

**КАК НАУЧИТЬСЯ МЫСЛИТЬ НА**



**ОТ СИНТАКСИСА  
К АРХИТЕКТУРНОМУ МЫШЛЕНИЮ**

Издательство АСТ

г. Москва

УДК 004.438  
ББК 32.973.2  
К16

*При составлении данного издания использован материал книги «Essential C#» (<https://www.programming-books.io/essential/csharp/>) из проекта «Essential Programming Books» (<https://www.programming-books.io>). Материал основан на документации Stack Overflow, контент распространяется под лицензией Creative Commons Attribution-ShareAlike 3.0 (<https://creativecommons.org/licenses/by-sa/3.0>).*

**Как** научиться мыслить на C#. — Москва : Издательство АСТ, 2026. — 672 с. : ил. —  
K16 (Программирование от А до C++).  
ISBN 978-5-17-178632-8.

Перед вами фундаментальное руководство по программированию на языке C# в экосистеме .NET, которое формирует не просто знание синтаксиса, а системное объектно-ориентированное мышление разработчика, способного проектировать надежные и масштабируемые приложения.

Книга строится на принципе глубокого понимания внутренних механизмов языка вместо механического заучивания конструкций. Каждая концепция раскрывается через трехмерную модель: что представляет собой инструмент (от базовых типов данных до событий и делегатов), почему он работает именно так в среде .NET и когда его следует применять. Читатель проходит осознанный путь от написания первых строк кода до создания сложных кроссплатформенных решений.

Особое внимание уделено культуре качественного кода и архитектурному планированию. Правильные практики управления памятью, оценка вычислительной сложности алгоритмов (например, при работе с рекурсией) и паттерны проектирования интегрированы в каждый раздел. Книга развивает алгоритмическое мышление — умение декомпозировать сложные проблемы, создавать правильные абстракции и избегать критических ошибок.

Автор рассматривает C# не как самоцель, а как мощный инструмент формирования инженерного мышления. Читатель научится видеть программу как живую экосистему взаимодействующих объектов.

Издание предназначено для всех, кто хочет освоить разработку на качественном уровне — от абсолютных новичков до специалистов, стремящихся систематизировать свои знания и обрести уверенность в написании профессионального кода.

**УДК 004.438**  
**ББК 32.973.2**

**ISBN 978-5-17-178632-8**

Перевод на русский язык: ООО «Интеджер».  
Издание на русском языке: ООО «Издательство АСТ».

# Содержание

<b>Начало работы с C#</b> .....	27
Версии .....	27
Создание нового консольного приложения в Visual Studio .....	28
Объяснение.....	28
Использование командной строки .....	28
Создание нового проекта консольного приложения в Visual Studio и запуск в режиме отладки.....	30
Создание новой программы с использованием Mono.....	33
Создание новой программы с использованием .NET Core.....	33
Вывод в командной строке.....	35
Создание нового запроса с использованием LinqPad .....	35
Создание нового проекта с использованием Xamarin Studio .....	38
<b>Литералы</b> .....	42
Синтаксис .....	42
Булевы литералы.....	42
Литералы типа byte .....	42
Литералы типа sbyte .....	42
Символьные литералы .....	42
Литералы типа uint.....	43
Литералы типа ulong .....	43
Литералы типа int.....	43
Литералы типа long.....	43
Строковые литералы.....	43
Литералы типа short .....	44
Литералы типа ushort.....	44
Литералы типа float .....	44
Литералы типа double.....	44
Литералы типа decimal .....	44
<b>Операторы</b> .....	45
Синтаксис .....	45
Приоритет операторов .....	45
Перегружаемые операторы (Overloadable Operators).....	47
Перегрузка операторов равенства .....	49
Реляционные операторы.....	50
Равенство.....	50
Не равно .....	51
Больше чем .....	51
Меньше чем.....	51
Больше или равно.....	52
Меньше или равно.....	52
Операторы неявного и явного приведения (Implicit Cast и Explicit Cast).....	52
Короткозамкнутые операторы.....	53
Логическое И.....	54
Логическое ИЛИ .....	54
Пример использования.....	54

