



# Содержание

Об авторах	xi
<b>Введение</b>	<b>xiii</b>
Основные темы книги	xiii
Предполагаемый круг читателей	xv
Что необходимо для работы с книгой	xv
Соглашения о стилях	xv
Как скопировать код примеров для этой книги	xvi
<b>ГЛАВА 1. Сетевые понятия и протоколы</b>	<b>1</b>
<b>Физическая сеть</b>	<b>2</b>
Линии глобальной сети	2
Протокол Ethernet	3
Физические компоненты	5
<b>Многоуровневая модель OSI</b>	<b>10</b>
Уровень 1: физический уровень	12
Уровень 2: канальный уровень	12
Уровень 3: сетевой уровень	13
Уровень 4: транспортный уровень	13
Уровень 5: сеансовый уровень	14
Уровень 6: представительский уровень	14
Уровень 7: прикладной уровень	14
<b>Сетевые протоколы</b>	<b>14</b>
Базовые протоколы	15
Протоколы Интернета	25
Протоколы электронной почты	29
Другие прикладные протоколы	30

<b>Сокеты</b>	<b>31</b>
<b>Имена доменов</b>	<b>31</b>
Служба whois	33
Серверы доменных имен	33
<b>Интернет</b>	<b>34</b>
Интрасети и экстрасети	34
Брандмауэры	34
Web-прокси	35
XML Web-сервисы	36
<b>Передача сообщений</b>	<b>38</b>
<b>Другие способы доступа к сетевым объектам</b>	<b>38</b>
<b>Организации и стандарты Интернета</b>	<b>39</b>
<b>Итоги</b>	<b>40</b>
<b>ГЛАВА 2. Потоки в .NET</b>	<b>41</b>
<b>Потоки в .NET</b>	<b>41</b>
Синхронный и асинхронный ввод/вывод	42
Класс Stream	42
Класс FileStream	46
Чтение и запись в классе FileStream	49
Класс BufferedStream	53
Класс MemoryStream	53
Класс NetworkStream	54
Класс CryptoStream	57
<b>Обработка потоков</b>	<b>59</b>
Кодирование строковых данных	60
Двоичные файлы	62
TextReader	65
TextWriter	67
<b>Сериализация</b>	<b>69</b>
Сериализация в формат XML	70
Сериализация с помощью объектов форматирования	72
<b>Итоги</b>	<b>76</b>
<b>ГЛАВА 3. Сетевое программирование в .NET</b>	<b>77</b>
<b>Классы пространства System.Net — обзор</b>	<b>77</b>
Поиск имен	78
IP-адреса	78
Аутентификация и авторизация	78
Запросы и ответы	78
Управление соединениями	79
Записи cookies	80
Прокси-сервер	80
Сокеты	80
<b>Работа с URI</b>	<b>81</b>
Класс Uri	82
Свойства класса Uri	83
Изменение URI с помощью класса UriBuilder	85
Абсолютные и относительные URI	85

