

ПАВЕЛ АГУРОВ



C#

СБОРНИК РЕЦЕПТОВ



Консольные программы

Программы Windows Forms

Базы данных

Работа с аппаратурой

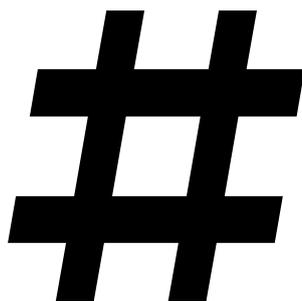
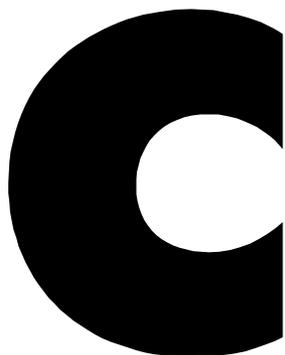
Готовые решения, советы

PRO

ПРОФЕССИОНАЛЬНОЕ
ПРОГРАММИРОВАНИЕ

+ CD

Павел Агуров



СБОРНИК РЕЦЕПТОВ

Санкт-Петербург

«БХВ-Петербург»

2007

УДК 681.3.068+800.92С#
ББК 32.973.26-018.1
А27

Агуров П. В.

А27 С#. Сборник рецептов. — СПб.: БХВ-Петербург, 2007. — 432 с.: ил.
ISBN 5-94157-969-1

В книге содержатся советы, алгоритмы и готовые примеры программ из различных областей: шифрование, файловые и сетевые операции, XML, ASP.NET, взаимодействие с MS Office и Internet Explorer и др. Описаны синтаксис языка С#, вопросы отладки и профилирования приложений, а также проблемы, возникающие при переходе с других языков программирования на язык С#. Рассматриваются примеры наиболее часто используемых регулярных выражений. Отдельная глава посвящена работе с аппаратурой. На компакт-диске размещены все исходные коды, приведенные в книге.

Для программистов

УДК 681.3.068+800.92С#
ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 15.09.06.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 34,83.

Тираж 2000 экз. Заказ №

"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.02.953.Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой
по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-969-1

© Агуров П. В., 2007
© Оформление, издательство "БХВ-Петербург", 2007

Оглавление

Введение	1
Для кого эта книга.....	1
О программном коде.....	1
Краткое описание глав.....	2
Благодарности	3
Обратная связь	4
Глава 1. Технология .NET	5
1.1. На каких платформах запускается .NET?	5
1.2. Какие компиляторы существуют для .NET?.....	5
1.3. Можно ли запустить программу без .NET Framework?	5
1.4. Можно ли вызывать функции Win32 из .NET-программ?	6
1.5. Проблемы установки Framework	6
1.6. Ускорение загрузки модулей	7
1.7. Одновременная установка Visual Studio 2003 и 2005	7
1.8. Не запускается отладчик в Visual Studio	8
1.8.1. Ошибка безопасности	8
1.8.2. Ошибка конфигурирования.....	8
1.8.3. Ошибка 401	9
1.9. Не запускается отладчик в Visual Studio 2005	9
Глава 2. Вопросы синтаксиса C#	11
2.1. Как выглядит простая C#-программа?	11
2.2. Оператор <i>switch</i> можно использовать для строк	11
2.3. Выходные параметры без инициализации	12
2.4. Константы и поля "только для чтения"	13

2.5. Как определить ссылочный тип?	14
2.6. Псевдонимы.....	15
2.7. Что означает символ @ в начале строки?	15
2.8. Что означает символ @ перед идентификатором?.....	15
2.9. Какая разница между оператором <i>is</i> и методом <i>IsSubclassOf</i> ?.....	16
2.10. Какая разница между оператором <i>as</i> и прямым приведением типа?.....	16
2.11. Какая разница между <i>string</i> и <i>System.String</i> ?	16
2.12. Какая разница между <i>override</i> и <i>new</i> ?	17
2.13. Какая разница между <i>==</i> и <i>Equals</i> ?.....	18
2.14. Перекрытие операторов.....	19
2.15. Как создать наследника, если конструктор класса имеет параметры?.....	22
2.16. Что такое делегат?	23
2.17. В чем разница между полем и свойством?	24
2.18. Свойство с индексатором	25
2.19. Можно ли сделать свойство с разным уровнем доступа для <i>get</i> и <i>set</i> ?	26
2.20. Как вызвать методы с одинаковыми именами из интерфейсов?	27
2.21. В чем разница <i>byte</i> и <i>sbyte</i> ?	28
2.22. Как сгенерировать документацию по коду?	28
2.23. Как инициализировать поля класса до вызова конструктора?.....	29
2.24. Использование переменного числа параметров.....	29
2.25. Указатели.....	30
2.26. Слабые ссылки	32
2.27. Файлы ресурсов.....	33

Глава 3. Для программистов C++ и Delphi **35**

3.1. Как удалить объекты, созданные с помощью оператора <i>new</i> ?	35
3.2. Как создать экземпляр класса на стеке?.....	35
3.3. Почему не вызывается деструктор?	35
3.4. Почему в операторе <i>if</i> больше нельзя использовать условия типа <i>int</i> ?	35
3.5. Как создать массив констант?.....	36
3.6. Как инициализировать массив структур?	36
3.7. Оператор <i>switch</i> работает не так, как в C++.....	37
3.8. Есть ли в C# макросы и препроцессор?	38
3.9. Как описать битовые поля?.....	38

