

Содержание

Об авторах	17
Введение	19
Часть I. Создание ваших первых программ на C#	27
Глава 1. Создание вашей первой Windows-программы на C#	29
Введение в машинные языки, C# и платформу .NET	29
Что такое программа?	30
Что такое C#?	30
Что такое .NET?	31
Что такое Visual Studio 2005? Visual C#?	32
Создание Windows-приложения на языке C#	32
Создание шаблона	33
Компиляция и запуск вашей первой программы Windows Forms	36
Украшение программы	37
Учим форму трудиться	42
Проверка конечного продукта	43
Программисты на Visual Basic 6.0, берегитесь!	44
Глава 2. Создание консольного приложения на C#	47
Создание шаблона консольного приложения	47
Создание исходной программы	47
Пробная поездка	49
Создание реального консольного приложения	49
Изучение шаблона консольного приложения	51
Схема программы	51
Комментарии	51
Тело программы	52
Часть II. Основы программирования в C#	55
Глава 3. Объявление переменных-значений	57
Объявление переменной	57
Что такое int	58
Правила объявления переменных	59
Вариации на тему int	59
Представление дробных чисел	60
Работа с числами с плавающей точкой	61
Объявление переменной с плавающей точкой	62
Более точное преобразование температур	63

Ограничения переменных с плавающей точкой	63
Десятичные числа — комбинация целых и чисел с плавающей точкой	64
Объявление переменных типа decimal	64
Сравнение десятичных, целых чисел и чисел с плавающей точкой	65
Логичен ли логический тип?	65
Символьные типы	66
Тип char	66
Специальные символы	66
Тип string	67
Что такое тип-значение?	67
Сравнение string и char	68
Объявление числовых констант	69
Преобразование типов	70
Глава 4. Операторы	73
Арифметика	73
Простейшие операторы	73
Порядок выполнения операторов	74
Оператор присваивания	75
Оператор инкремента	76
Логично ли логическое сравнение?	77
Сравнение чисел с плавающей точкой	78
Составные логические операторы	79
Тип выражения	80
Вычисление типа операции	80
Типы при присваивании	82
Немного экзотики — тернарный оператор	83
Глава 5. Управление потоком выполнения	85
Управление потоком выполнения	86
Оператор if	86
Инструкция else	89
Как избежать else	90
Вложенные операторы if	90
Циклы	93
Цикл while	93
Цикл do...while	98
Операторы break и continue	98
Цикл без счетчика	99
Правила области видимости	103
Цикл for	104
Пример	104
Зачем нужны разные циклы	105
Вложенные циклы	106
Конструкция switch	109
Оператор goto	111

Часть III. Объектно-основанное программирование	113
Глава 6. Объединение данных — классы и массивы	115
Классы	115
Определение класса	116
Что такое объект	117
Доступ к членам объекта	117
Ссылки	120
Классы, содержащие классы	122
Статические члены класса	123
Определение константных членов-данных	124
Массивы C#	124
Зачем нужны массивы	125
Массив фиксированного размера	125
Массив переменного размера	127
Массивы объектов	130
Конструкция foreach	133
Сортировка массива объектов	134
Глава 7. Функции функций	141
Определение и использование функции	141
Использование функций в ваших программах	143
Аргументы функции	149
Передача аргументов функции	150
Передача функции нескольких аргументов	150
Соответствие определений аргументов их использованию	152
Перегрузка функции	153
Реализация аргументов по умолчанию	154
Передача в функцию типов-значений	156
Возврат значений из функции	162
Возврат значения оператором return	162
Возврат значения посредством передачи по ссылке	163
Когда какой метод использовать	163
Определение функции без возвращаемого значения	166
Передача аргументов в программу	167
Передача аргументов из приглашения DOS	169
Передача аргументов из окна	170
Передача аргументов в Visual Studio 2005	173
Глава 8. Методы класса	177
Передача объекта в функцию	177
Определение функций объектов и методов	179
Определение функций — статических членов	179
Определение метода	181
Полное имя метода	182
Обращение к текущему объекту	183
Ключевое слово this	185

