseLogger	879
События ObjectMaterialized и SavingChanges	880
Доступ к объектному контексту	880
Событие ObjectMaterialized	880
Событие SavingChanges	880
Отделение модели от уровня доступа к данным	882
Развертывание в SQL Server Express	882
Развертывание в SQL Server Express с использованием миграций	882
Создание сценария миграции	883
Резюме	883
<b>Глава 23. Введение в Windows Communication Foundation</b>	<b>884</b>
Выбор API-интерфейса распределенных вычислений	884
Роль WCF	885
Обзор функциональных возможностей WCF	886
Обзор архитектуры, ориентированной на службы	886
Итоговые сведения об инфраструктуре WCF	888
Исследование основных сборок WCF	888
Шаблоны проектов WCF в Visual Studio	888
Шаблон проекта WCF Service для веб-сайта	890
Базовая структура приложения WCF	891
Адрес, привязка и контракт в WCF	892
Понятие контрактов WCF	892
Понятие привязок WCF	893
Понятие адресов WCF	896
Построение службы WCF	897
Атрибут [ServiceContract]	898
Атрибут [OperationContract]	899
Типы служб как контракты операций	900
Хостинг службы WCF	900
Установка ABC внутри файла App.config	901
Кодирование с использованием типа ServiceHost	902
Указание базовых адресов	903
Подробный анализ типа ServiceHost	904
Подробный анализ элемента <system.serviceModel>	906
Включение обмена метаданными	907
Построение клиентского приложения WCF	909
Генерация кода прокси с использованием svcutil.exe	909
Генерация кода прокси в Visual Studio	910
Конфигурирование привязки на основе TCP	912
Упрощение конфигурационных настроек	914
Использование стандартных конечных точек	914
Открытие доступа к одиночной службе WCF, использующей множество привязок	915
Изменение параметров привязки WCF	916
Использование конфигурации стандартного поведения MEX	918
Обновление клиентского прокси и выбор привязки	919

Использование шаблона проекта WCF Service Library	920
Построение простой математической службы	920
Тестирование службы WCF с помощью WcfTestClient.exe	921
Изменение конфигурационных файлов с использованием SvcConfigEditor.exe	922
Хостинг службы WCF внутри Windows-службы	923
Указание ABC в коде	924
Включение MEX	925
Создание программы установки для Windows-службы	926
Установка Windows-службы	927
Асинхронный вызов службы из клиента	928
Проектирование контрактов данных WCF	930
Использование веб-ориентированного шаблона проекта WCF Service	931
Обновление пакетов NuGet и установка AutoMapper и EntityFramework	933
Реализация контракта службы	934
Роль файла *.svc	934
Исследование файла Web.config	935
Тестирование службы	936
Резюме	936
<b>Часть VII. Windows Presentation Foundation</b>	937
<b>Глава 24. Введение в Windows Presentation Foundation и XAML</b>	938
Мотивация, лежащая в основе WPF	938
Унификация несходных API-интерфейсов	939
Обеспечение разделения обязанностей через XAML	940
Обеспечение оптимизированной модели визуализации	940
Упрощение программирования сложных пользовательских интерфейсов	941
Исследование сборок WPF	942
Роль класса Application	943
Роль класса Window	945
Роль класса System.Windows.Controls.ContentControl	946
Роль класса System.Windows.Controls.Control	947
Роль класса System.Windows.FrameworkElement	947
Роль класса System.Windows.UIElement	948
Роль класса System.Windows.Media.Visual	949
Роль класса System.Windows.DependencyObject	949
Роль класса System.Windows.Threading.DispatcherObject	949
Синтаксис XAML для WPF	949
Введение в Кахaml	949
Пространства имен XML и “ключевые слова” XAML	950
Управление видимостью классов и переменных-членов	953
Элементы XAML, атрибуты XAML и преобразователи типов	954
Понятие синтаксиса “свойство-элемент” в XAML	955
Понятие присоединяемых свойств XAML	956
Понятие расширений разметки XAML	957
Построение приложений WPF с использованием Visual Studio	958
Шаблоны проектов WPF	959
Панель инструментов и визуальный конструктор/редактор XAML	959
Установка свойств с использованием окна Properties	960
Обработка событий с использованием окна Properties	962
Обработка событий в редакторе XAML	962
Окно Document Outline	963