Глава 4. Отражение и связанные с ним вопросы **41**

4.1. Создание объекта по имени типа.....	41
4.2. Вызов метода по имени	43
4.3. Загрузка и создание класса из другой сборки.....	44
4.4. Получение списка типов из сборки	44

4.5. Получение всех свойств класса.....	45
4.6. Установка значения <i>private</i> -поля	45
4.7. Получение списка полей структуры	47
4.8. Получение полного имени сборки и ее свойств	48
4.9. Компиляция кода из программы.....	49
4.10. Перехват загрузки типов и модулей.....	50
Глава 5. Отладка и условная компиляция	53
5.1. Условная компиляция в режиме отладки.....	53
5.2. Методы <i>ASSERT</i> и <i>TRACE</i>	53
5.3. Перенаправление вывода <i>Trace</i>	54
5.4. Почему дублируются строки в отладочной информации?	55
5.5. Трассировка исключения	55
5.6. Трассировка стека вызовов	56
5.7. Оценка времени выполнения кода.....	58
5.7.1. Измерение с помощью <i>TickCount</i> (наименьшая точность).....	58
5.7.2. Измерение с помощью <i>Ticks</i> (средняя точность).....	59
5.7.3. Измерение с помощью <i>QueryPerformance</i> (высокая точность)	59
5.8. Как включить полный режим отладки кода?.....	59
5.9. Отладочная информация для ASP.NET.....	60
5.10. Как отлаживать <i>remoting</i> -код?	60
5.11. Глобальная обработка исключений.....	60
Глава 6. Командная строка, свойства программы, конфигурационные файлы.....	63
6.1. Параметры командной строки	63
6.2. Имя исполняемого файла	64
6.3. Установка свойства <i>CompanyName</i>	64
6.4. Конфигурационный файл.....	64
6.5. Где найти класс <i>ConfigurationManager</i> ?	65
6.6. Нестандартный конфигурационный файл.....	65
6.7. Как заставить программу использовать только Framework 2.0?.....	71
6.8. Версия и информация о модуле.....	72
6.9. Как обновлять <i>AssemblyInfo</i> для всего проекта сразу?.....	72
Глава 7. Преобразования.....	73
7.1. Преобразование числа в шестнадцатеричную строку.....	73
7.2. Преобразование строки в число.....	73
7.3. Преобразование строки с пробелами в число.....	74
7.4. Преобразование шестнадцатеричной строки в число.....	74

7.5. Преобразование двоичной строки в число	74
7.6. Преобразование числа в двоичную строку	74
7.7. Преобразование числа с плавающей точкой в строку	74
7.8. Преобразование строки в число с плавающей точкой	75
7.9. Перекодировка текста	77
7.10. Преобразование в <i>Base64</i> и обратно	77
7.11. Преобразование из Win1251 в KOI8 и обратно	77
7.12. Преобразование цвета в строку и обратно	78
7.13. Преобразование цвета в HTML-формат	78
7.14. Преобразование цвета в целое число и обратно	79
7.15. Преобразование HTML-текста	79
7.16. Преобразование массива байтов в базовые типы и обратно	80

Глава 8. Массивы, списки, таблицы, перечисления

8.1. Отображение всех элементов перечисления	81
8.2. Перевод перечисления в строковый вид и обратно	81
8.3. Проверка наличия элемента в перечислении	82
8.4. Одномерные массивы постоянного размера	82
8.4.1. Изменение размерности массива	83
8.4.2. Перебор всех элементов массива	83
8.4.3. Изменение порядка элементов на обратный	84
8.4.4. Поиск	84
8.4.5. Копирование элементов массива	85
8.4.6. Преобразование массива одного типа в массив другого типа	85
8.5. Многомерные массивы постоянного размера	85
8.6. Невыровненный массив	86
8.7. Свойства массивов	86
8.8. Методы, возвращающие массивы	87
8.9. Массивы переменного размера	88
8.10. Массив битов	88
8.11. Хранение таблицы "ключ/значение"	89
8.11.1. Добавление элементов	90
8.11.2. Отображение содержимого <i>Hashtable</i>	90
8.11.3. Порядок элементов <i>Hashtable</i>	90
8.11.4. Удаление элемента из <i>Hashtable</i>	91
8.11.5. Специальный <i>Hashtable</i>	91
8.12. Сортированный список	92
8.13. Очередь	94
8.14. Стек	94
8.15. Преобразование <i>IList</i> в <i>ArrayList</i>	95

Глава 9. Шифрование, кодирование, сжатие, математика	97
9.1. Вычисление контрольной суммы строки	97
9.2. Шифрование строк	98
9.3. Контроль арифметических операций	103
9.4. Вычисление математических выражений	104
Глава 10. Строковые и символьные операции	105
10.1. Проверка символов	105
10.2. Создание строки одинаковых символов	105
10.3. Проверка, является ли строка числом	106
10.4. Проверка, является ли строка допустимой датой	106
10.5. Разбиение строки по разделителям	106
10.6. Объединение строк через разделитель	107
10.7. Какие преимущества дает класс <i>StringBuilder</i> ?	107
10.8. Как сравнить все строки без учета регистра символов?	108
10.9. Прямая модификация содержимого строки	108
10.10. Специальные символы в строке	108
Глава 11. Консольные программы	111
11.1. Вывод цветного текста	111
11.2. Задержка закрытия консольной программы	114
11.3. Перенаправление вывода консольной программы	114
11.4. Изменение заголовка консольной программы	115
11.5. Получение размера окна консольной программы	116
Глава 12. Производительность	117
12.1. Не кэшируйте соединение с БД	117
12.2. Используйте <i>StringBuilder</i>	117
12.3. Используйте <i>Length</i> для проверки пустоты строки	118
12.4. Не используйте исключения, если это возможно	119
12.5. Используйте <i>AddRange</i> для добавления групп элементов	119
12.6. Создавайте <i>Hashtable</i> и <i>ArrayList</i> подходящего размера	119
12.7. Иногда выгоднее использовать невыровненный массив	120
12.8. Используйте <i>DataReader</i> для последовательного доступа к данным	120
12.9. Используйте хранимые процедуры	120
Глава 13. Списки файлов, каталогов	121
13.1. Получение списка логических дисков	121
13.2. Получение списка сетевых дисков	122
13.3. Список каталогов	122