<b>IP-адреса</b>	<b>86</b>
Предопределенные адреса	87
Порядок байтов, используемый в хосте и сети	87
<b>Класс Dns</b>	<b>88</b>
Разрешение имени в IP-адрес	88
Как разрешается IP-адрес?	89
Асинхронное разрешение IP-адреса	91
<b>Запросы и ответы</b>	<b>92</b>
WebRequest и WebResponse	93
Подключаемые протоколы	95
FileWebRequest и FileWebResponse	95
Формирование пула соединений	98
<b>Использование Web-прокси</b>	<b>98</b>
Класс WebProxy	99
Web-прокси по умолчанию	100
Изменение WebProxy для конкретных запросов	100
<b>Аутентификация</b>	<b>101</b>
<b>Разрешения</b>	<b>101</b>
Использование атрибутов разрешения	102
Конфигурирование разрешений	105
<b>Итоги</b>	<b>107</b>
<hr/> <b>ГЛАВА 4. Работа с сокетами</b>	<hr/> <b>109</b>
<b>Сокеты</b>	<b>109</b>
Типы сокетов	110
<b>Работа с сокетами в .NET</b>	<b>112</b>
Класс System.Net.Sockets.Socket	113
Создание приложения на потоковом соquete TCP	114
Управление исключениями в System.Net.Sockets	122
Опции сокетов	125
<b>Асинхронное программирование</b>	<b>127</b>
Асинхронное приложение-клиент	128
Асинхронное приложение-сервер	133
<b>Разрешения сокетов</b>	<b>137</b>
<b>Итоги</b>	<b>143</b>
<hr/> <b>ГЛАВА 5. TCP</b>	<hr/> <b>145</b>
<b>Обзор TCP</b>	<b>145</b>
Инкапсуляция	145
Терминология TCP	146
Заголовки TCP	146
Соединения TCP	147
Операции TCP	148
<b>Введение в TCP на платформе .NET</b>	<b>149</b>
Класс TcpClient	150
Построение реального приложения на сокетах	155
Реализация класса FtpWebRequest	161
Класс TcpListener	175

<b>.NET Remoting</b>	<b>182</b>
Домены приложений	183
Как работает Remoting	183
<b>Итоги</b>	<b>191</b>
<b>ГЛАВА 6. UDP</b>	<b>192</b>
<b>Обзор протокола UDP</b>	<b>192</b>
<b>UDP в .NET</b>	<b>199</b>
Класс <code>UdpClient</code>	200
Приложение интерактивного форума, использующее UDP	210
<b>Приложение передачи файла</b>	<b>214</b>
Файловый сервер	214
Приемник файла	217
Широковещательная передача	219
<b>Высокоуровневые протоколы, базирующиеся на UDP</b>	<b>220</b>
<b>Итоги</b>	<b>221</b>
<b>ГЛАВА 7. Сокеты групповой рассылки</b>	<b>222</b>
<b>Однонаправленные, широковещательные и групповые передачи</b>	<b>223</b>
<b>Модели приложений с групповой рассылкой</b>	<b>224</b>
<b>Архитектура сокетов групповой рассылки</b>	<b>225</b>
Протокол IGMP	226
Групповые адреса	227
Масштабируемость	232
Надежность	232
Безопасность	233
<b>Использование сокетов групповой рассылки в .NET</b>	<b>233</b>
Отправитель	233
Получатель	234
<b>Создание приложения интерактивного форума</b>	<b>235</b>
Пользовательский интерфейс	235
Параметры конфигурирования	237
Присоединение к группе, получающей рассылку	238
Получение сообщений, адресованных группе	239
Отправка групповых сообщений	240
Прекращение членства в группе	240
Запуск приложения интерактивного форума	241
<b>Приложение демонстрации изображений</b>	<b>241</b>
Реализация демонстрации изображений	242
Создание протокола для изображений	242
Сервер демонстрации изображений	247
Клиент приложения демонстрации изображений	257
<b>Итоги</b>	<b>263</b>
<b>ГЛАВА 8. HTTP</b>	<b>264</b>
<b>Обзор протокола HTTP</b>	<b>264</b>
HTTP-заголовки	266
HTTP-запросы	268
HTTP-ответы	270