Тернарный оператор.....	54
? . (Оператор условного обращения к null).....	56
Оператор «исключающего или».....	56
Оператор default.....	57
Тип-значение (where T : struct).....	57
Ссылочный тип (where T : class).....	57
Оператор присваивания <= >.....	57
sizeof.....	57
?? Оператор объединения с null.....	58
Операторы побитового сдвига.....	58
Левый сдвиг.....	59
Правый сдвиг.....	59
=> Лямбда-оператор.....	59
Операторы членов класса — условный доступ к члену.....	60
Операторы членов класса — условное индексирование.....	60
Постфиксный и префиксный инкремент и декремент.....	60
typeof.....	61
Бинарные операторы с присваиванием.....	61
Оператор nameof.....	62
Операторы доступа к членам класса — доступ к членам.....	62
Операторы доступа к членам класса — вызов функции.....	62
Операторы доступа к членам класса — индексирование агрегатного объекта.....	63
<b>Условные операторы.....</b>	<b>64</b>
Оператор If-Else.....	64
Условия оператора if являются стандартными булевыми выражениями и значениями.....	64
Оператор If-Else If-Else.....	65
Операторы Switch.....	66
<b>Оператор равенства.....</b>	<b>67</b>
Виды равенства в C# и оператор равенства.....	67
<b>Equals и GetHashCode.....</b>	<b>68</b>
Написание хорошего переопределения GetHashCode.....	68
Поведение Equals по умолчанию.....	69
Переопределение Equals и GetHashCode для пользовательских типов.....	70
Equals и GetHashCode в IEqualityComparator.....	71
<b>Оператор объединения с null (Null-Coalescing Operator).....</b>	<b>73</b>
Синтаксис.....	73
Базовое использование.....	73
Null-проваливание и создание цепочек.....	73
Не допускающий null.....	74
Допускающий null.....	74
Множественное объединение.....	74
Создание цепочек условного null.....	74
Оператор объединения с null при вызовах методов.....	75
Использование существующего или создание нового.....	75
Ленивая инициализация свойств (properties) с оператором объединения с null.....	75
Потокобезопасность.....	75
Синтаксический сахар C# 6 с использованием выражений-тел.....	75
Пример в паттерне MVVM.....	76
<b>Условные операторы проверки на null (Null-conditional Operators).....</b>	<b>77</b>
Синтаксис.....	77
Замечания.....	77

Условный оператор проверки на null (Null-Conditional Operator).....	77
Связывание оператора в цепочку (Chaining the Operator).....	78
Комбинирование с оператором объединения null (Combining with the Null-Coalescing Operator).....	78
Условный индекс проверки на null (The Null-Conditional Index).....	78
Избежание NullReferenceExceptions.....	78
Условный оператор проверки на null может использоваться с методами расширения (Extension Method).....	79
<b>Оператор nameof</b> .....	<b>80</b>
Введение .....	80
Синтаксис.....	80
Базовое использование.....	80
Вывод имени переменной.....	80
Вызов события PropertyChanged.....	80
Проверка аргументов и защитные условия (Guard Clauses).....	81
Строго типизированные ссылки на действия MVC.....	82
Обработка событий PropertyChanged .....	82
Применение к параметру обобщенного типа (generic type parameter).....	83
Применение к квалифицированным идентификаторам.....	83
Вывод имени параметра.....	83
<b>Дословная строка (Verbatim String)</b> .....	<b>84</b>
Синтаксис .....	84
Интерполированная дословная строка (Interpolated verbatim string) .....	84
Экранирование двойных кавычек .....	84
Дословные строки указывают компилятору не использовать символы экранирования.....	84
Многострочные строки .....	85
<b>Основные операции со строками</b> .....	<b>86</b>
Дополнение строки до фиксированной длины.....	86
Корректный переворот строки .....	86
Форматирование строки.....	88
Проверка на пустую строку с использованием String.IsNullOrEmpty и String.IsNullOrWhiteSpace.....	88
Получение x символов с правой стороны строки .....	88
Преобразование десятичного числа в двоичный, восьмеричный и шестнадцатеричный формат.....	91
Обрезка нежелательных символов с начала и/или конца строк .....	91
String.Trim() .....	91
String.TrimStart() и String.TrimEnd() .....	91
Создание строки из массива .....	92
Строка из массива символов (char array).....	92
Строка из списка символов (List of char) .....	92
Строка из списка строк (List of Strings).....	92
Строка из массива строк (array of strings).....	92
Форматирование с использованием ToString .....	92
Преобразование целого числа в отформатированную строку.....	92
Преобразование double в отформатированную строку .....	93
Форматирование DateTime с использованием ToString .....	93
Разделение строки по определенному символу .....	93
Объединение массива строк в новую строку .....	93
Получение символа по определенному индексу и перечисление строки .....	93
Определение того, начинается ли строка с заданной последовательности .....	94
Поиск строки внутри строки .....	94

