

# Содержание

Об авторе	21
Посвящение	22
Благодарности	22
<b>Введение</b>	<b>23</b>
Об этой книге	23
Глупые предположения	24
Пиктограммы, используемые в книге	25
Источники дополнительной информации	25
Что дальше	26
Ждем ваших отзывов!	27
<b>Часть 1. Основы программирования на C#</b>	<b>29</b>
<b>Глава 1. Ваше первое консольное приложение на C#</b>	<b>31</b>
Компьютерные языки, C# и .NET	31
Что такое программа	32
Что такое C#	32
Что такое .NET	33
Что такое Visual Studio 2017 и Visual C#	34
Создание первого консольного приложения	35
Создание исходной программы	36
Тестовая поездка	40
Заставим программу работать	41
Обзор консольного приложения	43
Каркас программы	43
Комментарии	43
Тело программы	44
Введение в хитрости панели элементов	45
Сохранение кода на панели элементов	45
Повторное использование кода из панели элементов	46
<b>Глава 2. Работа с переменными</b>	<b>47</b>
Объявление переменной	48
Что такое int	48

Правила объявления переменных	49
Вариации на тему <code>int</code>	50
Представление дробных чисел	51
Работа с числами с плавающей точкой	52
Объявление переменной с плавающей точкой	53
Ограничения переменных с плавающей точкой	54
Десятичные числа: комбинация целых чисел и чисел с плавающей точкой	56
Объявление переменных типа <code>decimal</code>	56
Сравнение десятичных и целых чисел, а также чисел с плавающей точкой	56
Логичен ли логический тип	57
Символьные типы	57
Тип <code>char</code>	58
Специальные символы	58
Тип <code>string</code>	59
Что такое тип-значение	60
Сравнение <code>string</code> и <code>char</code>	61
Вычисление високосных лет: <code>DateTime</code>	62
Объявление числовых констант	64
Преобразование типов	65
Позвольте компилятору <code>C#</code> вывести типы данных	66
<b>Глава 3. Работа со строками</b>	69
Неизменяемость строк	70
Основные операции над строками	72
Сравнение строк	72
Проверка равенства: метод <code>Compare()</code>	73
Сравнение без учета регистра	76
Изменение регистра	76
Отличие строк в разных регистрах	77
Преобразование символов строки в символы верхнего или нижнего регистра	77
Цикл по строке	78
Поиск в строках	79
Как искать	79
Пуста ли строка	80

Получение введенной пользователем информации	80
Удаление пробельных символов	80
Анализ числового ввода	81
Обработка последовательности чисел	84
Объединение массива строк в одну строку	86
Управление выводом программы	86
Использование методов Trim() и Pad()	86
Использование метода Concat()	89
Использование метода Split()	91
Форматирование строк	92
StringBuilder: эффективная работа со строками	97
<b>Глава 4. Операторы</b>	<b>99</b>
Арифметика	99
Простейшие операторы	100
Порядок выполнения операторов	100
Оператор присваивания	102
Оператор инкремента	102
Логично ли логическое сравнение	103
Сравнение чисел с плавающей точкой	104
Составные логические операторы	105
Тип выражения	107
Вычисление типа операции	108
Типы при присваивании	110
Перегрузка операторов	110
<b>Глава 5. Управление потоком выполнения</b>	<b>113</b>
Ветвление с использованием if и switch	114
Инструкция if	115
Инструкция else	118
Как избежать else	119
Вложенные инструкции if	120
Конструкция switch	123
Циклы	125
Цикл while	125
Цикл do...while	130
Операторы break и continue	130

Цикл без счетчика	131
Правила области видимости	135
Цикл <code>for</code>	136
Пример	136
Зачем нужны разные циклы	137
Вложенные циклы	138
Оператор <code>goto</code>	139
<b>Глава 6. Глава для коллекционеров</b>	<b>141</b>
<b>Массивы C#</b>	<b>142</b>
Зачем нужны массивы	142
Массив фиксированного размера	143
Массив переменного размера	145
Свойство <code>Length</code>	148
Инициализация массивов	148
Цикл <code>foreach</code>	148
Сортировка массива данных	150
Использование <code>var</code> для массивов	154
Коллекции C#	155
Синтаксис коллекций	156
Понятие <code>&lt;T&gt;</code>	157
Обобщенные коллекции	157
Использование списков	157
Инстанцирование пустого списка	158
Создание списка целых чисел	158
Создание списка для хранения объектов	159
Преобразования списков в массивы и обратно	159
Подсчет количества элементов в списке	159
Поиск в списках	160
Прочие действия со списками	160
Использование словарей	160
Создание словаря	161
Поиск в словаре	161
Итерирование словаря	161
Инициализаторы массивов и коллекций	163
Инициализация массивов	163
Инициализация коллекций	163

