Содержание

Об авторе 14
О техническом редакторе 14

Благодарности 15

Введение 16

Как организована эта книга 17
От издательства 20

Глава 1. Обзор C# 21

Отличия между C# и C++ 21
- Язык C# 21
- Язык C++ 22
- Сборка мусора CLR 23

Пример программы на C# 23
Обзор средств, добавленных в C# 2.0 25
Обзор средств, добавленных в C# 3.0 26
Обзор новых средств C# 4.0 27
Резюме 28

Глава 2. C# и CLR 29

JIT-компилятор и CLR 29
Сборки и загрузчик сборок 31
- Минимизация рабочего набора приложения 32
- Назначение сборкам имен 33
- Загрузка сборок 33

Метаданные 34
Совместимость между языками 35
Резюме 36

Глава 3. Обзор синтаксиса C# 37

C# — строго типизированный язык 37
Выражения 38
Операторы и выражения 39
Типы и переменные 40
- Типы значений 42
- Ссылочные типы 45
- Инициализация переменных по умолчанию 45
- Неявно типизированные локальные переменные 46
- Преобразования типов 48
- Операции `as` и `is` 50
- Обобщения 51

Пространства имен 53
- Определение пространств имен 54
- Использование пространств имен 55

Поток управления 56
- `if-else`, `while`, `do-while` и `for` 56
- `switch` 56

foreach 57
break, continue, goto, return и throw 58
Резюме 58

Глава 4. Классы, структуры и объекты 59

Определения классов 61
Поля 62
Конструкторы 65
Методы 66
Свойства 67
Инкапсуляция 72
Доступность 76
Интерфейсы 77
Наследование 79
Герметизированные классы 85
Абстрактные классы 86
Вложенные классы 87
Индексаторы 90
Частичные классы 92
Частичные методы 93
Статические классы 94
Зарезервированные имена членов 96
Определения типов значений 97
Смысл ключевого слова this 100
Финализаторы 102
Интерфейсы 102
Анонимные типы 103
Инициализаторы объектов 105
Упаковка и распаковка 108
Когда происходит упаковка 112
Эффективность и путаница 113
Класс System.Object 114
Эквивалентность и ее смысл 116
Интерфейс IComparable 116
Создание объектов 116
Ключевое слово new 116
Инициализация полей 118
Статические конструкторы (класса) 119
Конструктор экземпляра и порядок создания 122
Уничтожение объектов 125
Финализаторы 126
Детерминированное уничтожение 127
Обработка исключений 128
Одноразовые объекты 128
Интерфейс IDisposable 129
Ключевое слово using 131
Типы параметров методов 132
Аргументы-значения 132
Аргументы ref 133
Параметры out 134
Массивы params 135

Перегрузка методов 136
Необязательные аргументы 136
Именованные аргументы 137
Наследование и виртуальные методы 140
Виртуальные и абстрактные методы 140
Методы `new` и `override` 141
Методы `sealed` 143
Завершающие замечания о виртуальных методах C# 144
Наследование, включение и делегирование 144
Выбор между интерфейсом и наследованием класса 144
Сравнение делегирования и композиции с наследованием 146
Резюме 148

Глава 5. Интерфейсы и контракты 149

Интерфейсы определяют типы 150
Определение интерфейсов 151
Что может быть интерфейсом? 152
Наследование интерфейсов и сокрытие членов 152
Реализация интерфейсов 154
Неявная реализация интерфейса 155
Явная реализация интерфейса 155
Переопределение реализаций интерфейсов в производных классах 157
Остерегайтесь побочных эффектов от реализации интерфейсов типами значений 160
Правила сопоставления членов интерфейсов 161
Явная реализация интерфейса с помощью типа значений 164
Соображения, касающиеся версий 166
Контракты 167
Контракты, реализованные классами 167
Контракты интерфейса 169
Выбор между интерфейсами и классами 170
Резюме 173

Глава 6. Перегрузка операций 175

Можете — не значит должны 175
Типы и форматы перегруженных операций 175
Операции не должны изменять свои операнды 177
Имеет ли значение порядок параметров? 177
Перегрузка операции сложения 178
Операции, допускающие перегрузку 179
Операции сравнения 179
Операции преобразования 182
Булевские операции 185
Резюме 187