Разделение строки по другой строке .....	94
Получение подстрок заданной строки .....	94
Замена строки внутри строки.....	94
Изменение регистра символов в строке .....	95
Объединение массива строк в одну строку.....	95
Конкатенация строк.....	95
<b>String.Format .....</b>	<b>96</b>
Введение .....	96
Синтаксис.....	96
Параметры .....	96
Начиная с C# 6.0.....	96
Места, где String.Format встроен во фреймворк .....	96
Создание пользовательского поставщика формата .....	97
Форматирование даты.....	97
Форматирование валюты.....	99
Точность.....	100
Символ валюты.....	100
Позиция символа валюты.....	100
Пользовательский разделитель десятичных знаков.....	100
Использование пользовательского числового формата.....	101
Выравнивание по левому краю, по правому краю, заполнение пробелами.....	101
Числовые форматы.....	101
Экранирование фигурных скобок внутри выражения String.Format.....	102
ToString.....	102
Отношение с ToString .....	102
Предупреждения и ограничения форматирования .....	103
<b>Конкатенация строк.....</b>	<b>104</b>
Оператор +.....	104
Конкатенация строк с использованием System.Text.StringBuilder.....	104
Конкатенация элементов массива строк с использованием String.Join.....	104
Конкатенация строк с использованием форматирования строк \$ .....	105
<b>Манипуляции со строками .....</b>	<b>106</b>
Замена строки внутри строки.....	106
Изменение регистра символов в строке .....	106
Поиск строки внутри строки .....	106
Удаление пробелов из строки.....	107
Подстрока для извлечения части строки.....	107
Разделение строки с использованием разделителя .....	107
Объединение массива строк в одну строку.....	108
Конкатенация строк.....	108
<b>Интерполяция строк.....</b>	<b>109</b>
Основы интерполяции строк.....	109
Внутренняя реализация .....	109
Форматирование чисел в строках.....	109
Форматирование дат в строках.....	109
Выравнивание вывода .....	110
Выравнивание влево .....	110
Выравнивание вправо .....	110

Выравнивание со спецификаторами формата .....	111
Выражения.....	111
<b>Escape-последовательности строк.....</b>	<b>112</b>
Синтаксис .....	112
Замечания .....	112
Экранирование специальных символов в строковых литералах .....	112
Обратный слеш .....	112
Кавычки.....	113
Переводы строк.....	113
Escape-последовательности символов Unicode.....	113
Экранирование специальных символов в символьных литералах .....	113
Апострофы.....	113
Обратный слеш .....	113
Использование escape-последовательностей в идентификаторах.....	113
Нераспознанные escape-последовательности вызывают ошибки времени компиляции.....	114
<b>StringBuilder .....</b>	<b>115</b>
Что такое StringBuilder и когда его использовать.....	115
Использование StringBuilder для создания строки из большого количества записей .....	116
<b>Регулярные выражения (Regex) .....</b>	<b>117</b>
Разбор регулярных выражений (Regex Parsing) .....	117
Синтаксис .....	117
Параметры .....	117
Примечания .....	117
Одиночное совпадение (Single match) .....	118
Множественные совпадения (Multiple matches).....	118
<b>DateTime .....</b>	<b>119</b>
Форматирование DateTime .....	119
Пользовательское форматирование DateTime.....	119
DateTime.AddDays(Double) .....	120
DateTime.AddHours(Double) .....	120
DateTime.TryParse(String, DateTime).....	121
DateTime.Parse(String).....	121
DateTime.Compare(DateTime t1, DateTime t2).....	121
DateTime.DaysInMonth(Int32, Int32).....	121
DateTime.AddYears(Int32).....	122
Предупреждение о чистых функциях при работе с DateTime .....	122
DateTime.AddMillisecondsDouble .....	123
DateTime.TryParseExact(StringStringIFormatProviderDateTimeStylesDateTime .....	123
DateTime как инициализатор в цикле for .....	126
DateTime.ToString, ToShortDateString, ToLongDateString и ToString с форматированием .....	126
Текущая дата .....	126
Parse и TryParse с информацией о культуре .....	127
DateTime.ParseExact(StringStringIFormatProvider .....	127
Преобразование строки определенного формата в эквивалентный объект DateTime .....	127
Преобразование строки даты-времени в эквивалентный объект DateTime без какого-либо специфичного формата культуры .....	127
Преобразование строки даты-времени в эквивалентный объект DateTime без какого-либо специфичного формата культуры с различным форматом .....	127
DateTime.AddTimeSpan .....	128