<b>HTTP в .NET</b>	<b>271</b>
HttpRequest и HttpResponse	272
Приложение перевода валют	274
Отсылка данных на сервер	276
Передача данных порциями в HTTP	277
Поддержка активного соединения HTTP	279
Управление соединением HTTP	279
Класс WebClient	281
Аутентификация	286
Поддержка прокси-сервера	288
Чтение и запись cookie	289
<b>HTTP-сервер с поддержкой ASP.NET</b>	<b>293</b>
Конфигурационные файлы сервера	293
Кодирование сервера	294
<b>HTTP и .NET Remoting</b>	<b>304</b>
Построение простого приложения для среды Remoting	304
<b>Итоги</b>	<b>307</b>
<hr/> <b>ГЛАВА 9. Протоколы электронной почты</b>	<hr/> <b>308</b>
<b>Об электронной почте коротко</b>	<b>308</b>
<b>Как работает электронная почта</b>	<b>309</b>
<b>Протоколы электронной почты</b>	<b>310</b>
SMTP	310
Типичное сообщение электронной почты	314
Просмотр заголовков в Outlook	317
Как обстоит дело с MIME?	317
Получение электронной почты в системе клиент-сервер	320
<b>.NET и электронная почта</b>	<b>324</b>
SMTP	324
Приложение почтового SMTP-клиента	328
POP3	329
NNTP	336
<b>Итоги</b>	<b>344</b>
<hr/> <b>ГЛАВА 10. Криптография в .NET</b>	<hr/> <b>345</b>
<b>История криптографии</b>	<b>345</b>
<b>Что такое криптография</b>	<b>346</b>
<b>Зачем нужна криптография</b>	<b>347</b>
Концепции криптографии	348
<b>Криптографические алгоритмы</b>	<b>349</b>
Симметричные алгоритмы	349
Асимметричные алгоритмы	351
Алгоритмы дайджеста сообщения	352
Цифровые подписи	353
<b>Криптографическая терминология</b>	<b>354</b>
Блочные и поточные шифры	354
Заполнение	354
Режимы	355

---

<b>Пространство имен System.Security.Cryptography</b>	<b>356</b>
Иерархия криптографии классов	357
<b>Хеширование в .NET</b>	<b>358</b>
Класс HashAlgorithm	358
<b>Симметрические преобразования в .NET</b>	<b>364</b>
Класс SymmetricAlgorithm	364
Использование других симметричных алгоритмов	368
<b>Асимметричные преобразования в .NET</b>	<b>369</b>
Класс AsymmetricAlgorithm	369
Использование алгоритма RSA	370
Загрузка открытых и закрытых ключей	373
<b>Чтение сертификата X509</b>	<b>375</b>
<b>Криптография и сетевое программирование</b>	<b>377</b>
<b>Итоги</b>	<b>382</b>
<b>ГЛАВА 11. Протоколы аутентификации</b>	<b>383</b>
<hr/>	
<b>Протоколы аутентификации</b>	<b>383</b>
NTLM	384
Kerberos	388
Безопасность в .NET и Windows	393
Интерфейс System.Net.IAuthenticationModule	398
<b>Итоги</b>	<b>400</b>

## Об авторах

### Эндрю Кровчик

Эндрю Кровчик — консультант по разработке программного обеспечения, в настоящее время с головой ушел в разработку платформы .NET. Закоренелый фанат Web-сервисов, Эндрю тратит значительное время на написание и рецензирование книг и статей, посвященных платформе .NET, для издательства Wrox Press. Кроме того, он преподает в Колледже Элмхерста (Элмхерст, штат Иллинойс), читает лекции на вечерних курсах.

Свободное время (когда оно вдруг появляется) Эндрю любит проводить с женой Элеонор и собакой Луи, повозиться со своим “мустангом” выпуска 1967 г.

С Эндрю можно связаться через его почтовый ящик [Krowczyk@i-netway.com](mailto:Krowczyk@i-netway.com).

### Винод Кумар

Винод Кумар — автор, разработчик и технический рецензент в одном лице. Он специализируется на Web- и мобильных технологиях, использующих решения Microsoft. В настоящее время работает в индийском городе Ченнай в компании Emerald Software Ltd. Винод — ведущий автор готовящейся к изданию книги “Mobile Application development with .NET”, написал много технических статей для таких сайтов, как [ASPToday.com](http://ASPToday.com) и [CSharpToday.com](http://CSharpToday.com). Он также ведет общественный сайт <http://www.dotnetforce.com> — первый индийский сайт, предоставляющий ресурсы по платформе Mobile.NET. В свободное время любит не спеша погулять с друзьями по берегу моря. С Винодом можно контактировать по адресу:

[vinod@dotnetforce.com](mailto:vinod@dotnetforce.com).

