

# Содержание

<b>Предисловие</b> .....	<b>5</b>
Новое в этом издании .....	15
Трудный путь на самом деле прост .....	16
Чтение и ввод .....	17
Внимание к деталям .....	17
Обнаружение различий .....	17
Хватит смотреть, спрашивайте .....	18
Никакого копирования/вставки .....	18
Дополнительные видеоролики .....	18
О практике и настойчивости .....	19
Благодарности .....	20
<b>Упражнение 0. Настройка</b> .....	<b>21</b>
macOS .....	21
macOS: результат .....	22
Windows .....	23
Windows: результат .....	24
Linux .....	25
Linux: результат .....	26
Поиск в Интернете .....	27
Начинающим .....	27
Альтернативные текстовые редакторы .....	28
<b>Упражнение 1. Первая программа</b> .....	<b>30</b>
Результат выполнения .....	32
Практические задания .....	33
Распространенные вопросы .....	34
<b>Упражнение 2. Комментарии и символы #</b> .....	<b>36</b>
Результат выполнения .....	36
Практические задания .....	37
Распространенные вопросы .....	37
<b>Упражнение 3. Числа и математика</b> .....	<b>39</b>
Результат выполнения .....	40
Практические задания .....	41
Распространенные вопросы .....	41
<b>Упражнение 4. Переменные и имена</b> .....	<b>42</b>
Результат выполнения .....	43
Практические задания .....	43
Дополнительные рактические задания .....	44
Распространенные вопросы .....	46
<b>Упражнение 5. Дополнительно о переменных и выводе</b> .....	<b>46</b>
Результат выполнения .....	47

Практические задания	47
Распространенные вопросы	47
<b>Упражнение 6. Строки и текст</b>	49
Результат выполнения	50
Практические задания	50
Распространенные вопросы	51
<b>Упражнение 7. Еще о выводе</b>	52
Результат выполнения	53
Практические задания	53
Работа с ошибками	53
Распространенные вопросы	52
<b>Упражнение 8. Вывод, вывод</b>	55
Результат выполнения	55
Практические задания	56
Распространенные вопросы	56
<b>Упражнение 9. Вывод, вывод, вывод</b>	57
Результат выполнения	57
Практические задания	58
Распространенные вопросы	58
<b>Упражнение 10. Управляющие последовательности</b>	59
Результат выполнения	60
Управляющие последовательности	61
Практические задания	61
Распространенные вопросы	62
<b>Упражнение 11. Получение ответов на вопросы</b>	63
Результат выполнения	64
Практические задания	64
Распространенные вопросы	64
<b>Упражнение 12. Запрос ввода</b>	65
Результат выполнения	65
Практические задания	65
Распространенные вопросы	66
<b>Упражнение 13. Параметры, распаковка, переменные</b>	67
Внимание! У возможностей другое название	67
Результат выполнения	68
Практические задания	70
Распространенные вопросы	70
<b>Упражнение 14. Запросы и подтверждения</b>	72
Результат выполнения	73
Практические задания	73
Распространенные вопросы	74
<b>Упражнение 15. Чтение файлов</b>	76
Результат выполнения	77
Практические задания	78

Распространенные вопросы .....	79
<b>Упражнение 16.</b> Чтение и запись файлов .....	81
Результат выполнения.....	83
Практические задания.....	83
Распространенные вопросы .....	84
<b>Упражнение 17.</b> Еще о файлах .....	85
Результат выполнения.....	86
Практические задания.....	87
Распространенные вопросы .....	87
<b>Упражнение 18.</b> Имена, переменные, код, функции .....	89
Результат выполнения.....	91
Практические задания.....	91
Распространенные вопросы .....	92
<b>Упражнение 19.</b> Функции и переменные .....	94
Результат выполнения.....	95
Практические задания.....	96
Распространенные вопросы .....	96
<b>Упражнение 20.</b> Функции и файлы.....	98
Результат выполнения.....	99
Практические задания.....	99
Распространенные вопросы .....	100
<b>Упражнение 21.</b> Что возвращают функции.....	102
Результат выполнения.....	103
Практические задания.....	104
Распространенные вопросы .....	104
<b>Упражнение 22.</b> Что вы теперь знаете?.....	106
Что вы изучили.....	107
<b>Упражнение 23.</b> Строки, байты и кодировки символов .....	108
Предварительное исследование .....	109
Переключатели, общепринятые обозначения и кодировки... ..	111
Анализ результата выполнения кода.....	113
Анализ кода.....	114
Углубляемся в кодирование .....	117
Ломаем код .....	118
<b>Упражнение 24.</b> Дополнительная практика .....	119
Результат выполнения.....	120
Практические задания.....	121
Распространенные вопросы .....	121
<b>Упражнение 25.</b> И еще практика .....	122
Результат выполнения.....	123
Практические задания.....	125
Распространенные вопросы .....	126
<b>Упражнение 26.</b> Внимание, тест!.....	127
Распространенные вопросы .....	128