Использование множеств	164
Выполнение специфичных для множеств задач	164
Создание множества	165
Добавление элемента в множество	165
Выполнение объединения	166
Пересечение множеств	167
Получение разности	168
Не используйте старые коллекции	169
<b>Глава 7. Работа с коллекциями</b>	<b>171</b>
Обход каталога файлов	171
Использование программы LoopThroughFiles	172
Начало программы	173
Получение начальных входных данных	173
Создание списка файлов	174
Форматирование вывода	175
Вывод в шестнадцатеричном формате	177
Обход коллекций: итераторы	178
Доступ к коллекции: общая задача	179
Использование foreach	181
Обращение к коллекциям как к массивам: индексаторы	182
Формат индексатора	183
Пример программы с использованием индексатора	183
Блок итератора	187
Создание каркаса блока итератора	188
Итерирование дней в месяцах	189
Что же такое коллекция	191
Синтаксис итератора	192
Блоки итераторов произвольного вида и размера	194
<b>Глава 8. Обобщенность</b>	<b>199</b>
Обобщенность в C#	200
Обобщенные классы безопасны	200
Обобщенные классы эффективны	201
Создание собственного обобщенного класса	202
Очередь посылок	203
Очередь с приоритетами	203
Распаковка пакета	208

Метод <code>Main()</code>	210
Написание обобщенного кода	211
И наконец — обобщенная очередь с приоритетами	212
Использование простого необобщенного класса фабрики	215
Незавершенные дела	217
Пересмотр обобщенности	220
Вариантность	221
Контравариантность	221
Ковариантность	223
<b>Глава 9. Эти исключительные исключения</b>	<b>225</b>
Использование механизма исключений для сообщения об ошибках	226
О <code>try</code> -блоках	227
О <code>catch</code> -блоках	228
О <code>finally</code> -блоках	228
Что происходит при генерации исключения	229
Генерация исключений	231
Для чего нужны исключения	232
Исключительный пример	232
Что делает этот пример “исключительным”	234
Трассировка стека	235
Использование нескольких <code>catch</code> -блоков	235
Планирование стратегии обработки ошибок	238
Вопросы, помогающие при планировании	238
Советы по написанию кода с хорошей обработкой ошибок	239
Анализ возможных исключений метода	241
Как выяснить, какие исключения генерируются теми или иными методами	243
Последний шанс перехвата исключения	244
Генерирующие исключения выражения	245
<b>Глава 10. Списки элементов с использованием перечислений</b>	<b>247</b>
Перечисления в реальном мире	248
Работа с перечислениями	249
Использование ключевого слова <code>enum</code>	250
Создание перечислений с инициализаторами	251
Указание типа данных перечисления	252
Создание флагов-перечислений	252
Применение перечислений в конструкции <code>switch</code>	254

<b>Часть 2. Объектно-ориентированное программирование на C#</b>	257
<b>Глава 11. Что такое объектно-ориентированное программирование</b>	259
Объектно-ориентированная концепция № 1: абстракция	260
Процедурные поездки	261
Объектно-ориентированные поездки	261
Объектно-ориентированная концепция № 2: классификация	262
Зачем нужна классификация	263
Объектно-ориентированная концепция № 3: удобные интерфейсы	264
Объектно-ориентированная концепция № 3: управление доступом	265
Поддержка объектно-ориентированных концепций в C#	266
<b>Глава 12. Немного о классах</b>	267
Определение класса и объекта	268
Определение класса	268
Что такое объект	269
Доступ к членам объекта	270
Пример объектно-основанной программы	271
Различие между объектами	273
Работа со ссылками	273
Классы, содержащие классы	275
Статические члены класса	277
Определение константных членов-данных и членов-данных только для чтения	278
<b>Глава 13. Методы</b>	281
Определение и использование метода	282
Использование методов в ваших программах	283
Аргументы метода	291
Передача аргументов методу	291
Передача методу нескольких аргументов	292
Соответствие определений аргументов их использованию	293
Перегрузка методов	294
Реализация аргументов по умолчанию	296
Возврат значений из метода	299
Возврат значения оператором <code>return</code>	300
Определение метода без возвращаемого значения	301
Возврат нескольких значений с использованием кортежей	303