Включение и отключение отладчика XAML	964
Исследование файла App.xaml	965
Отображение разметки XAML окна на код C#	966
Роль BAML	967
Разгадывание загадки Main()	968
Взаимодействие с данными уровня приложения	968
Обработка закрытия объекта Window	970
Перехват событий мыши	971
Перехват событий клавиатуры	972
Изучение документации WPF	972
Резюме	973
<b>Глава 25. Элементы управления, компоновки, события и привязка данных в WPF</b>	<b>974</b>
Обзор основных элементов управления WPF	974
Элементы управления Ink API	974
Элементы управления документов WPF	975
Общие диалоговые окна WPF	976
Подробные сведения находятся в документации	976
Краткий обзор визуального конструктора WPF в Visual Studio	976
Работа с элементами управления WPF в Visual Studio	977
Работа с окном Document Outline	978
Управление компоновкой содержимого с использованием панелей	978
Позиционирование содержимого внутри панелей Canvas	979
Позиционирование содержимого внутри панелей WrapPanel	981
Позиционирование содержимого внутри панелей StackPanel	983
Позиционирование содержимого внутри панелей Grid	984
Позиционирование содержимого внутри панелей DockPanel	986
Включение прокрутки в типах панелей	988
Конфигурирование панелей с использованием визуальных конструкторов Visual Studio	988
Построение окна с использованием вложенных панелей	991
Построение системы меню	992
Построение панели инструментов	994
Построение строки состояния	994
Завершение проектирования пользовательского интерфейса	995
Реализация обработчиков событий MouseEnter/MouseLeave	996
Реализация логики проверки правописания	996
Понятие команд WPF	997
Внутренние объекты команд	997
Подключение команд к свойству Command	998
Подключение команд к произвольным действиям	999
Работа с командами Open и Save	1000
Понятие маршрутизируемых событий	1002
Роль пузырьковых маршрутизируемых событий	1003
Продолжение или прекращение пузырькового распространения	1004
Роль туннельных маршрутизируемых событий	1004
Более глубокий взгляд на API-интерфейсы и элементы управления WPF	1006
Работа с элементом управления TabControl	1006
Построение вкладки Ink API	1007
Проектирование панели инструментов	1008
Элемент управления RadioButton	1008

Добавление кнопок сохранения, загрузки и удаления	1009
Добавление элемента управления InkCanvas	1009
Предварительный просмотр окна	1009
Обработка событий для вкладки Ink API	1010
Добавление элементов управления в панель инструментов	1010
Элемент управления InkCanvas	1010
Элемент управления ComboBox	1012
Сохранение, загрузка и очистка данных InkCanvas	1014
Введение в модель привязки данных WPF	1015
Построение вкладки Data Binding	1016
Установка привязки данных	1016
Свойство DataContext	1016
Форматирование привязанных данных	1018
Преобразование данных с использованием интерфейса IValueConverter	1018
Установка привязок данных в коде	1019
Построение вкладки DataGrid	1020
Роль свойств зависимости	1021
Исследование существующего свойства зависимости	1023
Важные замечания относительно оболочек свойств CLR	1025
Построение специального свойства зависимости	1026
Добавление процедуры проверки достоверности данных	1028
Реагирование на изменение свойства	1029
Резюме	1030
<b>Глава 26. Службы визуализации графики WPF</b>	<b>1031</b>
Службы графической визуализации WPF	1031
Варианты графической визуализации WPF	1032
Визуализация графических данных с использованием фигур	1033
Добавление прямоугольников, эллипсов и линий на поверхность Canvas	1035
Удаление прямоугольников, эллипсов и линий с поверхности Canvas	1037
Работа с элементами Polyline и Polygon	1038
Работа с элементом Path	1039
Кисти и перья WPF	1042
Конфигурирование кистей с использованием Visual Studio	1042
Конфигурирование кистей в коде	1045
Конфигурирование перьев	1046
Применение графических трансформаций	1046
Первый взгляд на трансформации	1047
Трансформация данных Canvas	1048
Работа с редактором трансформаций Visual Studio	1050
Построение начальной компоновки	1050
Применение трансформаций на этапе проектирования	1052
Трансформация холста в коде	1052
Визуализация графических данных с использованием рисунков и геометрических объектов	1053
Построение кисти DrawingBrush с использованием геометрических объектов	1054
Рисование с помощью DrawingBrush	1055
Включение типов Drawing в DrawingImage	1056
Работа с векторными изображениями	1056
Преобразование файла с векторной графикой в файл XAML	1057
Импортирование графических данных в проект WPF	1058