13.4. Список каталогов по маске	123
13.5. Список файлов.....	123
13.6. Список файлов по маске.....	124
13.7. Создание временного файла.....	124

Глава 14. Файловые операции..... 125

14.1. Блокирование ошибки "Устройство не готово"	125
14.2. Получение списка файлов и каталогов.....	126
14.3. Диалог открытия папки	127
14.4. Основные операции с каталогами	127
14.5. Основные операции с файлами.....	128
14.6. Чтение и установка атрибутов файла	128
14.7. Создание и чтение бинарного файла	128
14.8. Создание текстового файла.....	129
14.9. Добавление в текстовый файл.....	130
14.10. Чтение и запись в файл строк на русском языке	130
14.11. Посимвольное чтение текстового файла.....	130
14.12. Построчное чтение текстового файла	131
14.13. Чтение файла полностью.....	131
14.14. Отслеживание изменений в файловой системе	132
14.15. Получение короткого имени файла из длинного и наоборот.....	133

Глава 15. Сетевая информация и сетевые операции 135

15.1. Разбор URL на составляющие.....	135
15.2. Получение DNS-имени компьютера.....	136
15.3. Получение NetBios-имени компьютера	136
15.4. Получение имени хоста	136
15.5. Получение списка IP-адресов компьютера.....	136
15.6. Перечисление RAS-соединений.....	137
15.7. Отправка письма через SMTP.....	140
15.7.1. Отправка письма через SMTP с авторизацией.....	141
15.7.2. Отправка письма через PickUp-каталог	142
15.7.3. Задание кодировки сообщения	142
15.7.4. Отправка письма без использования <i>System.Web.Mail</i>	142
15.8. Получение списка сетевых дисков	146
15.9. Получение имени текущего пользователя	147
15.10. Программная имперсонация	147
15.11. Как получить список групп домена, в которые входит пользователь?	149
15.12. Получение файла из Интернета	150

15.13. Получение данных из Интернета.....	151
15.14. Получение Web-страницы.....	151
15.15. Использование прокси-сервера.....	152
15.16. Подключен ли компьютер к Интернету?	152
15.17. Как проверить, доступен ли компьютер в сети?.....	153
15.18. Как сделать ping?	155
15.19. Получение файла с FTP через WinInet-функции.....	156
15.20. Перечисление компьютеров в сети.....	159

Глава 16. Рабочий стол и панель управления..... 163

16.1. Что такое <i>IWshRuntimeLibrary</i> и <i>Shell32</i> ?.....	163
16.2. Создание ярлыка	164
16.3. Как найти путь к папке автозапуска?	165
16.4. Регистрация меню запуска программы по расширению	165
16.5. Доступ к элементам панели управления	167
16.6. Как получить разрешение экрана?.....	168

Глава 17. Поток..... 169

17.1. Запуск и остановка потоков	169
17.2. Отмена остановки потока.....	174
17.3. Изолирование данных потока	174
17.4. Права потоков	177
17.5. Идентификаторы потоков	178
17.6. Пул потоков	178
17.7. Синхронизация потоков	179
17.8. Синхронизация потоков: "один пишет, многие читают".....	180
17.9. Использование <i>AutoResetEvent</i>	180

Глава 18. Процессы..... 183

18.1. Запуск внешних программ	183
18.2. Блокировка запуска второй копии программы.....	184
18.2.1. Вариант 1	184
18.2.2. Вариант 2	185
18.3. Информация о процессах	186
18.4. Определение всех процессов, использующих .NET	188
18.5. Перечисление всех оконных процессов	189
18.6. Запуск внешнего процесса и вывод результата на консоль (Windows NT)	189
18.7. Получение списка модулей, связанных с текущим процессом.....	191
18.8. Управление сервисами	192

