

Содержание

| | |
|---|----|
| Предисловие автора | 12 |
| Все это персонально | 13 |
| Используйте приложенные видеоролики | 14 |
| | |
| Часть I. Начальные знания | 16 |
| Что, если я ненавижу твое дурацкое персональное движение, Зед? | 18 |
| Что, если окажется, что у меня все плохо? | 18 |
| Упражнение 0. Настройка | 20 |
| Редактор программиста | 20 |
| Python 3.6 | 21 |
| Рабочий терминал | 21 |
| Рабочая конфигурация pip+virtualenv | 21 |
| Записная книжка | 22 |
| Учетная запись на Github.com | 22 |
| git | 22 |
| Дополнительно: программное обеспечение для записи с экрана | 22 |
| Дальнейшее обучение | 23 |
| Упражнение 1. Движение | 24 |
| Задача упражнения | 26 |
| Практические задания | 27 |
| Дальнейшее обучение | 28 |
| Упражнение 2. Креативность | 29 |
| Задача упражнения | 30 |
| Практическое задание | 31 |
| Упражнение 3. Качество | 32 |
| Задача упражнения | 34 |
| Практическое задание | 35 |
| | |
| Часть II. Быстрые задания | 36 |
| Как развивать креативность | 37 |
| Для начинающих программистов | 39 |
| Кодинг начинающего программиста | 40 |
| Упражнение 4. Аргументы командной строки | 41 |
| Задача упражнения | 42 |
| Решение | 43 |

| | |
|---|----|
| Практические задания..... | 43 |
| Упражнение 5. Команда <code>cat</code> | 44 |
| Задача упражнения..... | 44 |
| Решение..... | 45 |
| Практические задания..... | 46 |
| Дальнейшее обучение..... | 46 |
| Упражнение 6. Команда <code>find</code> | 47 |
| Задача упражнения..... | 48 |
| Практические задания..... | 49 |
| Дальнейшее обучение..... | 50 |
| Упражнение 7. Команда <code>grep</code> | 51 |
| Задача упражнения..... | 52 |
| Практические задания..... | 52 |
| Дальнейшее обучение..... | 53 |
| Упражнение 8. Команда <code>cut</code> | 54 |
| Задача упражнения..... | 55 |
| Практическое задание..... | 56 |
| Дальнейшее обучение..... | 56 |
| Упражнение 9. Команда <code>sed</code> | 57 |
| Задача упражнения..... | 58 |
| Практические задания..... | 59 |
| Дальнейшее обучение..... | 59 |
| Упражнение 10. Команда <code>sort</code> | 60 |
| Задача упражнения..... | 61 |
| Практические задания..... | 62 |
| Дальнейшее обучение..... | 62 |
| Упражнение 11. Команда <code>uniq</code> | 63 |
| Задача упражнения..... | 64 |
| Практические задания..... | 64 |
| Дальнейшее обучение..... | 64 |
| Упражнение 12. Обзор..... | 65 |
| Задача упражнения..... | 66 |
| Практические задания..... | 67 |
| Дальнейшее обучение..... | 67 |
| Часть III. Структуры данных..... | 68 |
| Обучение качеству с использованием структур данных..... | 70 |
| Как изучать структуры данных..... | 71 |
| Упражнение 13. Односвязные списки..... | 74 |
| Описание..... | 74 |
| Контроллер..... | 76 |

| | |
|--|-----|
| Тест | 78 |
| Вводный аудит | 80 |
| Задача упражнения | 82 |
| Аудит | 82 |
| Практическое задание | 82 |
| Упражнение 14. Двусвязные списки | 83 |
| Введение в инвариантные условия | 84 |
| Задача упражнения | 86 |
| Практическое задание | 87 |
| Упражнение 15. Стеки и очереди | 88 |
| Задача упражнения | 89 |
| Ломаем это | 90 |
| Дальнейшее обучение | 91 |
| Упражнение 16. Пузырьковая и быстрая сортировка, сортировка слиянием | 92 |
| Задача упражнения | 93 |
| Изучаем пузырьковую сортировку | 95 |
| Сортировка слиянием | 96 |
| Плутовство при сортировке слиянием | 97 |
| Быстрая сортировка | 99 |
| Практические задания | 100 |
| Упражнение 17. Словарь | 101 |
| Задача упражнения | 101 |
| «Мастер-копия» кода | 102 |
| Скопируйте код | 103 |
| Добавьте аннотации | 107 |
| Подведите итоги структуры данных | 107 |
| Запомните итоги | 108 |
| Реализуйте по памяти | 109 |
| Повторение | 110 |
| Практические задания | 110 |
| Ломаем это | 110 |
| Упражнение 18. Измерение производительности | 111 |
| Инструменты | 111 |
| Анализируем производительность | 115 |
| Задача упражнения | 117 |
| Практические задания | 118 |
| Ломаем это | 118 |
| Дальнейшее обучение | 118 |
| Упражнение 19. Повышение производительности | 119 |
| Задача упражнения | 121 |