Когда <code>this</code> используется явно	185
Что делать при отсутствии <code>this</code>	188
Помощь от Visual Studio — автоматическое завершение	190
Справка по встроенным функциям системной библиотеки	191
Помощь при использовании ваших собственных функций и методов	192
Внесение дополнений в справочную систему	193
Генерация XML-документации	197
Глава 9. Работа со строками в C#	199
Основные операции над строками	200
Объединение неразделимо!	200
Сравнение строк	201
Сравнение без учета регистра	205
Использование конструкции <code>switch</code>	205
Считывание ввода пользователя	206
Разбор числового ввода	207
Обработка последовательности чисел	210
Управление выводом программы	212
Использование методов <code>Trim()</code> и <code>Pad()</code>	212
Использование функции конкатенации	215
Использование функции <code>Split()</code>	217
Форматирование строки	218
Часть IV. Объектно-ориентированное программирование	223
Глава 10. Что такое объектно-ориентированное программирование	225
Объектно-ориентированная концепция №1 — абстракция	225
Приготовление блюд с помощью функций	226
Приготовление “объектно-ориентированных” блюд	226
Объектно-ориентированная концепция №2 — классификация	227
Зачем нужна классификация	227
Объектно-ориентированная концепция №3 — удобный интерфейс	228
Объектно-ориентированная концепция №4 — управление доступом	229
Поддержка объектно-ориентированных концепций в C#	229
Глава 11. Классы	231
Ограничение доступа к членам класса	231
Пример программы с использованием открытых членов	232
Прочие уровни безопасности	235
Зачем нужно управление доступом	235
Методы доступа	236
Пример управления доступом	237
Выводы	242
Определение свойств класса	242
Конструирование объектов посредством конструкторов	244
Конструкторы, предоставляемые C#	244

Конструктор по умолчанию	246
Создание объектов	247
Выполнение конструктора в отладчике	249
Непосредственная инициализация объекта — конструктор по умолчанию	252
Конструирование с инициализаторами	252
Перегрузка конструкторов	253
Устранение дублирования конструкторов	256
Фокусы с объектами	260
Глава 12. Наследование	261
Наследование класса	261
Зачем нужно наследование	263
Более сложный пример наследования	264
ЯВЛЯЕТСЯ или СОДЕРЖИТ	267
Отношение ЯВЛЯЕТСЯ	267
Доступ к BankAccount через содержание	268
Отношение СОДЕРЖИТ	269
Когда использовать отношение ЯВЛЯЕТСЯ, а когда — СОДЕРЖИТ	270
Поддержка наследования в C#	270
Изменение класса	270
Неверное преобразование времени выполнения	271
Ключевые слова is и as	272
Наследование и конструктор	274
Вызов конструктора по умолчанию базового класса	274
Передача аргументов конструктору базового класса	276
Обновленный класс BankAccount	278
Деструктор	281
Глава 13. Полиморфизм	283
Перегрузка унаследованного метода	283
Простейший случай перегрузки функции	284
Различные классы, различные методы	284
Соккрытие метода базового класса	285
Вызов методов базового класса	289
Полиморфизм	291
Что неверно в стратегии использования объявленного типа	292
Использование is для полиморфного доступа к скрытому методу	293
Объявление метода виртуальным	294
Абстракционизм в C#	297
Разложение классов	297
Голая концепция, выражаемая абстрактным классом	302
Как использовать абстрактные классы	302
Создание абстрактных объектов невозможно	304
Создание иерархии классов	304
Опечатывание класса	308

Часть V. За базовыми классами	309
Глава 14. Интерфейсы и структуры	311
Что значит МОЖЕТ_ИСПОЛЬЗОВАТЬСЯ_КАК	311
Что такое интерфейс	312
Краткий пример	313
Пример программы, использующей отношение МОЖЕТ_ИСПОЛЬЗОВАТЬСЯ_КАК	315
Создание собственного интерфейса	315
Предопределенные интерфейсы	316
Сборка воедино	317
Наследование интерфейса	323
Абстрактный интерфейс	323
Структуры C# и их отличия от классов	326
Структуры C#	327
Конструктор структуры	329
Методы структур	329
Пример применения структуры	330
Унификация системы типов	333
Предопределенные типы структур	333
Унификация системы типов с помощью структур	334
Упаковка типов-значений	337
Глава 15. Обобщенное программирование	339
Необобщенные коллекции	340
Необобщенные коллекции	340
Использование необобщенных коллекций	341
Обобщенные классы	343
Обобщенные классы безопасны	343
Обобщенные классы эффективны	344
Использование обобщенных коллекций	344
Понятие <T>	345
Использование List<T>	345
Создание собственного обобщенного класса	347
Очередь с приоритетами	348
Распаковка пакета	352
Функция Main()	353
Написание обобщенного кода	355
Обобщенная очередь с приоритетами	356
Незавершенные дела	358
Обобщенные методы	360
Обобщенные методы в необобщенных классах	362
Обобщенные методы в обобщенных классах	363
Ограничения для обобщенного метода	363
Обобщенные интерфейсы	364
Обобщенные и необобщенные интерфейсы	364
Использование (необобщенной) фабрики классов	365
Построение обобщенной фабрики	366