Глава 19. Разное.....	195
19.1. Перечисление всех принтеров системы	195
19.2. Установка принтера по умолчанию	196
19.3. Определение версии операционной системы	196
19.4. Получение текущей даты и времени	197
19.5. Установка значения десятичного разделителя	197
19.6. Определение числа процессоров	198
19.7. Запись в системный журнал событий.....	198
19.8. Создание своего журнала событий.....	198
19.9. Как "отвязать" программу от журнала событий.....	200
19.10. Перечисление доступных журналов событий.....	201
19.11. Звуковой сигнал	202
19.12. Генерация случайного числа	202
19.13. Принудительная сборка мусора.....	203
19.14. Получение пути к папке Framework	203
19.15. Проигрывание WAV-файла.....	204
19.16. Чтение реестра.....	205
19.17. Разрешение обращения к реестру.....	206
19.18. Сохранение объекта в реестре	206
19.19. Определение типа файла по расширению.....	208
19.20. Создание GUID	209
19.21. Блокировка компьютера	210
19.22. Как создать COM-объект по GUID?.....	211
19.23. Почему уничтожается remoting-объект?	211
19.24. Ошибка <i>SerializationException: "BinaryFormatter Version incompatibility"</i> при использовании .NET Remoting.....	211
19.25. Использование Speech API.....	212
Глава 20. Регулярные выражения	217
20.1. Поиск совпадения	217
20.2. Поиск и получение совпавшего текста.....	218
20.3. Поиск с заменой	219
20.4. Разделение строки на слова.....	219
20.5. Разбор CSV-файла с учетом кавычек	220
20.6. Выделение макроимен	221
20.7. Изменение формата даты	222
20.8. Замена заголовка HTML-файла	222
20.9. Тестирование регулярных выражений	222
20.10. Примеры регулярных выражений.....	223
20.10.1. E-mail.....	223

20.10.2. Положительные десятичные числа.....	223
20.10.3. HTML-теги	223
20.10.4. HTML-теги с Java-привязкой	223
20.10.5. HTML-теги с атрибутами	224
20.10.6. Строки HTML-цветов	224
20.10.7. XML-теги (закрытые теги)	224
20.10.8. MAC-адрес.....	224
20.10.9. Имя макроса (формат @аимя@а).....	225
20.10.10. Время	225
Глава 21. XML	227
21.1. Загрузка содержимого XML из файла.....	227
21.1.1. Вариант 1	227
21.1.2. Вариант 2	228
21.2. Загрузка содержимого XML из URL.....	228
21.3. Загрузка содержимого XML из строки	229
21.4. Обход всех элементов XML-файла.....	229
21.5. Поэлементное чтение XML-файла	230
21.6. Чтение атрибутов	231
21.7. Чтение всех атрибутов.....	231
21.8. Запись атрибутов.....	232
21.9. Запись данных в XML-файл.....	232
21.10. Запись комментариев в XML-файл	233
21.11. Запись пространства имен и префиксов в XML-файл	233
21.12. Запись в XML-файл со специальным форматированием	233
21.13. Выборка из XML с помощью XPath.....	234
21.14. Вычисление выражений с помощью XPath	235
21.15. Вычисление <i>min</i> и <i>max</i> с помощью XPath.....	235
21.16. Создание <i>XPathDocument</i> из строки	236
21.17. XSL-трансформация	236
21.18. XSL-трансформация с параметрами.....	237
21.19. Проверка файла с помощью XSD.....	238
21.20. В чем разница между <i>InnerXml</i> , <i>OuterXml</i> и <i>InnerText</i> ?	240
21.21. В чем разница между <i>Load</i> и <i>LoadXml</i> класса <i>XmlDocument</i> ?	241
21.22. Как добавить <i>XmlNode</i> из одного документа в другой?.....	241
21.23. Аналог простого ini-файла для хранения настроек	241
21.24. Работа с ini-файлами.....	242
Глава 22. Windows Forms.....	243
22.1. Сворачивание программы в трей.....	243
22.2. Как сделать форму, не видимую в панели задач?	244

22.3. Как скрыть главную форму при старте приложения?.....	244
22.4. Получение всех цветов в системе	244
22.5. Получение положения курсора мыши.....	245
22.6. Как отлавливать все исключения?	245
22.7. Как сделать обработчик для сообщений Win32?.....	246
22.8. Есть ли в C# аналог <i>ProcessMessages</i> из Delphi?.....	247
22.9. Проблемы с <i>ImageList</i> при использовании <i>EnableVisualStyles</i>	247
22.10. Как поменять фон MDI-окон?.....	248
22.11. Определение разрешения экрана	248
22.12. Определение рабочей области экрана без системного трея	248
22.13. Использование системного буфера обмена	248
22.14. Обработка изменения системных настроек (например, разрешения экрана)	249
22.15. Показ HTML на форме	250
22.16. Отображение подсказки из справочных файлов	250
22.17. Рисование без обработки события <i>Paint</i>	251
22.18. Как нарисовать точку?.....	251
22.19. Проверка на попадание в область.....	251
22.20. Вывод на консоль из приложения Windows Form.....	252
22.21. Использование автозавершения в текстовом поле.....	255

Глава 23. Базы данных.....259

23.1. Что следует использовать для закрытия соединения — <i>Close</i> или <i>Dispose</i> ?	259
23.2. Работа с файлами MS Access	259
23.3. Получение индекса объекта после добавления его в таблицу.....	260
23.4. Можно ли сделать <i>GroupBy</i> в <i>DataSet</i> ?	261
23.5. Перечисление доступных SQL-серверов	261
23.6. Создание расширений MS SQL 2005.....	264

Глава 24. ASP.NET267

24.1. Преобразование относительного пути в абсолютный	267
24.2. Управление буферизацией страниц.....	267
24.3. Управление кэшированием страниц.....	267
24.4. Перенаправление на другую страницу	268
24.5. Чем метод <i>Server.Transfer</i> отличается от метода <i>Response.Redirect</i> ?	268
24.6. Перенаправление на другую страницу при исключении	269
24.7. Использование cookies.....	269
24.8. Установка фокуса на элемент управления.....	270
24.9. Просмотр исходного кода страницы	271
24.10. Открытие страницы по кнопке в новом окне	271