| | |
|--|-----|
| Дальнейшее обучение | 122 |
| Упражнение 20. Двоичные деревья поиска | 123 |
| Особенности ДДП | 123 |
| Удаление | 125 |
| Задача упражнения | 125 |
| Практические задания | 126 |
| Упражнение 21. Двоичный поиск | 127 |
| Задача упражнения | 127 |
| Практические задания | 128 |
| Дальнейшее обучение | 128 |
| Упражнение 22. Суффиксные массивы | 129 |
| Задача упражнения | 130 |
| Практические задания | 131 |
| Дальнейшее обучение | 131 |
| Упражнение 23. Троичные деревья поиска | 132 |
| Задача упражнения | 132 |
| Практические задания | 135 |
| Упражнение 24. Быстрый поиск по URL | 136 |
| Задача упражнения | 136 |
| Практические задания | 138 |
| Дальнейшее обучение | 138 |
| | |
| Часть IV. Проекты следующего уровня | 139 |
| Отслеживание ошибок | 140 |
| Упражнение 25. Команда <code>xargs</code> | 141 |
| Задача упражнения | 141 |
| Практические задания | 142 |
| Упражнение 26. Команда <code>hexdump</code> | 143 |
| Задача упражнения | 144 |
| Практическое задание | 146 |
| Дальнейшее обучение | 146 |
| Упражнение 27. Команда <code>tr</code> | 147 |
| Задача упражнения | 148 |
| Критика 45-минутного подхода | 148 |
| Практические задания | 149 |
| Упражнение 28. Команда <code>sh</code> | 150 |
| Задача упражнения | 150 |
| Практическое задание | 151 |
| Дальнейшее обучение | 151 |
| Упражнение 29. Команды <code>diff</code> и <code>patch</code> | 152 |
| Задача упражнения | 152 |

| | |
|---|------------|
| Практическое задание..... | 154 |
| Дальнейшее обучение | 154 |
| Часть V. Анализ текста | 155 |
| Введение в покрытие кода | 156 |
| Упражнение 30. Конечные автоматы | 158 |
| Задача упражнения..... | 160 |
| Практические задания..... | 162 |
| Дальнейшее обучение | 163 |
| Упражнение 31. Регулярные выражения..... | 164 |
| Задача упражнения..... | 166 |
| Практические задания..... | 167 |
| Дальнейшее обучение..... | 167 |
| Упражнение 32. Лексические анализаторы | 168 |
| Небольшой лексический анализатор Pyny | 170 |
| Задача упражнения..... | 172 |
| Практические задания..... | 173 |
| Дальнейшее обучение..... | 173 |
| Упражнение 33. Синтаксические анализаторы | 175 |
| Синтаксический анализ методом рекурсивного спуска | 177 |
| Грамматика формы Бэкуса – Наура | 179 |
| Быстрый демонстрационный синтаксический анализатор ... | 181 |
| Задача упражнения..... | 184 |
| Практическое задание..... | 185 |
| Дальнейшее обучение..... | 185 |
| Упражнение 34. Семантические анализаторы | 186 |
| Шаблон Посетитель | 187 |
| Короткий семантический анализатор Pyny Python | 188 |
| Синтаксический анализатор против семантического анализатора..... | 192 |
| Задача упражнения..... | 192 |
| Практические задания..... | 193 |
| Дальнейшее обучение..... | 193 |
| Упражнение 35. Интерпретаторы | 194 |
| Интерпретаторы против компиляторов | 194 |
| Python – это и то и другое..... | 195 |
| Как написать интерпретатор | 196 |
| Задача упражнения | 197 |
| Практические задания | 197 |
| Дальнейшее обучение | 197 |
| Упражнение 36. Простой калькулятор | 198 |
| Задача упражнения | 198 |