*Эту книгу я хотел бы посвятить Шириди Сатъя Сай Баба (Shiridi Sathya Sai Baba), поскольку он всегда благословлял меня во всех моих начинаниях.*

*Я благодарен Wrox Press за предоставленную мне возможность участвовать в создании этой книги, особенно — Шарлотте и Джулиану за их помощь. Огромное спасибо Шарлотте за постоянную поддержку и тепливое отношение ко мне с самого первого дня сотрудничества с ASPToday.com.*

### Номан Лагари

Номан Лагари — руководитель группы программистов в компании Creative Chaos (pvt) Limited в Карачи, Пакистан. У него большой опыт программирования на C/C++, в разработке таких современных проектов, как реализация торговой системы реального времени, соединенной с Electronic Crossing Networks (ECN) и использующей протокол Financial Information eXchange (FIX) для Wall Street Brokerage. Когда Номан не занят проектированием системной архитектуры, он любит писать научные доклады и статьи, посвященные новым направлениям, особенно использующим платформы Microsoft. Номан жадно читает книги, а поддерживать свою физическую форму предпочитает, играя в крикет.

### Аджит Мунгале

В последние три года Аджит Мунгале является старшим разработчиком программного обеспечения в IBM GSI. За те шесть лет, которые он отдал программированию, Аджит приобрел опыт работы с самыми разнообразными технологиями. Он начинал с программирования интегральных схем CPLD и FPGA и разработки драйверов устройств, работал почти со всеми языками и технологиями Microsoft и с продуктами IBM. Он специализируется в COM/DCOM/MTS с использованием

ATL/VB и является экспертом в VB, ASP, C++, XML, IBM MQ Series и .NET Framework. Совсем недавно Аджит подал заявку на выдачу патента по шифрованию и Web-безопасности.

Аджит любит природу, обладает коллекцией кактусов, карликовых деревьев и других растений. С ним можно пообщаться через почтовый ящик

[ajit\\_mungale@hotmail.com](mailto:ajit_mungale@hotmail.com).

*Я хотел бы посвятить эту книгу моим родителям, которые были для меня источником вдохновения. Особо хочу поблагодарить семью Назар (Nazar) за поддержку, а Джулиана – за постоянную помощь в работе над проектом.*

## Кристиан Нагел

Кристиан Нагел – преподаватель и консультант в компании Global Knowledge – самой крупной независимой организации, предоставляющей профессиональную подготовку в области информационных технологий. Кристиан начинал свою работу в области вычислительной техники с платформ PDP 11 и VAX/VMS, освоив самые разные языки и технологии. С платформой .NET и языком C# он работает с июля 2000 г., когда они впервые были официально объявлены. Обладая глубокими знаниями технологий Microsoft – он сертифицирован компанией Microsoft как дипломированный преподаватель (Microsoft Certified Trainer (MCT)), разработчик решений (Solution Developer (MCSD)) и системный инженер (System Engineer (MCSE)), – Кристиан также обучает других программированию и архитектуре распределенных решений. В ИТ-бизнесе у него и другие роли: основал группу пользователей платформы .NET в Австрии, является региональным директором MSDN, выступает с докладами на международных конференциях и руководит европейским отделением INTEA (Международная ассоциация групп пользователей платформы .NET). Web-сайт Кристиана вы найдете по адресу: <http://christian.nagel.net>.

*Хочу поблагодарить Эйлин Крейн, Стэси Джард и Эрика Ивинга из Microsoft за поддержку, а Кристиана Сейдлера – за сотрудничество в Global Knowledge. Особая благодарность – моей жене Элизабет за любовь и поддержку.*

## Тим Паркер

Уже 25 лет Тим Паркер – программист, писатель и преподаватель. Он написал более 60 книг и 3,5 тыс. журнальных статей. Тим работает в Web-среде с самого ее возникновения и спроектировал сотни Web-сайтов. В свободное время он занимается подводным плаванием, пилотирует самолет и управляет капризной сетью из 30 компьютеров, установленной в его доме в Оттаве, Канада.