24.11. Обработка исключений на странице	271
24.12. Глобальная обработка исключений ASP.NET	272
24.13. Ссылка на файл для скачивания	274
24.14. Управление созданием обработчиков <i>IHttpHandler</i>	275
24.15. Сложная привязка данных с помощью <i>DataBinder</i>	276
24.16. Что плохого в использовании сессий?	277
24.17. Настройка хранения сессий.....	277
24.18. Передача между страницами значений серверного элемента управления.....	278
24.19. Как перехватить загрузку и сохранение <i>ViewState</i> ?	280
24.20. Создание общей сессии между ASP- и ASP.NET-приложениями	280
24.21. Что такое AJAX?	280
24.22. Получение серверных данных без AJAX	287
24.23. Получение HTML-кода элемента	290
24.24. Печать страницы на принтер по умолчанию	291
24.25. Проблемы несовместимости браузеров	292
24.26. Быстрая генерация табличного представления Web-формы.....	292
24.27. Почему <i>Page_Load</i> вызывается два раза?.....	293
24.28. Получение пути к странице.....	293

Глава 25. MS Office, Internet Explorer.....295

25.1. Получение данных со страницы MS Excel.....	295
25.2. Доступ к данным MS Excel	296
25.3. Работа с данными в буфере обмена в Excel-формате CSV.....	299
25.4. Как обойти проблемы с использованием объектов MS Office?	299
25.5. Как напечатать документ MS Office?	300
25.6. Формирование документов MS Excel и MS Word из ASP.NET	300
25.7. Работа с MS Office через DDE	302
25.8. Работа с календарем MS Outlook.....	302
25.9. Использование Word Spell Checker для проверки орфографии	303
25.10. Журнал истории Internet Explorer	308

Глава 26. WMI311

26.1. Что такое WMI?	311
26.2. Работа с WMI на удаленной машине.....	311
26.3. Получение свойств видеоконтроллера и частоты обновления экрана	312
26.4. Получение информации о компьютере.....	313
26.5. Получение информации о производителе.....	314
26.6. Получение информации об операционной системе	315
26.7. Выполнение <i>logoff</i> , <i>shutdown</i> , <i>reboot</i>	320
26.8. Получение информации о рабочем столе	321

26.9. Определение типа компьютера	323
26.10. Определение физических параметров компьютера	324
26.11. Установка имени компьютера.....	328
26.12. Получение информации о процессорах	328
26.13. Получение списка общих файлов и каталогов	334
26.14. Создание и удаление общего каталога	335
26.15. Перечисление подключенных сетевых ресурсов	336
26.16. Получение информации о диске	337
26.16.1. Вариант 1	337
26.16.2. Вариант 2	338
26.17. Получение переменных окружения.....	338
26.18. Получение информации о загрузчике системы	339
26.19. Получение списка процессов	340
26.20. Получение списка запущенных и остановленных сервисов.....	341
26.21. Получение информации о разделах диска	342
26.22. Получение учетных записей локальной машины или домена.....	344
26.23. Получение списка групп локальной машины или домена.....	345

Глава 27. Работа с аппаратурой

27.1. Импортирование функций Setup API	349
27.2. Перечисление устройств.....	352
27.3. Получение состояния устройства	355
27.4. Программное отключение устройства	358
27.5. Отслеживание изменения аппаратной конфигурации	360
27.6. Перечисление устройств ввода с помощью функций Direct X	361
27.7. Чтение скан-кодов клавиатуры с помощью функций Direct X	362
27.8. Чтение данных с последовательного порта	363
27.8.1. Вариант 1	363
27.8.2. Вариант 2	366
27.8.3. Вариант 3	371

Глава 28. Изображения

28.1. Изменение размеров картинки.....	373
28.2. Доступ к пикселям изображения	374
28.3. Создание негатива изображения.....	374
28.4. Изменение цветовой матрицы изображения.....	375
28.4.1. Преобразование цветов изображения	375
28.4.2. Создание серого изображения	376
28.4.3. Создание "затемненного" изображения	377

28.5. Рисование заблокированной кнопки.....	377
28.6. Создание серого оттенка цвета	377
28.7. Загрузка изображения из буфера обмена	378
28.8. Замена цвета в изображении	379
28.9. Преобразование значков в изображение.....	380
28.10. Создание графической экранной копии формы	380
28.10.1. Вариант 1	380
28.10.2. Вариант 2	381
28.11. Создание графической копии экрана	382
28.12. Изменение размера изображения	383
28.13. Создание пиктограммы	384
28.14. Анимированный GIF.....	385
28.15. Градиентная заливка	386
28.16. Исключение при установке прозрачного фона.....	386
28.17. Как нарисовать математическую формулу?	387

Приложение 1. Интернет-ресурсы	389
---------------------------------------------	------------

Приложение 2. Описание компакт-диска.....	391
--------------------------------------------------	------------

Список литературы	401
--------------------------------	------------

Предметный указатель	403
-----------------------------------	------------

Введение

Огромное количество классов Framework, с одной стороны, позволяет выполнять множество операций, которые в других языках программирования требовали установки дополнительных компонентов (только не подумайте, что я говорю, будто бы C# хуже или лучше C++ и Delphi). С другой стороны — это оборачивается необходимостью помнить множество классов и методов. Ничего страшного в этом нет, но мне всегда хотелось минимизировать затраты на поиск стандартных решений.