| | |
|--|-----|
| Практические задания..... | 200 |
| Дальнейшее обучение..... | 200 |
| Упражнение 37. Немного Бейсика..... | 201 |
| Задача упражнения..... | 202 |
| Практические задания..... | 202 |
| Часть VI. SQL и объектно-реляционное отображение..... | 203 |
| Понимать SQL – это понимать таблицы..... | 204 |
| Что вы изучите..... | 205 |
| Упражнение 38. Введение в SQL..... | 207 |
| Что такое SQL?..... | 208 |
| Настройка..... | 209 |
| Изучаем словарь SQL..... | 210 |
| Грамматика SQL..... | 212 |
| Дальнейшее обучение..... | 212 |
| Упражнение 39. Создание в SQL..... | 213 |
| Создание таблиц..... | 213 |
| Создание многотабличной базы данных..... | 214 |
| Вставка данных..... | 215 |
| Вставка ссылочных данных..... | 216 |
| Задача упражнения..... | 216 |
| Дальнейшее обучение..... | 218 |
| Упражнение 40. Чтение в SQL..... | 218 |
| Выбор среди множества таблиц..... | 219 |
| Задача упражнения..... | 220 |
| Дальнейшее обучение..... | 221 |
| Упражнение 41. Обновление в SQL..... | 222 |
| Обновление комплексных данных..... | 223 |
| Замена данных..... | 223 |
| Задача упражнения..... | 224 |
| Дальнейшее обучение..... | 225 |
| Упражнение 42. Удаление в SQL..... | 226 |
| Удаление с использованием других таблиц..... | 227 |
| Задача упражнения..... | 228 |
| Дальнейшее обучение..... | 229 |
| Упражнение 43. Администрирование SQL..... | 230 |
| Уничтожение и изменение таблиц..... | 230 |
| Миграция и развитие данных..... | 232 |
| Задача упражнения..... | 233 |
| Дальнейшее обучение..... | 234 |
| Упражнение 44. Использование API баз данных Python..... | 235 |
| Изучение API..... | 235 |

| | |
|---|-----|
| Задача упражнения. | 236 |
| Дальнейшее обучение. | 237 |
| Упражнение 45. Создание объектно-реляционного менеджера. | 238 |
| Задача упражнения. | 238 |
| Дальнейшее обучение. | 239 |
| Часть VII. Финальные проекты. | 240 |
| Каково ваше движение? | 241 |
| Упражнение 46. Инструмент <code>blog</code> | 243 |
| Задача упражнения. | 243 |
| Практические задания. | 245 |
| Упражнение 47. Язык <code>bc</code> | 246 |
| Задача упражнения. | 246 |
| Практическое задание. | 247 |
| Упражнение 48. Команда <code>ed</code> | 248 |
| Задача упражнения. | 248 |
| Практические задания. | 249 |
| Упражнение 49. Команда <code>sed</code> | 250 |
| Задача упражнения. | 251 |
| Практическое задание. | 252 |
| Упражнение 50. Текстовый редактор <code>vi</code> | 253 |
| Задача упражнения. | 254 |
| Практические задания. | 254 |
| Упражнение 51. Создание веб-сервера (<code>lessweb</code>). | 255 |
| Задача упражнения. | 255 |
| Ломаем это. | 256 |
| Упражнение 52. Создание веб-сервера (<code>moreweb</code>). | 258 |
| Задача упражнения. | 258 |
| Ломаем это. | 259 |
| Дальнейшее обучение. | 260 |
| Предметный указатель. | 261 |

Предисловие автора

Движение, креативность и качество. Тщательно заучивайте эти три слова в процессе чтения данной книги. Движение. Креативность. Качество. В книге есть множество упражнений на изучение важных тем, в которых должен разбираться каждый программист, но настоящие знания заключаются в этих трех словах. При написании этой книги по программированию моей целью было обучить вас тому, что, в моем представлении, является тремя наиболее важными основами для любого программиста. Без движения вы будете блуждать, не зная, как приступить к работе, и у вас появятся проблемы с сохранением прогресса на длинных дистанциях. Без креативности вы не сможете решать проблемы, с которыми как программист будете сталкиваться каждый день. Без качества вы не поймете, хорошо ли вообще то, что вы делаете.

Обучить трем указанным принципам нетрудно. Я мог бы просто написать в блоге три поста и сказать: «Ну вот, теперь вы знаете, что означают эти три слова». Это не сделает вас лучше как программиста и, определенно, не сделает вас человеком, который смог бы самостоятельно заниматься разработкой в течение следующих 10 или 20 лет. Иметь представление о чем-то не подразумевает умение применять это на практике. Чтение поста о креативности не позволит узнать, насколько креативны вы сами. Для настоящего понимания этих сложных тем их необходимо усвоить, и лучший способ сделать это – применить их в простых проектах.

Когда вы приступите к выполнению упражнений в этой книге, я сообщу вам, над какой из тем вы будете работать. В этом отличие от других моих книг, где я «подло» обучаю вас принципам без вашего ведома. На сей раз я буду откровенен, ведь мне важно, чтобы вы держали принцип в уме и могли практиковать его во время упражнения. Затем вы оцените, насколько удачной была ваша попытка применить знание на практике, и подумаете, что предпринять, чтобы в следующий раз сделать ее более удачной. В этой книге ключевым моментом является способность рефлексировать над своими возможностями и самосовершенствоваться. Лучше всего это делать, сосредоточиваясь на одной технике или виде практики за раз, выполняя одновременно какие-либо другие задачи.