*Маргарет Френсис, доброму и надежному другу – с благодарностью.*

## Шриниваса Шивакумар

Шриниваса Шивакумар – консультант по программному обеспечению, разработчик и писатель. Он специализируется в Web-технологиях и мобильных средствах, использующих решения Microsoft. В настоящее время работает в Чикаго в компании TransTech, LLC. Шриниваса участвовал в написании книг Professional ASP.NET Web Services, Professional ASP.NET Web Services with VB.NET, ASP.NET Mobile Controls – Tutorial Guide, Early Adopter .NET Compact Framework, Beginning ASP.NET 1.0 with VB.NET, Visual Basic .NET Threading Handbook, Beginning ASP.NET 1.0 with C#.NET, Professional ASP.NET Security и технических статей для ASPToday.com, CSharpToday.com, .NET Developer и т. д. В свободное время любит смотреть тамильские фильмы и слушать тамильские записи (особенно пение Бала-субраманияма (S. P. Balasubramaniyam)).



---



# *Введение*

**С**етевое программирование является одной из центральных задач при разработке уровня бизнес-приложений — потребность в эффективном и безопасном взаимодействии разных компьютеров, находящихся в одном здании или разбросанных по всему миру, остается основной для успеха многих систем. Со средой .NET Framework приходит новый набор классов для решения задач сетевого обмена.

Прочитав книгу, вы станете уверенным сетевым программистом на платформе .NET и будете понимать базовые протоколы. В настоящее время набор протоколов, поддерживаемых классами .NET, ограничен для транспортного уровня протоколами TCP и UDP, а на прикладном уровне — протоколами HTTP и SMTP. В этой книге мы не только полностью освещаем соответствующие классы, но и рассматриваем примеры реализации в .NET протоколов прикладного уровня. Таким образом, книга будет очень полезна для всякого читателя, нуждающегося в использовании протоколов, не поддерживаемых в настоящее время в .NET, а также для всех тех, кто хочет овладеть предписанными протоколами.

## **Основные темы книги**

В главе 1 — введение в некоторые основные сетевые понятия и протоколы. Что бы вам ни было нужно от сетевого программирования — разработка серверных приложений, выполняемых как Windows-сервисы, которые предоставляют данные клиентам с использованием специализированного протокола, написание клиентских приложений, запрашивающих данные от Web-серверов, создание широковеб-ательных приложений или приложений, функционирующих как почтовые службы, эта глава будет вашим первым портом захода. Мы начинаем с рассмотрения физической сети и оборудования, применяемого в локальных вычислительных сетях. Затем мы обращаемся к таким вещам, как семиуровневая модель OSI, и рассматриваем соответствие набора протоколов TCP/IP уровням OSI. Наконец мы изучаем разнообразные протоколы сетевого обмена, Интернета и электронной почты.

В главе 2 предоставлены сведения о работе с потоками. Поток — это абстрактное представление последовательного устройства, осуществляющего побайтовое сохранение и считывание данных. Таким устройством может быть, например, файл,

принтер или сетевой сокет. Через эту абстракцию один и тот же процесс может обращаться к разным устройствам, и схожий программный код можно использовать, к примеру, для чтения данных из входного файлового или входного сетевого потоков. В таком случае программист освобождается от необходимости беспокоиться о физической сути конкретного устройства. В данной главе мы рассматриваем потоки в .NET — класс `Stream` — и работаем с конкретным классом `FileStream`. Здесь же освещается чтение из двоичных и текстовых файлов, запись в них и сериализация (преобразование в последовательную форму) объектов в XML и двоичный формат.