<b>Массивы</b> .....	<b>129</b>
Синтаксис .....	129
Примечания .....	129
Объявление массива .....	129
Неявно типизированные массивы .....	130
Копирование массивов .....	130
Инициализация массива, заполненного повторяющимся не-дефолтным значением .....	131
Сравнение массивов на равенство .....	131
Многомерные массивы .....	132
Получение и установка значений массива .....	132
Итерация по массиву .....	132
Создание массива последовательных чисел .....	133
Использование .....	133
Зубчатые массивы .....	133
Объявление зубчатого массива .....	134
Получение/установка значений .....	134
Примечание о порядке скобок .....	134
Ковариантность массивов .....	135
Проверка, содержит ли один массив другой массив .....	135
Массивы как экземпляры IEnumerable .....	136
<b>Алгоритм циклического поворота массива</b> .....	<b>137</b>
Введение .....	137
Пример универсального метода, который поворачивает массив на заданное смещение .....	137
<b>Enum</b> .....	<b>139</b>
Введение .....	139
Синтаксис .....	139
Примечания .....	139
Основы Enum .....	139
Enum как флаги .....	141
Использование нотации для флагов .....	143
Тестирование значений перечисления флагового типа с помощью побитовой логики .....	144
Преобразование enum в строку и обратно .....	144
Добавление и удаление значений из флагового перечисления .....	145
Значение по умолчанию для enum равно НУЛЮ .....	145
Добавление дополнительной описательной информации к значению перечисления .....	146
Перечисления могут иметь неожиданные значения .....	147
Получение всех значений членов перечисления .....	147
Побитовая манипуляция с использованием перечислений .....	148
Пример 1: C [Flags] .....	148
Пример 2: Без [Flags] .....	148
<b>Кортежи (Tuples)</b> .....	<b>149</b>
Доступ к элементам кортежа .....	149
Создание кортежей .....	149
Сравнение и сортировка кортежей .....	149
Возврат нескольких значений из метода .....	150
<b>Обзор коллекций C#</b> .....	<b>151</b>
HashSet .....	151
Dictionary<TKey, TValue> .....	151

---

Дублирующий ключ при использовании инициализации коллекции.....	152
List .....	152
T[] — Массив T.....	152
SortedSet .....	153
Stack.....	153
LinkedList.....	153
Queue.....	154
<b>GUID .....</b>	<b>155</b>
Введение .....	155
Примечания .....	155
Получение строкового представления Guid.....	155
Создание Guid .....	155
Объявление GUID, допускающего значение null.....	156
<b>BigInteger.....</b>	<b>157</b>
Примечания.....	157
Когда использовать .....	157
Альтернативы.....	157
Вычисление первого 1000-значного числа Фибоначчи.....	157
<b>Инициализаторы коллекций (Collection Initializers) .....</b>	<b>159</b>
Инициализаторы коллекций .....	159
Индексные инициализаторы C# 6 (Index Initializers).....	159
Инициализация Dictionary .....	160
Инициализаторы коллекций в пользовательских классах .....	161
Использование инициализатора коллекций внутри инициализатора объекта .....	162
Инициализаторы коллекций с массивами параметров (Parameter Arrays) .....	162
<b>Циклы.....</b>	<b>164</b>
Цикл Foreach.....	164
Синтаксис.....	164
Замечания.....	164
Цикл For .....	165
Цикл Do – While.....	166
Стили циклов.....	166
While .....	166
Do .....	166
For.....	166
Foreach .....	167
Метод Foreach.....	167
Linq Parallel Foreach.....	167
Вложенные циклы .....	167
break .....	168
Цикл While.....	168
continue.....	169
<b>Итераторы.....</b>	<b>170</b>
Примечания.....	170
Создание итераторов с использованием Yield.....	170
Пример простого числового итератора .....	171