Кортеж с двумя элементами	303
Применение метода <code>Create()</code>	304
Многоэлементные кортежи	304
Создание кортежей более чем с двумя элементами	306
<b>Глава 14. Поговорим об этом</b>	<b>307</b>
Передача объекта в метод	307
Определение методов	309
Определение статического метода	309
Определение метода экземпляра	311
Полное имя метода	313
Обращение к текущему объекту	314
Ключевое слово <code>this</code>	315
Когда <code>this</code> используется явно	316
Что делать при отсутствии <code>this</code>	319
Использование локальных функций	321
<b>Глава 15. Класс: каждый сам за себя</b>	<b>323</b>
Ограничение доступа к членам класса	324
Пример программы с использованием открытых членов	324
Прочие уровни безопасности	327
Зачем нужно управление доступом	328
Методы доступа	329
Пример управления доступом	330
Выводы	334
Определение свойств класса	334
Статические свойства	335
Побочные действия свойств	336
Дайте компилятору написать свойства для вас	337
Методы и уровни доступа	337
Конструирование объектов с помощью конструкторов	338
Конструкторы, предоставляемые <code>C#</code>	338
Замена конструктора по умолчанию	340
Конструирование объектов	341
Непосредственная инициализация объекта	343
Конструирование с инициализаторами	344
Инициализация объекта без конструктора	345

Применение членов с кодом	346
Создание методов с кодом	346
Определение свойств с кодом	347
Определение конструкторов и деструкторов с кодом	347
Определение методов доступа к свойствам с кодом	347
Определение методов доступа к событиям с кодом	348
<b>Глава 16. Наследование</b>	<b>349</b>
Наследование класса	350
Зачем нужно наследование	352
Более сложный пример наследования	353
<b>ЯВЛЯЕТСЯ</b> или <b>СОДЕРЖИТ</b>	356
Отношение <b>ЯВЛЯЕТСЯ</b>	356
Доступ к <code>BankAccount</code> через содержание	357
Отношение <b>СОДЕРЖИТ</b>	358
Когда использовать отношение <b>ЯВЛЯЕТСЯ</b> и когда — <b>СОДЕРЖИТ</b>	359
Поддержка наследования в <code>C#</code>	360
Заменяемость классов	360
Неверное преобразование времени выполнения	361
Избегание неверных преобразований с помощью оператора <code>is</code>	362
Избегание неверных преобразований с помощью оператора <code>as</code>	363
Класс <code>object</code>	363
Наследование и конструктор	365
Вызов конструктора по умолчанию базового класса	365
Передача аргументов конструктору базового класса	366
Указание конкретного конструктора базового класса	368
Обновленный класс <code>BankAccount</code>	369
<b>Глава 17. Полиморфизм</b>	<b>375</b>
Перегрузка унаследованного метода	376
Простейший случай перегрузки метода	376
Различные классы, различные методы	377
Скрытие метода базового класса	377
Вызов методов базового класса	382
Полиморфизм	384
Что неверно в стратегии использования объявленного типа	385
Использование <code>is</code> для полиморфного доступа к скрытому методу	387

Объявление метода виртуальным и перекрытие	388
Получение максимальной выгоды от полиморфизма	391
Визитная карточка класса: метод ToString()	391
Абстракционизм в C#	392
Разложение классов	392
Абстрактный класс: ничего, кроме идеи	397
Как использовать абстрактные классы	398
Создание абстрактных объектов невозможно	400
Опечатывание класса	400
<b>Глава 18. Интерфейсы</b>	<b>403</b>
Что значит МОЖЕТ_ИСПОЛЬЗОВАТЬСЯ_КАК	403
Что такое интерфейс	405
Реализация интерфейса	406
Именованые интерфейсы	407
Зачем C# включает интерфейсы	407
Наследование и реализация интерфейса	407
Преимущества интерфейсов	408
Использование интерфейсов	409
Тип, возвращаемый методом	409
Базовый тип массива или коллекции	410
Более общий тип ссылки	410
Использование предопределенных типов интерфейсов C#	411
Пример программы, использующей отношение МОЖЕТ_ИСПОЛЬЗОВАТЬСЯ_КАК	411
Создание собственного интерфейса	411
Реализация интерфейса IComparable<T>	413
Сборка воедино	414
Вернемся к Main()	418
Унификация иерархий классов	419
Что скрыто за интерфейсом	421
Наследование интерфейсов	424
Использование интерфейсов для внесения изменений в объектно-ориентированные программы	425
Гибкие зависимости через интерфейсы	426
Абстрактный или конкретный? Когда следует использовать абстрактный класс, а когда — интерфейс	426
Реализация отношения СОДЕРЖИТ с помощью интерфейсов	427