Глава 7. Безопасность и обработка исключений 189

Как CLR трактует исключения 189
Механизм обработки исключений в C# 190
Генерация исключений 190
Изменения, касающиеся необработанных исключений, которые появились в .NET 2.0 190
Обзор синтаксиса операторов `try`, `catch` и `finally` 192
Повторная генерация и трансляция исключений 194

Исключения, сгенерированные в блоке finally 196
Исключения, сгенерированные в финализаторах 197
Исключения, сгенерированные в статических конструкторах 198
Кто должен обрабатывать исключения? 199
Избегайте применения исключений для управления потоком выполнения 200
Обеспечение нейтральности к исключениям 200
Базовая структура нейтрального к исключениям кода 201
Ограниченные области выполнения 206
Критичные финализаторы и SafeHandle 208
Создание пользовательских классов исключений 212
Работа с выделенными ресурсами и исключениями 214
Обеспечение поведения отката 218
Резюме 221

Глава 8. Работа со строками 223

Обзор String 223
Строковые литералы 224
Спецификаторы формата и глобализация 225
Object.ToString, IFormattable и CultureInfo 226
Создание и регистрация пользовательских типов CultureInfo 227
Форматные строки 229
Console.WriteLine и String.Format 230
Примеры строкового форматирования в пользовательских типах 231
ICustomFormatter 232
Сравнение строк 234
Работа со строками из внешних источников 236
StringBuilder 238
Поиск строк с помощью регулярных выражений 239
Поиск с помощью регулярных выражений 240
Поиск и группирование 241
Замена текста с помощью Regex 245
Варианты создания Regex 247
Резюме 248

Глава 9. Массивы, типы коллекций и итераторы 249

Представление массивов 249
Неявно типизированные массивы 251
Возможность преобразования и ковариантность 253
Возможности сортировки и поиска 254
Синхронизация 254
Сравнение векторов и массивов 255
Многомерные прямоугольные массивы 256
Многомерные зубчатые массивы 258
Типы коллекций 260
Сравнение ICollection<T> и ICollection 260
Синхронизация коллекций 261
Списки 262
Словари 263
Наборы 264
System.Collections.ObjectModel 264
Эффективность 267

IEnumerable<T>, IEnumerator<T>, IEnumerable и IEnumerator 268
Типы, производящие коллекции 272
Итераторы 272
Прямые, обратные и двунаправленные итераторы 277
Инициализаторы коллекций 281
Резюме 282

Глава 10. Делегаты, анонимные функции и события 283

Обзор делегатов 283
Создание и использование делегатов 284
Одиночный делегат 285
Цепочки делегатов 286
Итерация по цепочкам делегатов 288
Несвязанные делегаты (делегаты открытого экземпляра) 289
События 292
Анонимные методы 296
Захваченные переменные и замыкания 298
Остерегайтесь сюрпризов, связанных с захваченными переменными 300
Анонимные методы как привязки параметров делегатов 303
Шаблон Strategy 306
Резюме 307

Глава 11. Обобщения 309

Разница между обобщениями и шаблонами C++ 310
Эффективность и безопасность типов обобщений 311
Определения обобщенных типов и конструируемые типы 313
Обобщенные классы и структуры 313
Обобщенные интерфейсы 316
Обобщенные методы 317
Обобщенные делегаты 319
Преобразование обобщенного типа 322
Выражение значения по умолчанию 322
Типы, допускающие значения null 324
Контроль доступа к конструируемым типам 325
Обобщения и наследование 326
Ограничения 327
Ограничения на неклассовых типах 332
Ковариантность и контравариантность 332
Ковариантность 334
Контравариантность 337
Инвариантность 338
Вариантность и делегаты 339
Обобщенные системные коллекции 343
Обобщенные системные интерфейсы 344
Проблемы выбора и их решение 346
Преобразования и операции внутри обобщенных типов 346
Динамическое создание конструируемых типов 354
Резюме 356

Глава 12. Многопоточность в C# 357

Многопоточность в C# и .NET 357

Запуск потоков 358
Передача данных новым потокам 360
Использование ParameterizedThreadStart 361
Шаблон IOU и асинхронные вызовы методов 362
Состояния потока 362
Завершение потоков 364
Останавливающиеся и пробуждающиеся потоки 366
Ожидание завершения потока 368
Потоки переднего плана и фоновые потоки 368
Локальное хранилище потока 369
Как неуправляемые потоки и апартаменты COM приспособлены друг к другу 372
Синхронизация работы между потоками 373
Легковесная синхронизация с помощью класса Interlocked 375
Класс SpinLock 381
Класс Monitor 382
Блокирующие объекты 391
Семафоры 396
События 398
Объекты синхронизации Win32 и WaitHandle 399
Использование ThreadPool 402
Асинхронные вызовы методов 403
Таймеры 410
Параллельное программирование 411
Класс Task 412
Класс Parallel 413
Простой вход в пул потоков 417
Классы коллекций, безопасные в отношении потоков 418
Резюме 419