<b>Типы значений против ссылочных типов .....</b>	<b>172</b>
Синтаксис .....	172
Введение .....	172
Типы значений.....	172
Ссылочные типы .....	172
Основные различия.....	172
Типы значений существуют в стеке, ссылочные типы существуют в куче.....	172
Типы значений не изменяются, когда вы изменяете их в методе, ссылочные типы изменяются.....	173
Что если я хочу изменить их?.....	173
Типы значений не могут быть null, ссылочные типы могут.....	173
Передача по ссылке с использованием ключевого слова ref.....	173
Изменение значений в другом месте .....	175
Передача по ссылке .....	176
Различие между параметрами метода ref и out .....	176
Параметры ref против out .....	177
Присвоение.....	178
<b>Встроенные типы .....</b>	<b>179</b>
Преобразование упакованных типов значений.....	179
Тип значения — char .....	179
Тип значения — short int long знаковые 16-битные 32-битные 64-битные целые числа .....	179
Тип значения — ushort uint ulong беззнаковые 16-битные 32-битные 64-битные целые числа .....	180
Тип значения — bool.....	180
Сравнения с упакованными типами значений .....	181
Неизменяемый ссылочный тип — string.....	181
<b>Псевдонимы встроенных типов.....</b>	<b>182</b>
Таблица встроенных типов.....	182
<b>Анонимные типы .....</b>	<b>183</b>
Анонимные vs динамические .....	183
Создание анонимного типа.....	183
Равенство анонимных типов.....	184
Обобщенные методы с анонимными типами .....	184
Создание экземпляров обобщенных типов с анонимными типами .....	184
Неявно типизированные массивы.....	185
<b>Динамический тип (Dynamic type).....</b>	<b>186</b>
Примечания.....	186
Создание динамического объекта со свойствами (properties).....	186
Создание динамической переменной.....	186
Возврат dynamic.....	187
Обработка конкретных типов, неизвестных во время компиляции.....	187
<b>Преобразование типов (Type Conversion).....</b>	<b>189</b>
Примечания.....	189
Явное преобразование типов (Explicit Type Conversion) .....	189
Пример неявного оператора (implicit operator) из MSDN.....	189
<b>Кастинг (Casting).....</b>	<b>191</b>
Замечания .....	191
Кастинг объекта к базовому типу.....	191
Проверка совместимости без кастинга.....	192
Явный кастинг (Explicit Casting).....	192

Неявный кастинг (Implicit Casting) .....	192
Явные числовые преобразования (Explicit Numeric Conversions).....	192
Операторы преобразования (Conversion Operators).....	193
Операции кастинга LINQ.....	194
Безопасный явный кастинг с оператором as.....	195
<b>Обнуляемые типы (Nullable types).....</b>	<b>196</b>
Синтаксис .....	196
Примечания .....	196
Получение значения по умолчанию из обнуляемого типа .....	196
Инициализация обнуляемого типа.....	197
Проверка, имеет ли обнуляемый тип значение.....	197
Получение значения обнуляемого типа .....	198
Проверка, является ли параметр универсального типа обнуляемым типом .....	198
Значение по умолчанию обнуляемых типов — null .....	198
Эффективное использование базового аргумента NullableT.....	199
<b>Конструкторы и финализаторы .....</b>	<b>201</b>
Статический конструктор.....	201
Паттерн конструктора синглтона .....	202
Конструктор по умолчанию.....	203
Принуждение к вызову статического конструктора.....	204
Вызов конструктора из другого конструктора.....	204
Вызов конструктора базового класса .....	205
Финализаторы в производных классах.....	206
Исключения в статических конструкторах.....	206
Вызов виртуальных методов в конструкторе .....	207
Обобщенные статические конструкторы .....	208
Инициализация конструктора и свойств.....	208
<b>Модификаторы доступа .....</b>	<b>211</b>
Примечания.....	211
public.....	211
private .....	211
internal.....	212
protected .....	213
protected internal .....	213
<b>Интерфейсы .....</b>	<b>216</b>
Реализация интерфейса.....	216
Явная реализация интерфейса .....	216
Реализация нескольких интерфейсов .....	218
Основы интерфейсов .....	218
IComparableT как пример реализации интерфейса.....	220
Зачем мы используем интерфейсы.....	222
Соккрытие членов с помощью явной реализации .....	223
<b>Статические классы.....</b>	<b>225</b>
Ключевое слово static.....	225
Время жизни статического класса .....	226
<b>Реализация Singleton.....</b>	<b>228</b>
Статически инициализированный Singleton .....	228
Ленивый потокобезопасный Singleton с использованием Lazy .....	228
Ленивый потокобезопасный Singleton с использованием двойной проверки блокировки .....	229