Глава 3 помогает приступить к сетевому программированию в .NET с использованием классов из пространства имен `System.Net`. Мы начинаем с обсуждения этих классов — они играют фундаментальную роль во всех последующих главах книги. Точнее говоря, мы узнаем, как работать с URI, IP-адресами и поиском в DNS. Мы увидим, как с помощью классов `WebRequest` и `WebResponse` обрабатывать запросы и ответы, после чего приступим к рассмотрению аутентификации, авторизации и разрешений, относящихся к сетевому программированию.

Глава 4 посвящена программированию сокетов, в ней рассматривается программирование на низших уровнях для выполнения задач сетевого обмена. Сокет — это один конец дуплексного канала связи между двумя программами, которые выполняются в сети. Мы рассмотрим поддержку сокетов в .NET — класс `System.Net.Sockets.Socket` и создадим синхронное и асинхронное приложения “клиент-сервер”.

В главе 5 мы отправимся в путешествие по высокоуровневым сетевым классам в .NET Framework и начнем с рассмотрения `Transmission Control Protocol (TCP)`. Представим общее введение в TCP, его архитектуру и структуры данных, а затем перейдем к исследованию классов `TcpClient` и `TcpListener`, позволяющих работать с TCP. С помощью этих классов мы строим приложения “клиент-сервер”, полнофункциональный почтовый клиент, демонстрирующий мощь `TcpClient`, и создаем работающий по принципу эха многопоточный сервер, опираясь на поддержку классов многопоточной обработки .NET. В заключение главы мы вкратце рассматриваем .NET Remoting Framework и, в частности, транспортный канал `TcpChannel`, обеспечиваемый средой .NET Framework.

В главе 6 рассказывается о классе `UdpClient`, с помощью которого реализуется `User Datagram Protocol (UDP)`. Рассматриваются основы протокола UDP и использование класса `UdpClient`. Хотя протокол TCP надежнее, чем UDP, он значительно увеличивает накладные расходы. Соответственно, UDP работает быстрее и хорошо приспособлен для передачи таких мультимедийных данных, как потоки видеоизображения, в которых точный порядок прибытия пакетов несущественен. В этой главе также рассматриваются высокоуровневые протоколы, основанные на UDP.

Глава 7 посвящена групповой рассылке. Например, благодаря этой технологии в 1994 г. стала возможной трансляция в прямом эфире по Интернету концерта группы “Rolling Stones”. Она позволяет наблюдать за работой космонавтов в космосе или проводить совещания по Интернету. При использовании групповой рассылки сервер должен послать каждое сообщение лишь один раз, и оно распространяется среди целой группы клиентов. Начинается глава со сравнения однонаправленной передачи, групповой рассылки и широковещательной передачи, затем рассматривается архитектура групповой рассылки и реализация сокетов групповой рассылки в .NET. С использованием средств групповой рассылки создаются два приложения Windows. Первое приложение позволяет вести переписку между несколькими системами, каждая из которых является и отправителем, и получателем. Второе приложение демонстрирует изображения, оно показывает, как объемные пакеты данных можно посылать нескольким клиентам, не занимая значительную часть пропускной способности сети.

В главе 8 освещается протокол HTTP и его надежная реализация, предлагаемая в .NET. Важность HTTP как прикладного протокола велика, поскольку в настоящее время он используется в значительной доле Web-трафика. Эта глава начинается с обзора протокола HTTP — заголовков, формата запросов и ответов. Рассматриваются классы в .NET, позволяющие работать с HTTP, и рассказывается, как считывать и записывать cookie. Затем с помощью ASP.NET мы создаем HTTP-сервер и, знакомясь с транспортным каналом HTTP, продолжаем рассмотрение .NET Remoting.

В главе 9 мы принимаемся за электронную почту. Начинаем с высокоуровневого обзора разнообразных почтовых протоколов, узнаем, как в среде .NET получить к ним доступ и как ими пользоваться; рассмотрим основы протоколов SMTP, POP3, IMAP и NNTP и увидим, как они совместно работают при отправке и получении электронных сообщений через Интернет. Мы также рассмотрим отправку сообщений электронной почты через SMTP с использованием классов .NET Framework и разработку некоторых основных классов, реализующих протоколы POP3 и SMTP.