<b>Упражнение 27.</b> Обучение логике .....	129
Терминология .....	130
Таблицы истинности .....	130
Распространенные вопросы .....	132
<b>Упражнение 28.</b> Логические выражения .....	133
Результат выполнения .....	135
Практические задания .....	136
Распространенные вопросы .....	136
<b>Упражнение 29.</b> Что если.....	137
Результат выполнения .....	138
Практические задания .....	138
Распространенные вопросы .....	138
<b>Упражнение 30.</b> А если иначе...	139
Результат выполнения .....	141
Практические задания .....	141
Распространенные вопросы .....	141
<b>Упражнение 31.</b> Принятие решений .....	142
Результат выполнения .....	143
Практические задания .....	144
Распространенные вопросы .....	144
<b>Упражнение 32.</b> Циклы и списки.....	145
Результат выполнения .....	146
Практические задания .....	147
Распространенные вопросы .....	148
<b>Упражнение 33.</b> Циклы <code>while</code> .....	149
Результат выполнения .....	150
Практические задания .....	151
Распространенные вопросы .....	151
<b>Упражнение 34.</b> Доступ к элементам списка.....	153
Практические задания .....	155
<b>Упражнение 35.</b> Ветви и функции.....	156
Результат выполнения .....	158
Практические задания .....	158
Распространенные вопросы .....	159
<b>Упражнение 36.</b> Разработка и отладка .....	160
Правила конструкций <code>if</code> .....	160
Правила циклов .....	161
Советы по отладке .....	161
Домашнее задание .....	163
<b>Упражнение 37.</b> Знакомство с символами .....	163
Ключевые слова .....	163
Типы данных .....	165
Управляющие последовательности .....	165
Форматирование строк в старом стиле .....	166

Операторы .....	167
Чтение кода .....	169
Практические задания .....	170
Распространенные вопросы .....	170
<b>Упражнение 38.</b> Работа со списками .....	171
Результат выполнения .....	173
Для чего нужны списки .....	174
В каких случаях используются списки .....	175
Практические задания .....	175
Распространенные вопросы .....	176
<b>Упражнение 39.</b> Словари .....	178
Пример словаря .....	179
Результат выполнения .....	181
Для чего нужны словари? .....	182
Практические задания .....	183
Распространенные вопросы .....	183
<b>Упражнение 40.</b> Модули, классы и объекты .....	185
Модули в сравнении со словарями .....	185
Классы как мини-модули .....	187
Объекты как мини-импорты .....	188
Три способа .....	190
Первоклассный пример .....	190
Результат выполнения .....	191
Практические задания .....	191
Распространенные вопросы .....	192
<b>Упражнение 41.</b> Поговорим об ООП .....	193
Терминология .....	193
Чтение кода .....	194
Смешанное упражнение .....	195
Перевод с кода на русский язык .....	195
Перевод с русского языка в код .....	198
Дополнительное упражнение по чтению кода .....	198
Распространенные вопросы .....	199
<b>Упражнение 42.</b> Композиция, наследование, объекты и классы .....	200
Пример кода .....	201
О синтаксисе <code>class имя (объект)</code> .....	204
Практические задания .....	204
Распространенные вопросы .....	205
<b>Упражнение 43.</b> Основы объектно-ориентированного анализа и дизайна .....	206
Анализ простого игрового движка .....	207
Запись или зарисовка задачи .....	208
Извлечение ключевых концепций и их анализ .....	208