Альтернативная реализация ленивого потокобезопасного singleton для .NET 3.5 или старше .....	229
Освобождение экземпляра Singleton, когда он больше не нужен .....	230
<b>Dependency Injection .....</b>	<b>232</b>
Замечания .....	232
Dependency Injection в C# и ASP.NET с использованием Unity .....	232
Dependency injection с использованием MEF .....	235
<b>Частичные классы и методы .....</b>	<b>238</b>
Введение .....	238
Синтаксис .....	238
Примечания .....	238
Частичные классы .....	238
Частичные методы .....	239
Частичные классы, наследующие от базового класса .....	240
<b>Инициализаторы объектов (Object initializers) .....</b>	<b>241</b>
Синтаксис .....	241
Примечания .....	241
Простое использование .....	241
Использование без конструктора по умолчанию .....	241
Использование с анонимными типами (anonymous types) .....	242
<b>Методы .....</b>	<b>243</b>
Вызов метода .....	243
Анонимный метод .....	243
Параметры и аргументы .....	244
Типы возвращаемых значений .....	244
Параметры по умолчанию .....	245
Перегрузка методов .....	246
Объявление метода .....	247
Права доступа .....	248
<b>Методы расширения (Extension Methods) .....</b>	<b>249</b>
Синтаксис .....	249
Примечания .....	249
Какое пространство имен (namespace)? .....	249
Применимость .....	249
Методы расширения — обзор .....	250
Проверка на null .....	253
Явное использование метода расширения (extension method) .....	253
Когда вызывать методы расширения как статические методы .....	254
Использование static .....	254
Обобщенные методы расширения (Generic Extension Methods) .....	254
Методы расширения могут видеть только публичные (public) или внутренние (internal) члены расширяемого класса .....	256
Методы расширения для создания цепочек .....	256
Методы расширения с перечислениями .....	257
Диспетчеризация методов расширения на основе статического типа .....	259
Методы расширения (extension methods) для интерфейсов .....	260
Методы расширения не поддерживаются динамическим кодом .....	260
Методы расширения в сочетании с интерфейсами .....	261
Расширения и интерфейсы вместе обеспечивают DRY-код и функциональность, подобную миксинам .....	262

Пример метода расширения для IList для сравнения двух списков .....	263
Методы расширения как строго типизированные обертки .....	264
Использование методов расширения (extension methods) для создания классов-мапперов .....	265
Использование методов расширения для создания новых типов коллекций, например, DictList .....	266
Методы расширения для обработки особых случаев .....	267
Использование методов расширения со статическими методами и обратными вызовами (callbacks) .....	268
<b>Именованные аргументы (Named Arguments) .....</b>	<b>271</b>
Порядок аргументов необязателен .....	271
Именованные аргументы могут сделать ваш код более понятным .....	271
Именованные аргументы и необязательные параметры .....	272
<b>Именованные и необязательные аргументы .....</b>	<b>274</b>
Примечания .....	274
Именованные аргументы (Named Arguments) .....	274
Необязательные аргументы (Optional Arguments) .....	274
Необязательные аргументы .....	274
Использование атрибута Optional .....	275
Именованные аргументы .....	276
Ограничения использования именованных аргументов .....	277
<b>Аннотация данных (Data Annotation) .....</b>	<b>278</b>
Основы аннотации данных .....	278
Использование .....	278
Создание пользовательского атрибута валидации .....	278
Ручное выполнение атрибутов валидации .....	279
Контекст валидации .....	279
Валидация объекта и всех его свойств .....	279
Валидация свойства объекта .....	280
Атрибут моделирования данных EditableAttribute .....	280
Атрибуты валидации .....	281
Пример: RequiredAttribute .....	282
Пример: StringLengthAttribute .....	282
Пример: RangeAttribute .....	282
Пример: CustomValidationAttribute .....	282
Атрибут отображения DisplayNameAttribute .....	283
<b>Ключевые слова (Keywords) .....</b>	<b>285</b>
Введение .....	285
Примечания .....	285
as .....	285
Пример использования .....	286
goto .....	287
goto как: .....	287
Повторная попытка при исключении (Exception Retry) .....	287
volatile .....	288
checked и unchecked .....	290
virtual, override и new .....	291
virtual и override .....	291
new .....	292