Взаимодействие с изображением	1058
Визуализация графических данных с использованием визуального уровня	1059
Базовый класс <code>Visual</code> и производные дочерние классы	1059
Первый взгляд на класс <code>DrawingVisual</code>	1060
Визуализация графических данных в специальном диспетчере компоновки	1062
Реагирование на операции проверки попадания	1064
Резюме	1065
<b>Глава 27. Ресурсы, анимация, стили и шаблоны WPF</b>	1066
Система ресурсов WPF	1066
Работа с двоичными ресурсами	1067
Программная загрузка изображения	1068
Работа с объектными (логическими) ресурсами	1070
Роль свойства <code>Resources</code>	1071
Определение ресурсов уровня окна	1071
Расширение разметки <code>{StaticResource}</code>	1073
Расширение разметки <code>{DynamicResource}</code>	1074
Ресурсы уровня приложения	1074
Определение объединенных словарей ресурсов	1075
Определение сборки, включающей только ресурсы	1076
Службы анимации WPF	1078
Роль классов анимации	1078
Свойства <code>To</code> , <code>From</code> и <code>By</code>	1079
Роль базового класса <code>Timeline</code>	1080
Реализация анимации в коде C#	1080
Управление темпом анимации	1082
Запуск в обратном порядке и циклическое выполнение анимации	1082
Реализация анимации в разметке XAML	1083
Роль раскадровок	1084
Роль триггеров событий	1084
Анимация с использованием дискретных ключевых кадров	1085
Роль стилей WPF	1086
Определение и применение стиля	1087
Переопределение настроек стиля	1087
Влияние атрибута <code>TargetType</code> на стили	1088
Создание подклассов существующих стилей	1089
Определение стилей с триггерами	1090
Определение стилей с множеством триггеров	1090
Анимированные стили	1091
Применение стилей в коде	1091
Логические деревья, визуальные деревья и стандартные шаблоны	1093
Программное инспектирование логического дерева	1093
Программное инспектирование визуального дерева	1095
Программное инспектирование стандартного шаблона элемента управления	1096
Построение шаблона элемента управления с помощью инфраструктуры триггеров	1099
Шаблоны как ресурсы	1100
Встраивание визуальных подсказок использованием триггеров	1101
Роль расширения разметки <code>{TemplateBinding}</code>	1102
Роль класса <code>ContentPresenter</code>	1103
Встраивание шаблонов в стили	1103
Резюме	1104

<b>Глава 28. Уведомления, проверка достоверности, команды и MVVM</b>	1105
Введение в паттерн MVVM	1105
Модель	1106
Представление	1106
Модель представления	1106
Анемичные модели или анемичные модели представлений	1106
Система уведомлений привязки WPF	1107
Наблюдаемые модели и коллекции	1107
Добавление привязок и данных	1109
Изменение данных об автомобиле в коде	1110
Наблюдаемые модели	1111
Наблюдаемые коллекции	1112
Проверка достоверности WPF	1116
Модификация примера для демонстрации проверки достоверности	1117
Класс Validation	1117
Варианты проверки достоверности	1117
Использование аннотаций данных в WPF	1127
Настройка свойства ErrorMessage	1129
Итоговые сведения о проверке достоверности	1132
Создание специальных команд	1132
Реализация интерфейса ICommand	1132
Добавление команды ChangeColorCommand	1132
Присоединение команды к CommandManager	1133
Изменение файла MainWindow.xaml.cs	1133
Изменение файла MainWindow.xaml	1134
Тестирование приложения	1134
Объекты RelayCommand	1136
Итоговые сведения о командах	1138
Перенос кода и данных в модель представления	1139
Перенос кода MainWindow.xaml.cs	1139
Обновление кода и разметки MainWindow	1140
Обновление разметки элементов управления	1140
Итоговые сведения о моделях представлений	1141
Изменение сборки AutoLotDAL для MVVM	1141
Обновление класса EntityBase	1141
Обновление частичного класса Inventory	1142
Добавление PropertyChanged.Fody в проект Models	1142
Добавление Entity Framework и строки подключения в проект приложения WPF	1142
Обновление разметки XAML для MainWindow	1142
Обновление модели представления	1142
Обновление команды AddCarCommand	1143
Использование события ObjectMaterialized с инфраструктурой Entity Framework	1143
Резюме	1143
<b>Часть VIII. ASP.NET</b>	1145
<b>Глава 29. Введение в ASP.NET MVC</b>	1146
Введение в паттерн MVC	1146
Модель	1146
Представление	1147
Контроллер	1147
Почему появилась инфраструктура MVC?	1147