Формирование иерархии классов и схемы объектов на основе концепций .....	210
Кодинг классов и тестовый запуск .....	211
Исправление ошибок и доработка кода .....	213
Нисходящий подход против восходящего .....	214
Код игры «Готоны с планеты Перкаль 25» .....	214
Результат выполнения .....	222
Практические задания .....	224
Распространенные вопросы .....	225
<b>Упражнение 44. Наследование и композиция .....</b>	<b>226</b>
Что такое «наследование»? .....	227
Неявное наследование .....	227
Явное переопределение .....	228
Видоизменение до или после .....	229
Комбинация взаимодействий .....	231
Причины использования функции <code>super()</code> .....	232
Использование функции <code>super()</code> с методом <code>__init__</code> .....	233
Композиция .....	233
Наследование или композиция: что выбрать? .....	235
Практические задания .....	236
Распространенные вопросы .....	236
<b>Упражнение 45. Разработка игры .....</b>	<b>238</b>
Проверка созданной игры .....	239
Оформление функций .....	239
Оформление классов .....	240
Оформление кода .....	241
Оформление комментариев .....	241
Выставление оценки .....	242
<b>Упражнение 46. Каркас проекта .....</b>	<b>243</b>
Установка в среде macOS/Linux .....	243
Установка в среде Windows 10 .....	245
Подготовка каркаса каталогов проекта .....	247
Окончательная структура каталогов .....	249
Проверка проекта .....	251
Использование каркаса .....	251
Обязательный опросник .....	252
Распространенные вопросы .....	252
<b>Упражнение 47. Автоматизированное тестирование .....</b>	<b>254</b>
Создание примера для тестирования .....	254
Руководство по тестированию .....	257
Результат выполнения .....	257
Практические задания .....	257
Распространенные вопросы .....	257
<b>Упражнение 48. Расширенный пользовательский ввод .....</b>	<b>260</b>

Игровой словарь .....	261
Разделение предложений .....	261
Лексические кортежи .....	261
Анализ ввода .....	262
Исключения и числа .....	262
Тактика «сначала тест» .....	263
Что нужно тестировать? .....	265
Практические задания .....	267
Распространенные вопросы .....	267
<b>Упражнение 49.</b> Формирование предложений .....	269
Соответствия и считывание .....	269
Строение предложений .....	270
Пара слов об исключениях .....	271
Код синтаксического анализатора .....	271
Эксперименты с синтаксическим анализатором .....	274
Что нужно тестировать? .....	275
Практические задания .....	276
Распространенные вопросы .....	276
<b>Упражнение 50.</b> Ваш первый веб-сайт .....	277
Установка фреймворка Flask .....	277
Создание простого проекта .....	278
Что происходит? .....	279
Создание базовых шаблонов .....	280
Работа над ошибками .....	283
Практические задания .....	284
Распространенные вопросы .....	285
<b>Упражнение 51.</b> Получение ввода из браузера .....	287
Как устроена Всемирная паутина .....	287
Принцип работы веб-формы .....	290
Создание HTML-форм .....	292
Подготовка макета шаблона .....	294
Разработка автоматических тестов для веб-форм .....	296
Практические задания .....	298
Ломаем код .....	298
<b>Упражнение 52.</b> Игра для Всемирной паутины .....	300
Доработка игры из упражнения 43 .....	300
Разработка движка .....	306
Ваш выпускной экзамен .....	310
Распространенные вопросы .....	311
<b>Дальнейшее обучение</b> .....	312
Как изучить любой язык программирования .....	313
<b>Совет бывалого программиста</b> .....	315
<b>Приложение.</b> Экспресс-курс по оболочке командной строки. . . . .	317
Введение в оболочку командной строки .....	317