За несколько лет использования C# у меня накопилось множество небольших программ, решающих каждодневно возникающие задачи. Если мне нужно выполнить проверку XML-файла или обратиться к устройству USB — я просто открываю свой сборник и копирую из него нужный фрагмент кода. Если необходимого примера там нет, я его добавляю. Кроме того, сюда включены некоторые вопросы, которые я задаю на интервью при приеме на работу. Так накопился не очень большой, но, как мне кажется, полезный сборник небольших примеров или просто ответов на часто возникающие вопросы, который я и решился представить вашему вниманию.

Для кого эта книга

Эта книга для программистов-практиков. Не ищите в ней теоретических знаний — для этого есть множество замечательных изданий. Эта книга — набор готовых решений, советов и исходного кода.

Предполагается, что читатель знаком с синтаксисом языка C# и имеет представление об архитектуре платформы .NET.

О программном коде

Код, приведенный в книге, собирался в течение нескольких лет из различных источников. Это и книги, и Интернет, и вопросы интервью... Я старался упоминать автора, но, конечно, это не всегда было возможно.

Каждый листинг, приведенный в книге, проверялся и при необходимости корректировался.

Из-за того что книга содержит, по большей части, неполные листинги программ (в большинстве примеров удалена автоматически генерируемая и системная информация), мне пришлось пойти на некоторый компромисс. С одной стороны, использование оператора `using` прибавило бы читабельности кода, но с другой — необходимость каждый раз давать пояснение, что "для работы этого фрагмента кода следует указывать следующие описания...", усложнило бы и без того нелегкую жизнь. Итак, директивы `using` во фрагментах кода не указываются, зато имена классов приводятся полностью.

Полный и компилируемый исходный код содержится на компакт-диске.

Краткое описание глав

Книга состоит из 28 глав, двух приложений и списка литературы.

В главах 1—12 обсуждаются общие вопросы платформы .NET.

- *Глава 1* описывает общие вопросы, связанные с платформой .NET, установкой Microsoft Visual Studio и установкой Framework.
- *Глава 2* содержит некоторые конструкции C#, которые могут оказаться небесполезными в повседневной работе, а также отвечает на несколько каверзных вопросов, которые часто задаются на интервью.
- *Глава 3* будет интересна тем, кто переходит на C# с других языков программирования.
- *Глава 4* содержит примеры работы с технологией работы с информацией времени исполнения.
- *Глава 5* описывает вопросы, связанные с отладкой кода и возникающими при этом проблемами.
- *Глава 6* содержит информацию обо всем, что связано с командной строкой, конфигурациями и свойствами программ.
- *Глава 7* предлагает примеры всевозможных преобразований, как строки в число, так и кодировки Win1251 в KOI-8 или HTML в обычный текст.
- *Глава 8* содержит информацию о всевозможных операциях с множествами.
- *Глава 9* содержит примеры кода, позволяющего зашифровать строку, вычислить контрольную сумму и т. п.
- *Глава 10* содержит информацию, связанную со строками.
- *Глава 11* предоставляет примеры кода, полезного при написании консольных программ.

□ *Глава 12* содержит советы по повышению производительности приложений.

Главы 13—14 посвящены файлам и файловым операциям.

□ *Глава 13* содержит примеры работы со списками файлов и каталогов.

□ *Глава 14* предлагает примеры работы с файлами и каталогами.

Глава 15 содержит примеры работы с сетью, сетевые операции и работу с Интернетом.

В *главах 16—19* обсуждаются примеры системных операций.

□ *Глава 16* описывает работу с ярлыками программ, папкой автозапуска, получение разрешения экрана и т. п.

□ *Глава 17* показывает примеры работы с потоками и описывает вопросы синхронизации.

□ *Глава 18* содержит примеры запуска внешних процессов, получение списка процессов, управление сервисами и т. д.

□ *Глава 19* содержит множество интересных примеров, которым не нашлось место в предыдущих главах.

Глава 20 посвящена регулярным выражениям. Это исключительно мощный инструмент! Об этом надо обязательно знать и уметь им пользоваться.

Глава 21 посвящена работе с XML и всем связанным с ним вопросам.

Глава 22 содержит примеры программ, отвечающих на часто возникающие вопросы при разработке приложений Windows Application.

Глава 23 отвечает на некоторые вопросы, связанные с разработкой баз данных.

Глава 24 содержит решение вопросов, возникающих при разработке приложений ASP.NET.

Глава 25 показывает примеры взаимодействия с MS Office и Internet Explorer.

Глава 26 содержит примеры работы с WMI.

Глава 27 показывает примеры работы с аппаратурой, в частности, с COM- и USB-портами.

Глава 28 демонстрирует примеры работы с изображениями: градиентная заливка, создание скриншотов, преобразование форматов изображений и т. д.

Благодарности

Хочется поблагодарить всех сотрудников компании EPAM (www.epam.com), где я имею честь работать уже несколько лет. Работать в компании более по-

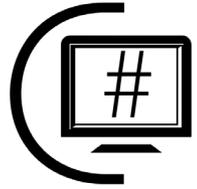
лутора тысяч человек действительно интересно, и это не могло не повлиять на наполнение моего сборника.

Благодарю всех сотрудников издательства "БХВ-Петербург", помогавших мне в процессе создания книги, особенно Игоря Шишигина и Евгения Рыбакова.

Обратная связь

Почему-то хочется верить, что в этой книге нет ошибок и опечаток. Но в действительности, это, конечно же, не так. Если вы найдете неточности или ошибки, или просто захотите пообщаться — пишите **mail@bhv.ru**.

ГЛАВА 1



Технология .NET

1.1. На каких платформах запускается .NET?