Появление ASP.NET MVC	1148
Шаблон приложения ASP.NET MVC	1148
Мастер создания проекта	1148
Окно обзора проекта	1151
Файлы в корневой папке проекта	1151
Файл Global.asax.cs	1151
Папка Models	1152
Папка Controllers	1152
Папка Views	1153
Папки ASP.NET	1153
Папка App_Start	1154
Папка Content	1156
Папка Fonts	1156
Папка Scripts	1157
Обновление пакетов NuGet проекта	1157
Обновление настроек проекта	1157
Маршрутизация	1158
Шаблоны URL	1158
Создание маршрутов для страниц Contact и About	1159
Перенаправление пользователей с применением маршрутизации	1160
Добавление библиотеки AutoLotDAL	1161
Контроллеры и действия	1162
Результаты действий	1162
Добавление контроллера Inventory	1162
Исследование и обновление InventoryController	1165
Механизм представлений Razor	1172
Синтаксис Razor	1172
Операторы, блоки кода и разметка	1173
Встроенные вспомогательные методы HTML	1174
Специальные вспомогательные методы HTML	1175
Функции Razor	1176
Делегаты Razor	1176
Представления MVC	1177
Компоновки	1177
Частичные представления	1178
Отправка данных представлениям	1179
Аннотации данных Display	1180
Специальные файлы метаданных	1180
Шаблоны Razor	1181
Создание специального шаблона для отображения	1181
Создание специального шаблона для редактирования	1183
Работа с формами	1184
Вспомогательный метод HTML по имени BeginForm()	1184
Вспомогательный метод HTML по имени AntiForgeryToken()	1185
Обновление представления Delete	1185
Проверка достоверности	1185
Отображение ошибок	1185
Отключение проверки достоверности клиентской стороны	1186
Проверка достоверности клиентской стороны	1186
Итоговые сведения об инфраструктуре ASP.NET MVC	1187
Резюме	1188

<b>Глава 30. Введение в ASP.NET Web API</b>	1189
Инфраструктура ASP.NET Web API	1189
Создание проекта Web API	1190
Создание класса InventoryController	1191
Маршрутизация	1192
Формат JSON	1194
Результаты действий Web API	1195
Проблема сериализации, связанная с использованием Entity Framework	1196
Получение складских данных	1197
Добавление метода Dispose()	1197
Использование Fiddler	1197
Обновление складской записи (HttpPut)	1199
Добавление складских записей (HttpPost)	1200
Удаление складских записей (HttpDelete)	1201
Итоговые сведения о Web API	1202
Обновление проекта CarLotMVC для использования CarLotWebAPI	1202
Добавление приложения CarLotMVC	1202
Обновление класса InventoryController в проекте MVC	1203
Обновление метода действия Index()	1203
Обновление метода действия Details()	1204
Обновление метода действия Create()	1204
Обновление метода действия Edit()	1205
Обновление метода действия Delete()	1206
Резюме	1207
<b>Часть IX. .NET Core</b>	1209
<b>Глава 31. Философия .NET Core</b>	1210
От проекта Project K до .NET Core	1210
Будущее полной платформы .NET Framework	1211
Цели .NET Core	1211
Межплатформенная поддержка	1212
Производительность	1213
Переносимые библиотеки классов, соответствующие стандарту .NET Standard	1213
Переносимые или автономные модели развертывания	1214
Полная поддержка командной строки	1214
Открытый код	1214
Возможность взаимодействия с .NET Framework	1214
Состав .NET Core	1215
Исполняющая среда .NET Core (CoreCLR)	1215
Библиотеки инфраструктуры (CoreFX)	1215
Инструменты SDK и хост приложений dotnet	1215
Компиляторы языков	1216
Жизненный цикл поддержки .NET Core	1216
Установка .NET Core 2.0	1217
Сравнение с полной платформой .NET Framework	1218
Сниженное количество поддерживаемых моделей приложений	1218
Меньшее число реализованных API-интерфейсов и подсистем	1218
Резюме	1219