Как использовать данное приложение .....	318
Способы запомнить информацию .....	318
Подготовка .....	319
Практикум .....	320
macOS .....	320
Linux .....	320
Windows .....	321
Что вы изучили .....	321
Дополнительно .....	322
Linux/macOS .....	322
Windows .....	323
Пути, папки и каталоги ( <code>pwd</code> ) .....	324
Практикум .....	324
Linux/macOS .....	324
Windows .....	325
Что вы изучили .....	325
Дополнительно .....	326
Если вы заблудились .....	326
Практикум .....	327
Что вы изучили .....	327
Создание каталога ( <code>mkdir</code> ) .....	327
Практикум .....	327
Linux/macOS .....	327
Windows .....	328
Что вы изучили .....	329
Дополнительно .....	329
Смена каталога ( <code>cd</code> ) .....	330
Практикум .....	330
Linux/macOS .....	330
Windows .....	330
Что вы изучили .....	333
Дополнительно .....	334
Вывод содержимого каталога ( <code>ls</code> ) .....	335
Практикум .....	335
Linux/macOS .....	335
Windows .....	336
Что вы изучили .....	339
Дополнительно .....	339
Удаление каталога ( <code>rmdir</code> ) .....	340
Практикум .....	342
Linux/macOS .....	340
Windows .....	341
Что вы изучили .....	342
Дополнительно .....	342



Работа со стеком ( <code>pushd</code> , <code>popd</code> ) .....	343
Практикум .....	343
Linux/macOS .....	343
Windows .....	344
Что вы изучили .....	345
Дополнительно .....	346
Создание пустых файлов ( <code>touch</code> , <code>New-Item</code> ) .....	346
Практикум .....	346
Linux/macOS .....	346
Windows .....	347
Что вы изучили .....	347
Дополнительно .....	347
Копирование файла ( <code>cp</code> ) .....	348
Практикум .....	348
Linux/macOS .....	348
Windows .....	348
Что вы изучили .....	351
Дополнительно .....	351
Перемещение файла ( <code>mv</code> ) .....	352
Практикум .....	352
Linux/macOS .....	352
Windows .....	352
Что вы изучили .....	354
Дополнительно .....	354
Просмотр файла ( <code>less</code> , <code>more</code> ) .....	354
Практикум .....	355
Linux/macOS .....	355
Windows .....	355
Что вы изучили .....	355
Дополнительно .....	356
Вывод содержимого файла ( <code>cat</code> ) .....	356
Практикум .....	356
Linux/macOS .....	356
Windows .....	357
Что вы изучили .....	357
Дополнительно .....	357
Удаление файла ( <code>rm</code> ) .....	358
Практикум .....	358
Linux/macOS .....	358
Windows .....	359
Что вы изучили .....	360
Дополнительно .....	360
Выход из оболочки ( <code>exit</code> ) .....	360
Практикум .....	360

Linux/macOS .....	360
Windows .....	361
Что вы изучили .....	361
Дополнительно .....	361
Дальнейшее обучение.....	362
Руководства по Unix Bash.....	362
Руководства по PowerShell.....	362
Предметный указатель .....	363

# Предисловие

Эта простая книга предназначена для обучения вас программированию с нуля. Хотя ее название<sup>1</sup> звучит как «трудный способ изучения кодинга», на самом деле это не так. Слово «трудный» используется потому, что в книге применена техника обучения, называемая инструкцией. Инструкции заключаются в последовательности созданных мной упражнений, направленных на закрепление навыков программирования через повторение. Такой метод обучения весьма результативен для новичков, которые не имеют знаний и должны приобрести базовые навыки, прежде чем смогут понять более сложные темы. Данный метод используется во всех сферах, от боевых искусств и музыки до изучения элементарной математики и обучения чтению.

Эта книга заложит и укрепит навыки программирования на Python с помощью техник практики и запоминания, позволяя переходить к решению все более сложных задач. К концу книги вы приобретете знания, необходимые для изучения более сложных тем программирования. Я бы сказал, что моя книга даст вам «черный пояс программиста». Это означает, что вы приобретете достаточно навыков, чтобы начать программировать.

Если вы будете усердно работать, не станете торопиться и приобретете эти навыки, вы научитесь кодингу.

## Новое в этом издании

В этой книге описана работа с версией Python 3.6. Я выбрал эту версию Python, потому что в ней реализована новая улучшенная система форматирования строк, которой проще пользоваться, чем предыдущей версией 4 (или 3, уже не помню, так как их было много). Новички могут столкнуться с некоторыми трудностями в освоении версии Python 3.6, но в этой книге я покажу, как с ними справляться. Особенно неприятная трудность заключается в том, что в некоторых случаях Python 3.6 выдает очень скудные сообщения об ошибках, но я помогу вам с ними разобраться.