Глава 13. В поисках канонических форм C# 421

Канонические формы ссылочных типов 422
Классы должны помечаться как sealed по умолчанию 422
Использование шаблона NVI 423
Является ли Object клонируемым? 426
Является ли Object одноразовым? 431
Нужен ли финализатор для Object? 434
Что означает эквивалентность для данного объекта? 441
Ссылочные типы и эквивалентность идентичности 441
Эквивалентность значений 444
Переопределение Object.Equals для ссылочных типов 445
Если переопределен метод Equals, то необходимо переопределить и GetHashCode 448
Поддерживает ли объект упорядочивание? 451
Является ли Object форматируемым? 453
Является ли Object преобразуемым? 457
Всегда отдавайте предпочтение безопасности типов 459
Использование неизменяемых ссылочных типов 463
Канонические формы типов значений 465
Переопределение Equals для повышения производительности 466
Поддерживают ли значения этого типа какие-то интерфейсы? 470
Реализация безопасных к типам версий для членов интерфейса и унаследованных методов 471

Резюме 473
Список вопросов для ссылочных типов 473
Список вопросов для типов значений 475

Глава 14. Расширяющие методы 477

Введение в расширяющие методы 477
Как компилятор находит расширяющие методы? 478
Что происходит “за кулисами” 481
Читабельность или понятность кода 481
Рекомендации по использованию 483
Использование расширяющих методов вместо наследования 483
Изоляция расширяющих методов в отдельном пространстве имен 484
Изменение контракта типа может нарушить работу расширяющих методов 485
Трансформации 485
Цепочки операций 489
Пользовательские итераторы 491
Заимствование из функционального программирования 492
Шаблон Visitor 498
Резюме 501

Глава 15. Лямбда-выражения 503

Введение в лямбда-выражения 503
Лямбда-выражения и замыкания 504
Замыкания в C# 1.0 507
Замыкания в C# 2.0 508
Лямбда-операторы 509
Деревья выражений 509
Операции над выражениями 512
Функции как данные 513
Полезные применения лямбда-выражений 513
Вернемся к итераторам и генераторам 514
Замыкание (захват переменной) и мемоизация 517
Каррирование 521
Анонимная рекурсия 523
Резюме 524

Глава 16. LINQ: язык интегрированных запросов 525

Мост к данным 526
Выражения запросов 526
Вернемся к расширяющим методам и лямбда-выражениям 528
Стандартные операции запросов 529
Ключевые слова запросов C# 530
Конструкция `from` и переменные диапазона 530
Конструкция `join` 532
Конструкция `where` и фильтры 534
Конструкция `orderby` 534
Конструкция `select` и проекция 535
Конструкция `let` 537
Конструкция `group` 538
Конструкция `into` и продолжение 541

Преимущества лени 542
Поощрение лени итераторами C# 542
Ниспровержение лени 543
Немедленное выполнение запросов 545
Еще раз о деревьях выражений 545
Приемы функционального программирования 546
Пользовательские стандартные операции запросов и “ленивое” вычисление 546
Замена операторов `foreach` 553
Резюме 555

Глава 17. Динамические типы 557

Что означает динамический? 557
Как работает тип `dynamic`? 559
Замечательная унификация 561
Места вызовов 562
Объекты со специальным динамическим поведением 564
Эффективность 566
Упаковка с помощью типа `dynamic` 567
Динамические преобразования 568
Неявные преобразования динамических выражений 569
Динамическое разрешение перегрузки 570
Динамическое наследование 572
Нельзя наследовать от `dynamic` 572
Нельзя реализовывать интерфейсы `dynamic` 573
Возможность наследования от динамических базовых типов 575
Утиная типизация в C# 576
Ограничения динамических типов 579
Динамическое создание объектов с помощью `ExpandoObject` 579
Резюме 583

Предметный указатель 584