Часть VI. Великолепные десятки	373
Глава 16. Десять наиболее распространенных ошибок компиляции	375
The name 'memberName' does not exist in the class or namespace 'className'	375
Cannot implicitly convert type 'x' into 'y'	377
'className.memberName' is inaccessible due to its protection level	379
Use of unassigned local variable 'n'	380
Unable to copy the file 'programName.exe' to 'programName.exe'. The process cannot...	380
'subclassName.methodName' hides inherited member 'baseclassName.methodName'.	
Use the new keyword if hiding was intended	381
'subclassName' : cannot inherit from sealed class 'baseclassName'	382
'className' does not implement interface member 'methodName'	383
'methodName' : not all code paths return a value	383
} expected	384
Глава 17. Десять основных отличий C# и C++	385
Отсутствие глобальных данных и функций	386
Все объекты размещаются вне кучи	386
Переменные-указатели запрещены	387
Обобщенные классы C# и шаблоны C++	387
Никаких включаемых файлов	388
Не конструирование, а инициализация	388
Корректное определение типов переменных	389
Нет множественного наследования	389
Проектирование хороших интерфейсов	389
Унифицированная система типов	389
Часть VII. Дополнительные главы	391
Глава 18. Эти исключительные исключения	393
Старый способ обработки ошибок	393
Возврат индикатора ошибки	395
Чем плохи коды ошибок	398
Использование механизма исключений для сообщения об ошибках	400
Пример	402
Создание собственного класса исключения	405
Использование нескольких catch-блоков	406
Как исключения протекают сквозь пальцы	408
Регенерация исключения	411
Как реагировать на исключения	412
Перекрытие класса Exception	413
Глава 19. Работа с файлами и библиотеками	419
Разделение одной программы на несколько исходных файлов	419
Разделение единой программы на сборки	421
Объединение исходных файлов в пространства имен	422

Объявление пространств имен	422
Важность пространств имен	424
Доступ к классам с использованием полностью квалифицированных имен	425
Директива using	426
Использование полностью квалифицированных имен	427
Объединение классов в библиотеки	430
Создание проекта библиотеки классов	430
Создание классов для библиотеки	431
Создание проекта драйвера	432
Хранение данных в файлах	434
Использование StreamWriter	435
Повышение скорости чтения с использованием StreamReader	440
Глава 20. Работа с коллекциями	445
Обход каталога файлов	445
Написание собственного класса коллекции: связанный список	451
Пример связанного списка	452
Зачем нужен связанный список	461
Обход коллекций: итераторы	461
Доступ к коллекции: общая задача	462
Использование foreach	464
Обращение к коллекциям как к массивам: индексаторы	465
Формат индексатора	465
Пример программы с использованием индексатора	465
Блок итератора	469
Итерация месяцев	473
Что такое коллекция	474
Синтаксис итератора	475
Блоки итераторов произвольного вида и размера	476
Где надо размещать итераторы	479
Глава 21. Использование интерфейса Visual Studio	487
Настройка расположения окон	487
Состояния окон	488
Скрытие окна	490
Перестановка окон	490
Наложение окон	491
Модные штучки	493
Работа с Solution Explorer	493
Упрощение жизни с помощью проектов и решений	494
Отображение проекта	495
Добавление класса	497
Завершение демонстрационной программы	498
Преобразование классов в программу	501
Как должен выглядеть код	502
Помогите мне!	506
F1	506

Предметный указатель	507
Поиск	509
Дополнительные возможности	510
Автоперечисление членов	511
Отладка	512
Жучки в программе: а дуством не пробовали?	512
Пошаговая отладка	514
Главное — вовремя остановиться	517
Стек вызовов	520
Я сделал это!	523
Глава 22. C# по дешевке	525
Работа без сети — но не без платформы .NET	526
Получение бесплатных компонентов	526
Обзор цикла разработки	527
Программирование на C# в программе SharpDevelop	528
Изучение SharpDevelop	528
Сравнение возможностей SharpDevelop и Visual Studio	529
Получение справочной информации	530
Настройка программы SharpDevelop	531
Добавление инструмента для запуска отладчика	531
Запуск отладчика из SharpDevelop	532
Отсутствующие возможности отладчика	534
Программирование на C# в TextPad	534
Создание класса документов .CS для языка C#	537
Добавление собственных инструментов: Build C# Debug	538
Настройка инструмента для компиляции финальной версии	540
Объяснение опций настройки инструментов Debug и Release	541
Работа над ошибками компиляции	545
Настройка остальных инструментов	545
Тестирование с помощью программы NUnit	548
Запуск программы NUnit	548
Тестирование	549
Написание тестов NUnit	550
Исправление ошибок в проверяемой программе	557
Написание исходного текста Windows Forms без Form Designer	559
Это всего лишь код	559
Работа в стиле визуального инструмента	560
Частичные классы	561
Самостоятельное написание	562
Убедитесь, что пользователи смогут запустить вашу программу	563
Visual Studio для бедных	564
Предметный указатель	565