Также я дополнил видеоуроки, вложив в них свой более чем пятилетний опыт обучения пользователей работе с Python. Вы можете посмотреть эти видеоролики на сайте [informit.com/title/9780134692883](http://informit.com/title/9780134692883). Ранее, в видеороликах было записано, как я выполняю упражнения. В новых видеороликах, помимо этого,

<sup>1</sup> В оригинале книга называется Learn Python 3 the Hard way. — Прим. ред.

вы увидите, как сделать каждое упражнение неправильно, а затем все исправить. Так вы приобретете навык отладки кода. Он научит вас исправлять проблемы, с которыми вы столкнетесь, а также, покажет, как Python выполняет созданные вами программы. Цель данной методологии заключается в том, чтобы дать вам представление о способах выполнения Python вашего кода, чтобы было проще разобраться, почему он не работает. Вы также узнаете много полезных подсказок для отладки неработающих программ.

И наконец, в этой книге полностью от начала и до конца поддерживается операционная система Microsoft Windows 10. В предыдущем издании основное внимание уделялось системам типа Unix, таким как macOS и Linux, а Windows была не в приоритете. Перед тем, как я сел за эту книгу, корпорация Microsoft начала серьезно относиться к инструментам разработки с открытым исходным кодом и программистам, и я не мог проигнорировать это событие и не применить среду разработки под Windows для программирования на языке Python. Видеоролики записаны в среде Microsoft Windows и отражают выполнение различных сценариев, но показана работа и в операционных системах macOS и Linux для полноты восприятия. Я расскажу вам обо всех шагах на каждой платформе, опишу инструкции по установке и дам все необходимые советы.

## Трудный путь на самом деле прост

Читая эту книгу и изучая язык программирования, вы будете выполнять вероятно простые действия, через которые прошли все программисты:

1. Изучайте каждое упражнение.
2. С точностью вводите код каждого примера.
3. Запускайте программу.

И все. Процесс может быть довольно сложным на первых порах, но все равно придерживайтесь данного курса. Если вы будете читать эту книгу и выполнять упражнения пару часов в сутки, то приобретете отличную основу для перехода к изучению другой книги. С помощью этой книги вы, может, и не получите «реальных» навыков программирования, но точно освоите основы, необходимые для изучения языка.

Суть этой книги в том, чтобы научить вас трем самым основным навыкам, необходимым каждому начинающему программисту: чтению и вводу, вниманию к деталям и обнаружению различий.

## Чтение и ввод

Очевидно, что, если у вас есть проблемы с вводом команд, вам будет сложно разобраться с кодом. Особенно, если вы набираете явно лишние символы в исходном коде. Без этого простого навыка вы не сможете даже на самом элементарном уровне разобраться в том, как работает программное обеспечение.

Набирая и выполняя примеры программ, вы запоминаете имена элементов языка программирования и учитесь читать код.

## Внимание к деталям

Еще один навык, который отличает хороших программистов от плохих, – это внимание к деталям. На самом деле, он применим в любой профессии. Не обращая внимания на мельчайшие детали в вашей работе, вы пропускаете важнейшие элементы результата вашего труда. В программировании отсутствие данного навыка чревато багами, а также трудностями, возникающими у пользователей программного обеспечения.

Читая книгу и в точности воспроизводя каждый пример, вы натренируетесь отслеживать детали каждого своего действия.

## Обнаружение различий

Очень важный навык, который у большинства программистов развивается с течением времени, – это способность обращать внимание на различия между двумя фрагментами кода. Опытный программист может сразу определить мельчайшие различия. Существуют программные инструменты, позволяющие упростить и автоматизировать процесс сравнения, но мы не будем использовать их. Сначала вы должны пойти трудным путем – научиться определять различия наглядно, а потом уже можно использовать инструменты.

Во время выполнения упражнений и набора кода вы будете делать ошибки. Это неизбежно; даже опытные программисты совершают их. Ваша задача сравнить исходный код и написанный вами, а затем устранить все недочеты. Поступая таким образом, вы будете тренироваться замечать опечатки, ошибки, баги и другие проблемы.