Глава 10 посвящена защите сетевого обмена. Пространство имен System.Security.Cryptography среды .NET Framework обеспечивает программный доступ к разнообразным сервисам шифрования, которые мы включаем в наши приложения, чтобы шифровать и дешифровать данные, гарантировать целостность данных и обрабатывать цифровые подписи и сертификаты. В данной главе исследуем это пространство имен и предоставляем введение в криптографию и все ее ключевые понятия (за каламбур просим прощения). Мы также рассматриваем обеспечение безопасности для приложения форума, созданного в главе 6.

## Предполагаемый круг читателей

Мы освещаем в книге как основные, так и более сложные сетевые понятия. Читатель, знакомый с сетевым программированием в другой среде, сможет довольно быстро освоить содержание этой книги, которая все же будет ему полезна.

**Все примеры программного кода, приведенные в книге, написаны на C#, поэтому предполагается практическое владение читателя этим языком.**

## Что необходимо для работы с книгой

Чтобы запускать примеры, приведенные в книге, вам требуется машина с установленной средой .NET Framework. Это значит, что на ней должна работать одна из следующих операционных систем:

- Windows 2000 Professional (или более высокий уровень)
- Windows XP

Также для этой книги рекомендуется использовать версию Visual Studio .NET.

## Соглашения о стилях

Мы использовали несколько разных стилей текста и разное расположение в книге, чтобы провести различия между типами информации. Здесь приведены примеры использованных стилей и объясняется их значение.

Программный код записывается несколькими шрифтами. Если это слово, о котором идет речь в тексте, например, при обсуждении цикла `for ( . . . )`, то оно запи-

сывается таким шрифтом. Если это блок кода, который можно ввести как программу и запустить на выполнение, то он записывается на сером фоне.

```
IPHostEntry ipHost = Dns.Resolve("127.0.0.1");
```

Иногда можно увидеть код в смешанном стиле, например, так:

```
IPHostEntry ipHost = Dns.Resolve("127.0.0.1");
IPAddress ipAddr = ipHost.AddressList[0];
IPEndPoint ipEndPoint = new IPEndPoint(ipAddr, 11002);
Socket sender = new Socket(AddressFamily.InterNetwork,
    SocketType.Stream, ProtocolType.Tcp);
Sender.Connect(ipEndPoint);
```

В подобных случаях на белом фоне помещается код, с которым мы уже хорошо знакомы. Строка, выделенная серым фоном, содержит дополнение к рассмотренному ранее коду.

*Совет, рекомендация и вводная информация представляются таким типом шрифта.*

**Важная информация выделяется такими прямоугольниками.**

Буллиты (пули) показываются с отступами, каждый буллит отмечает новый пункт перечисления следующим образом:

- **Важные слова** выделяются полужирным шрифтом
- Слова, появляющиеся на экране или в меню, например Open или Close, представлены таким же шрифтом, какой вы могли видеть на рабочем столе Windows
- Клавиши, которые вы нажимаете на клавиатуре, например *Ctrl* и *Enter*, представляются курсивом

## Как скопировать код примеров для этой книги

Войдя на Web-сайт Wrox [www.wrox.com](http://www.wrox.com), найдите название книги с помощью нашего средства Search или используйте один из списков названий. Затем щелкните по ссылке Download Code на странице с подробной информацией о книге или по элементу Download в столбце Code списка названий.

Файлы, доступные для копирования с нашего сайта, архивированы программой WinZip. Когда вы сохраните архивы в какой-либо папке на вашем жестком диске, вам потребуется извлечь файлы из архива с использованием программы распаковки WinZip или PKUnzip. При извлечении файлов код обычно помещается в папки по главам. Начиная процесс извлечения, позаботьтесь, чтобы для вашей утилиты распаковки была установлена возможность использования имен папок.