<b>Глава 32. Введение в Entity Framework Core</b>	1220
Сравнение наборов функциональных возможностей	1220
Средства, которые не дублировались	1221
Изменения по сравнению с EF 6	1221
Новые функциональные средства в EF Core	1221
Сценарии использования	1222
Создание AutoLotCoreDAL_Core2	1223
Создание проектов и решения	1223
Добавление пакетов NuGet	1223
Добавление классов моделей	1224
Создание класса AutoLotContext	1227
Создание базы данных с помощью миграций	1232
Инициализация базы данных начальной информацией	1233
Добавление хранилищ для многократного использования кода	1236
Добавление интерфейса IRepo	1237
Добавление класса BaseRepo	1237
Создание InventoryRepo	1240
Тестирование библиотеки AutoLotDAL_Core2	1242
Резюме	1243
<b>Глава 33. Введение в веб-приложения ASP.NET Core</b>	1244
Шаблон ASP.NET Core Web Application	1244
Мастер создания проекта	1244
Организация проекта ASP.NET Core	1246
Добавление библиотеки доступа к данным	1246
Обновление пакетов NuGet	1247
Запуск приложений ASP.NET Core	1247
Развертывание приложений ASP.NET Core	1249
Нововведения ASP.NET Core	1249
Унифицированный подход к построению веб-приложений и служб	1250
Встроенное внедрение зависимостей	1250
Система конфигурации, основанная на среде и готовая к взаимодействию с облачными технологиями	1251
Способность выполняться под управлением .NET Core или полной платформы .NET Framework	1253
Легковесный и модульный конвейер запросов HTTP	1253
Интеграция современных инфраструктур клиентской стороны	1253
Вспомогательные функции дескрипторов	1253
Компоненты представлений	1263
Изменения механизма представлений Razor	1264
Построение AutoLotMVC_Core2	1264
Файл Program.cs	1264
Файл Startup.cs	1266
Управление пакетами с помощью Bower	1269
Объединение в пакеты и минификация	1270
Содержимое клиентской стороны (папка wwwroot)	1274
Папки для моделей, контроллеров и представлений	1274
Контроллеры и действия	1275
Базовый класс Controller	1275
Действия	1275
Типы ViewResult	1277

Добавление контроллера Inventory	1277
Обновление класса InventoryController	1279
Итоговые сведения о контроллерах и действиях	1283
Представления	1283
Обновление файла импортирования представлений	1283
Представление компоновки	1284
Частичное представление со сценариями проверки достоверности	1285
Шаблон для отображения складских данных	1285
Шаблон для редактирования складских данных	1286
Представление Index	1287
Представление Details	1288
Представление Create	1288
Представление Edit	1289
Представление Delete	1290
Итоговые сведения о представлениях	1290
Компоненты представлений	1290
Написание кода серверной стороны	1291
Написание кода клиентской стороны	1292
Обращение к компонентам представлений	1293
Обращение к компонентам представлений как к специальным вспомогательным функциям дескрипторов	1293
Добавление компонента представления в AutoLotMVC_Core2	1293
Резюме	1294
<b>Глава 34. Введение в приложения служб ASP.NET Core</b>	1295
Шаблон ASP.NET Core Web API	1295
Мастер создания проекта	1295
Организация проекта приложения службы ASP.NET Core	1297
Добавление библиотеки доступа к данным	1297
Обновление и добавление пакетов NuGet	1297
Запуск и развертывание приложений служб	1297
Изменения в приложениях служб ASP.NET Core	1297
Формат возвращаемых данных JSON	1297
Явная маршрутизация для методов HTTP	1298
Управление привязкой моделей в ASP.NET Core	1298
Построение AutoLotAPI_Core2	1299
Добавление строки подключения	1299
Обновление файла Program.cs для инициализации данных	1299
Обновление файла Startup.cs	1300
Добавление обработки исключений на уровне приложений	1301
Добавление контроллера Inventory	1303
Итоговые сведения о приложениях служб ASP.NET Core	1308
Обновление AutoLotMVC_Core2 для использования AutoLotAPI_Core2	1309
Копирование и добавление приложения AutoLotMVC_Core2	1309
Удаление AutoLotDAL_Core2 из AutoLotMVC_Core2	1309
Создание нового контроллера Inventory	1310
Итоговые сведения о миграции AutoLotMVC_Core2	1315
Резюме	1315
<b>Предметный указатель</b>	1316