Для работы программ поддерживаются Windows XP, Windows 2000, NT4 SP6a и Windows ME/98. Операционная система Windows 95 не поддерживается. На некоторых платформах поддерживается не весь набор функциональности, например, ASP.NET работает только на Windows XP и Windows 2000. Операционные системы Windows 98/ME не могут использоваться для разработки. IIS не поддерживается в Windows XP Home Edition, которая также не может выступать в качестве сервера ASP.NET. Однако Web-сервер ASP.NET Web Matrix может работать на Windows XP Home Edition. Проект Mono (www.go-mono.com) позволяет работать на Linux. Еще один Framework-проект — DotGNU — представлен на сайте dotgnu.org.

1.2. Какие компиляторы существуют для .NET?

Microsoft предлагает компиляторы для C#, C++, VB и JScript. Другие производители предлагают компиляторы для COBOL, Eiffel, Perl, Smalltalk, Python и т. д. Обширный список языков можно найти на странице www.dotnetpowered.com/languages.aspx.

1.3. Можно ли запустить программу без .NET Framework?

Нет, нельзя.

Однако компания RemoteSoft (<http://www.remotesoft.com/linker>) выпустила полезный инструмент, который умеет "привязывать" исполняющую среду к программе на .NET.

В Windows Vista ситуация изменится — библиотека .NET будет встроена в операционную систему.

1.4. Можно ли вызывать функции Win32 из .NET-программ?

Да, можно. Для этого надо описать прототип функции с атрибутом `DllImport`. Для работы требуется библиотека `System.Runtime.InteropServices`, как показано в листинге 1.1.

Листинг 1.1. Вызов функции Win32 из .NET-программы

```
using System;
using System.Runtime.InteropServices;

namespace MessageBox
{
    class MainApp
    {
        [DllImport("user32.dll", EntryPoint="MessageBox",
            SetLastError=true, CharSet=CharSet.Auto)]
        public static extern int MessageBox(int hWnd, String strMessage,
            String strCaption, uint uiType);
        public static void Main()
        {
            MessageBox(0, "Вызов функции Win32!", ".NET", 0);
        }
    }
}
```

На сайтах custom.programming-in.net и pinvoke.net можно найти множество оберток для функций Win32.

1.5. Проблемы установки Framework

При попытке установки пакета Framework появляются странные ошибки, установка обрывается. Одна из вероятных причин — антивирус. Особенно это относится к DrWeb. Попробуйте удалить антивирус из системы, установить Framework и проинсталлировать антивирус снова.

1.6. Ускорение загрузки модулей

С помощью утилиты Ngen.exe можно загрузить модуль в специальный кэш. Это позволит ускорить загрузку и исполнение модуля:

```
ngen [options] [assemblyName | assemblyPath]
```

Например, загрузка в кэш:

```
ngen MyAssembly.dll
```

Удаление из кэша:

```
ngen /delete MyAssembly.dll
```

1.7. Одновременная установка Visual Studio 2003 и 2005

После установки Visual Studio 2005 и APS 2.0 предыдущая версия (2003) не работает. Ошибка выглядит примерно так:

*Auto-attach to process '[1312] aspnet_wp.exe' on machine 'EPSW012' failed.
Error code 0x8013134b.*

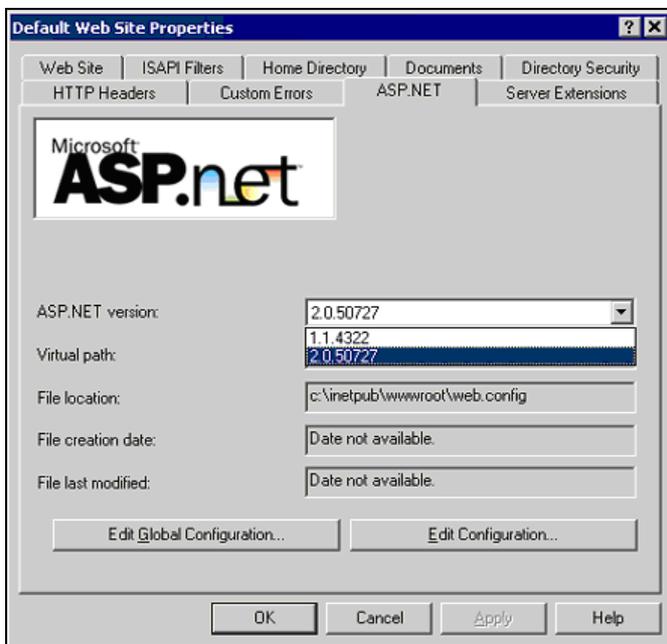


Рис. 1.1. Переключение версии ASP.NET

Для исправления проблемы следует зайти в настройки **Default Web Site** в **IIS**, выбрать команду **Свойства** в контекстном меню и переключить версию ASP на нужную (рис. 1.1). Если это не помогло, следует выполнить команду:

```
aspnet_regiis.exe -i
```

После установки новой версии Enterprise Manager следует перерегистрировать модуль SQLDMO.DLL.

1.8. Не запускается отладчик в Visual Studio

1.8.1. Ошибка безопасности

При запуске отладчика возникает ошибка:

Error while trying to run project: Unable to start debugging on the web server. You do not have permissions to debug. Verify that you are a member of the 'Debugger Users' group on the sever.

Как минимум, должны быть выполнены следующие условия:

- пользователь, с правами которого производится запуск отладчика, должен входить в группу `Debugger Users`;
- в **IIS** должна быть включена опция **Integrated Windows Authentication**;
- должна быть выставлена опция `debug="true"` в файле `web.config`;
- должна быть включена опция **Enable ASP.NET debugging** в свойствах проекта.