Использование <code>override</code> не является обязательным.....	293
Производные классы могут вводить полиморфизм.....	294
Виртуальные методы не могут быть приватными.....	295
<code>stackalloc</code> .....	295
Примечания .....	296
<code>break</code> .....	296
<code>const</code> .....	298
<code>for</code> .....	299
<code>async await</code> .....	300
<code>abstract</code> .....	301
<code>is</code> .....	303
<code>default</code> .....	304
<code>fixed</code> .....	305
Фиксированные переменные (Fixed Variables).....	305
Фиксированный размер массива (Fixed Array Size).....	305
<code>sealed</code> .....	305
<code>typeof</code> .....	306
<code>readonly</code> .....	306
<code>this</code> .....	307
<code>dynamic</code> .....	308
<code>foreach</code> .....	309
<code>try catch finally throw</code> .....	310
<code>namespace</code> .....	311
<code>void</code> .....	312
<code>ref out</code> .....	313
Использование ключевого слова <code>out</code> в качестве модификатора обобщений (Generic Modifier).....	314
<code>float double decimal</code> .....	314
<code>float</code> .....	314
<code>double</code> .....	314
<code>decimal</code> .....	315
<code>char</code> .....	315
<code>params</code> .....	316
<code>base</code> .....	317
Выбор конструктора.....	317
Вызов базовой реализации виртуального метода.....	317
<code>operator</code> .....	318
<code>string</code> .....	320
<code>null</code> .....	320
<code>return</code> .....	321
<code>while</code> .....	321
<code>continue</code> .....	323
<code>ulong</code> .....	323
<code>uint</code> .....	324
<code>unsafe</code> .....	324
<code>int</code> .....	326
<code>var</code> .....	327
<code>lock</code> .....	328
Варианты использования.....	329
<code>true false</code> .....	329
<code>internal</code> .....	329

Использование .....	329
Модификаторы доступа (Access modifiers).....	330
where .....	330
Тип значения (value type).....	330
Ссылочный тип (reference type).....	331
Гибридный тип значения/ссылочный тип (hybrid value/reference type).....	331
Конструктор по умолчанию (default constructor).....	331
extern.....	332
switch.....	333
when.....	334
struct.....	335
static.....	336
Недостатки.....	338
if if...else if... else if.....	338
do.....	339
partial.....	340
unchecked.....	342
bool.....	342
interface.....	342
long.....	343
sizeof.....	343
in.....	344
enum.....	344
ushort.....	345
delegate.....	345
implicit.....	346
sbyte.....	347
event.....	347
<b>Рекурсия.....</b>	<b>349</b>
Замечания.....	349
Рекурсия простыми словами.....	349
Последовательность Фибоначчи.....	350
Использование рекурсии для получения дерева каталогов.....	350
Рекурсивное описание структуры объекта.....	353
Вычисление степени числа.....	354
Вычисление факториала.....	355
<b>Соглашения по именованию.....</b>	<b>357</b>
Введение.....	357
Примечания.....	357
Выбирайте легко читаемые имена идентификаторов.....	357
Отдавайте предпочтение читаемости перед краткостью.....	357
Не используйте венгерскую нотацию.....	357
Аббревиатуры и акронимы.....	357
Соглашения по использованию заглавных букв.....	357
Pascal Casing (Паскаль-нотация).....	357
Camel Casing (Верблюжья нотация).....	358
Uppercase (Верхний регистр).....	358
Правила.....	358
Перечисления (Enums).....	358
Используйте имя в единственном числе для большинства перечислений.....	358