## Хватит смотреть, спрашивайте

Когда вы пишете код, обязательно случаются баги. «Баг» – это дефект, ошибка или проблема в коде, который вы написали. Легенда гласит, что термин закрепился в речи программистов из-за того, что некое насекомое залетело в один из первых компьютеров и привело к выходу его из строя. Чтобы ввести компьютер в строй, было необходимо его починить или отладить. В мире программ существует много багов. Крайне много.

Как и то легендарное насекомое, ваши баги будут прятаться где-то в коде, и вам нужно будет их разыскать. Вы не можете просто сидеть перед монитором своего компьютера, разглядывая код, в надежде, что ответ придет к вам сам. Так вы не получите необходимой информации – данных.

Вам нужно встать и найти насекомое.

Чтобы это сделать, вам придется выполнить свой код и понять, что происходит (или не происходит), или взглянуть на проблему под другим углом. В этой книге я часто говорю вам «хватит смотреть, спрашивайте». Я расскажу, как сделать так, чтобы по коду вы могли определить, что происходит, и вывести полученную информацию в возможное решение проблемы. Я также покажу, как разными способами «смотреть» на код, чтобы у вас было больше информации и понимания.

## Никакого копирования/вставки

Необходимо вводить код каждого упражнения вручную. Смысла в копировании/вставке нет, можно просто не выполнять их. Смысл этих упражнений заключается в получении навыков чтения, написания и понимания кода. Если копировать и вставлять код, эффективность упражнений сходит на нет.

## Дополнительные видеоролики

К книге прилагается обширный набор видеороликов, демонстрирующих, как работает код, и, самое главное, как его «сломать». Видео – идеальный способ продемонстрировать множество намеренно допущенных типичных ошибок и способов их исправления.

Я также продемонстрирую некоторые приемы отладки. В своих видеороликах я покажу вам, как «заканчивать смотреть на код и начать спрашивать» о том, что пошло не так. Вы можете смотреть эти видеоролики в Интернете на сайте [informit.com/title/9780134692883](http://informit.com/title/9780134692883).

## О практике и настойчивости

Пока вы учитесь программировать, я учусь играть на гитаре. Сначала я тренировался около двух часов каждый день. Я играл звукоряды, аккорды и арпеджио в течение часа, а затем изучал теорию музыки, сольфеджио, песни и все остальное. Спустя несколько дней я стал тренироваться играть на гитаре и изучать музыку уже восемь часов в день, потому что проникся и понял, как это весело. Для меня повторение – естественный и простой способ чему-то научиться. И я знаю, что, чтобы достичь хороших результатов, нужно практиковаться каждый день, даже если в определенный день хочется прогулять занятие (что бывает частенько) или же процесс обучения продвигается с трудом. Продолжайте пытаться, и в конечном счете, все станет проще и веселее.

Пока я писал эту книгу и аналогичное руководство по языку Ruby, я открыл для себя рисование и живопись. Я влюбился в изобразительное искусство в возрасте 39 лет и стал проводить каждый день, изучая его таким же образом, как обучался игре на гитаре, музыке и программированию. Я собрал коллекцию обучающих книг, делал все, что там было написано, рисовал каждый день, и сосредотачивался на том чтобы получать удовольствие от процесса обучения. Я ни в коем случае не «художник», но теперь могу заявить, что умею рисовать. Так же как я изучал искусство, я обучаю и вас с помощью этой книги. Если раздробить задачу на маленькие упражнения и уроки, и выполнять их каждый день, вы можете научиться абсолютно всему. Если вы сосредоточитесь на медленном постижении знаний и получении удовольствия от процесса обучения, вы выиграете, независимо от того, насколько хорошо будете в своем деле преуспевать.

Когда вы изучите эту книгу и продолжите программировать, помните, что начинание на первых порах может быть трудным. Возможно, вы боитесь неудачи, поэтому сдаетесь при первой трудности. А может вам не хватает самодисциплины, и вы ничего не делаете, так как «это скучно». Или, может, вы уверяете себя в своей «одаренности», поэтому все, что может выставить вас глупым или неопытным, не стоит того, чтобы попытаться. Может быть, вы конкурентоспособны и несправедливо сравнивать вас с кем-то вроде меня, кто занимается программированием в течение более 20 лет.