Если это не помогает, включите сайт **http://localhost** в список **Trusted Sites**. Второй вариант — попробовать сбросить настройки безопасности Internet Explorer в значения по умолчанию (**Reset IE security settings**).

Ссылки на статьи:

- www.gotdotnet.com/team/csharp/learn/whitepapers/howtosolvedebuggerproblems.doc;
- support.microsoft.com/default.aspx?scid=kb;EN-US;319842.

1.8.2. Ошибка конфигурирования

Сообщение об ошибке имеет вид:

Unable to start debugging on the web server. The project is not configured to be debugged.

Для ее исправления следует открыть конфигурацию сайта в **IIS** и нажать кнопку **Create**.

Ссылки на статью:

- ❑ www.prezzatech.com/kb/articles/kb-1020-unable_to_start_debuggin_on_web_server.asp;
- ❑ www.cryer.co.uk/brian/mswinswdev/ms_ybnet_unable_start_debugging.htm.

1.8.3. Ошибка 401

Сообщение об ошибке имеет вид:

HTTP 401.1 — Unauthorized: Logon Failed.

Для исправления Microsoft советует проверить наличие ключей в реестре в ветке

`HKKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa`

Ключ `DisableLoopbackCheck` должен иметь значение 1 (тип `DWORD`). Если такого ключа нет, то его нужно создать.

Ссылка на статью: support.microsoft.com/default.aspx?scid=kb;en-us;896861.

1.9. Не запускается отладчик в Visual Studio 2005

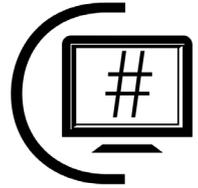
При запуске отладчика в Visual Studio 2005 возникает ошибка:

Error while trying to run project: Unable to run debugging. The binding handle is invalid.

Для решения проблемы следует зайти в свойства проекта (**Project | [ProjectName] Properties | Debug**) и выключить опцию **Enable the Visual Studio hosting process**. Если проект состоит из нескольких модулей, нужно выполнить эту процедуру для каждого модуля.

Ссылка на статью в MSDN: lab.msdn.microsoft.com/productfeedback/ViewWorkaround.aspx?FeedbackID=FDBK36247#1.

ГЛАВА 2



Вопросы синтаксиса C#

2.1. Как выглядит простая C#-программа?

В C# не существует глобальных функций и переменных, поэтому минимальная программа должна содержать хотя бы один класс (для простой программы указание пространства имен не обязательно, хотя среда Visual Studio вставляет это описание по умолчанию):

```
class CApplication
{
    public static void Main()
    {
        System.Console.WriteLine("Минимум C#");
    }
}
```

2.2. Оператор *switch* можно использовать для строк

В C# оператор *switch* может работать со строками так же, как с целыми числами (листинг 2.1). Его использование значительно нагляднее и эффективнее, чем набор операторов сравнения.

Листинг 2.1. Работа с оператором *switch*

```
string command;
.....
switch (command) {
```

```
// Вычисление среднего арифметического
case "MEAN":
{
    .....
    break;
}
// Найти минимальное число
case "MIN":
{
    .....
    break;
}
// Найти максимальное число
case "MAX":
{
    .....
    break;
}
default:{
    Console.WriteLine("Неправильная команда!");
    break;
}
}
```

Однако, в отличие от других языков, C# требует обязательного наличия оператора `break` в конце каждой ветки. Для перехода между ветками использует оператор `goto` (см. разд. 3.7).

2.3. Выходные параметры без инициализации

В C# требуется обязательная инициализация переменных перед любым их использованием. Однако иногда предварительная инициализация переменных выглядит несколько странно:

```
void SomeFunc(ref int out_var)
{
    out_var = 1;
}
.....
int index= 0;
SomeFunc(ref index);
```

Как видно, требования синтаксиса C# выполнены, но с другой стороны, инициализация параметра, значение которого будет вычислено внутри метода,

выглядит несколько странно. Проблема заключается в том, что компилятор не может знать, будет ли значение этой переменной использоваться в методе до первой инициализации или нет. Для явного обозначения параметра как выходного применяется ключевое слово `out`:

```
void SomeFunc(out int out_var)
{
    // Использование out-параметра без инициализации
    // приведет к ошибке компиляции
    // int test = out_var;

    out_var = 1;
}
.....
int index;
SomeFunc(out index);
```

Теперь компилятор будет блокировать использование переменной `out_var` внутри метода до ее инициализации.

2.4. Константы и поля "только для чтения"

В стандарте C# предусматриваются константы и поля "только для чтения" (`readonly`):

```
public readonly int ReadonlyValue = 22;
public const      int ConstValue  = 15;
```

Разница между этими описаниями заключается в том, что константы вычисляются на стадии компиляции, а значения "только для чтения" определяются лишь на стадии выполнения программы. Одним из последствий этого отличия является ситуация, когда библиотека и использующая ее программа компилируются отдельно. Если в библиотеке использовать константу, то при изменении ее значения (и перекомпиляции библиотеки) нужно перекомпилировать и программу. Если же использовать поля "только для чтения", то программу перекомпилировать не надо, т. к. значение поля определяется на стадии исполнения.

Так как `readonly`-поля определяются на стадии выполнения, то следующий код будет нормально компилироваться и работать:

```
class Test
{
    public string Name;
}
```