<b>Глава 19. Делегирование событий</b>	429
Звонок домой: проблема обратного вызова	430
Определение делегата	431
Пример передачи кода	433
Делегирование задания	433
Очень простой первый пример	433
Более реальный пример	435
Обзор большего примера	435
Создание приложения	436
Знакомимся с кодом	439
Жизненный цикл делегата	441
Анонимные методы	443
События C#	444
Проектный шаблон Observer	445
Что такое событие. Публикация и подписка	445
Как издатель оповещает о своих событиях	446
Как подписаться на событие	447
Как опубликовать событие	447
Как передать обработчику события дополнительную информацию	449
Рекомендованный способ генерации событий	449
Как наблюдатели “обрабатывают” событие	450
<b>Глава 20. Пространства имен и библиотеки</b>	453
Разделение одной программы на несколько исходных файлов	454
Разделение единой программы на сборки	455
Выполнимый файл или библиотека	455
Сборки	456
Выполнимые файлы	457
Библиотеки классов	458
Объединение классов в библиотеки	458
Создание проекта библиотеки классов	458
Создание автономной библиотеки классов	459
Добавление второго проекта к существующему решению	460
Создание классов для библиотеки	462
Использование тестового приложения	463
Дополнительные ключевые слова для управления доступом	464
internal: строим глазки ЦРУ	465

protected: поделимся с подклассами	467
protected internal: более изошренная защита	469
Размещение классов в пространствах имен	470
Объявление пространств имен	472
Пространства имен и доступ	473
Использование полностью квалифицированных имен	475
<b>Глава 21. Именованные и необязательные параметры</b>	<b>477</b>
Изучение необязательных параметров	478
Ссылочные типы	480
Выходные параметры	481
Именованные параметры	482
Разрешение перегрузки	483
Альтернативные методы возврата значений	483
Работа с переменными out	484
Возврат значений по ссылке	485
<b>Глава 22. Структуры</b>	<b>487</b>
Сравнение структур и классов	488
Ограничения структур	488
Различия типов-значений	489
Когда следует использовать структуры	489
Создание структур	490
Определение базовой структуры	490
Добавление распространенных элементов структур	491
Использование структур как записей	497
Управление отдельной записью	498
Добавление структур в массивы	498
Перекрытие методов	499
<b>Часть 3. Вопросы проектирования на C#</b>	<b>501</b>
<b>Глава 23. Написание безопасного кода</b>	<b>503</b>
Проектирование безопасного программного обеспечения	504
Определение того, что следует защищать	505
Документирование компонентов программы	505
Разложение компонентов на функции	505
Обнаружение потенциальных угроз в функциях	506
Оценка рисков	507

Построение безопасных приложений Windows	507
Аутентификация с использованием входа в Windows	508
Шифрование информации	511
Безопасность развертывания	511
Построение безопасных приложений Web Forms	512
Атаки SQL Injection	513
Уязвимости сценариев	514
Наилучшие методы защиты приложений Web Forms	515
Использование System.Security	517
<b>Глава 24. Обращение к данным</b>	<b>519</b>
Знакомство с System.Data	520
Классы данных и каркас	522
Получение данных	523
Использование пространства имен System.Data	524
Настройка образца схемы базы данных	524
Подключение к источнику данных	525
Работа с визуальными инструментами	531
Написание кода для работы с данными	532
Использование Entity Framework	536
<b>Глава 25. Рыбалка в потоке</b>	<b>541</b>
Где водится рыба: файловые потоки	541
Потоки	542
Читатели и писатели	542
Использование StreamWriter	544
Пример использования потока	545
Как это работает	547
Наконец-то мы пишем!	551
Использование конструкции using	552
Использование StreamReader	556
Еще о читателях и писателях	560
Другие виды потоков	562
<b>Глава 26. Доступ к Интернету</b>	<b>563</b>
Знакомство с System.Net	564
Как сетевые классы вписываются в каркас	565
Использование пространства имен System.Net	567

Проверка состояния сети	567
Загрузка файла из Интернета	569
Отчет по электронной почте	572
Регистрация сетевой активности	574
<b>Глава 27. Создание изображений</b>	<b>579</b>
Знакомство с System.Drawing	580
Графика	580
Перья	581
Кисти	581
Текст	582
Классы рисования и каркас .NET	583
Использование пространства имен System.Drawing	584
Приступая к работе	584
Настройка проекта	585
Обработка счета	586
Создание подключения к событию	587
Рисование доски	588
Запуск новой игры	589
<b>Предметный указатель</b>	<b>591</b>