В дополнение к движению, креативности и качеству вы также узнаете, в каких шести важных вопросах, по моему мнению, должен разбираться современный программист. Эти темы могут измениться в будущем, но вот уже на протяжении десятилетий они оказывались совершенно необходимыми, и,

если не случится резкого изменения в технологиях, по-прежнему останутся актуальными. Даже что-то вроде языка SQL, о котором говорится в части VI по-прежнему применяется, поскольку учит вас тому, как структурировать данные, чтобы в дальнейшем они не разваливались на части. Ваши вторичные образовательные цели заключаются в следующем:

начало работы: вы научитесь быстро начинать проект;

структуры данных: я не рассматриваю каждую отдельную структуру данных, но подвожу вас к более полному их изучению;

алгоритмы: структуры данных бессмысленны без способа их обработки;

разбор (анализ) текста: это основа компьютерных наук, и умение его выполнять поможет вам в изучении языков программирования;

моделирование данных: я использую SQL, чтобы научить вас основам моделирования хранимых данных логичным способом;

инструменты Unix: инструменты командной строки используются на протяжении всей книги в качестве копируемых проектов; затем вы также изучите продвинутые инструменты командной строки Unix.

В каждой части книги внимание будет уделяться одному, двум или трем упражнениям за раз, пока, наконец, в завершающей части VII вы не примените все знания при создании простого веб-сайта. Ваши итоговые проекты не будут выглядеть привлекательно. Вы не узнаете, как создать свой следующий стартап. Но эти приятные небольшие проекты позволят вам использовать то, что вы узнаете при изучении Django.

Все это персонально

Множество книг учат этим трем принципам в контексте работы в команде. Когда в них идет речь о движении, все сводится к тому, что вы работаете над проектом по поддержанию кода с другим человеком. Когда речь идет о творчестве – к тому, как вы со своей командой отправляетесь на встречу, чтобы задавать вопросы клиентам. К сожалению, большая часть этих «профессиональных» книг по-настоящему не обучают качеству. В этом нет ничего страшного, но у многих новичков с таким «командным» подходом возникает две проблемы.

1. У вас нет команды, поэтому применять на практике то, что изучаете, вы не можете. Книги, ориентированные на командное взаимодействие, предназначены для младших программистов, у которых уже есть работа – и необходимость работать в новой команде. Пока вы не окажетесь в таком положении, любая подобная книга будет для вас бесполезна.
2. Какой смысл учиться работать в команде, если ваш собственный уровень движения, креативности и качества чрезвычайно низок? Что бы там ни говорили приверженцы «командной игры», подавляющее большинство программистских задач выполняются в одиночку, как и процедура вашей оценки собственных навыков. Если вы работаете в команде, но при этом пишете низкокачественный код, так что постоянно приходится просить членов команды о помощи, ваш босс не отзовется о вас хорошо. При всех разговорах о прелестях командного подхода, они никогда не винят команду, если программист не может работать в одиночку. Они винят самого программиста.

Эта книга не о том, как стать хорошим рабочим трутнем в ООО «Мега Компания». Ее предназначение – помочь вам улучшить свои навыки, чтобы вы смогли действовать в одиночку, когда получите работу. Если вы будете совершенствовать лично себя, тогда вы естественным образом станете сильнее и в командном смысле тоже. Это также значит, что вы сможете создавать и развивать собственные идеи, с чего и начинается подавляющее большинство проектов.

Используйте приложенные видеоролики

К «Легкому способу продолжить изучение *Python 3* еще глубже» прилагается обширный набор видеороликов, иллюстрирующих процесс работы кода, отладки, и, что важнее, решения к заданиям. Эти видео – лучшее место для демонстрации многих распространенных ошибок. Ошибки нарочно вносятся в код Python, чтобы затем можно было показать, как все исправить. Кроме того, я использую различные хитрости и приемы отладки и получения данных. На видео я показываю, как «перестать паяться и спросить» код, что не так. Эти видео можно просмотреть онлайн по адресу informit.com/title/9780134123486.

Зарегистрируйте свою копию «Легкого способа выучить *Python 3* еще глубже» на сайте InformIT, чтобы получать доступ к обновлениям и исправлениям по

мере их появления. Для начала процесса регистрации перейдите по адресу **informit.com/register**, затем войдите в систему или создайте учетную запись. Введите номер ISBN (9780134123486) и ответьте на простой вопрос для подтверждения покупки. Затем перейдите на вкладку «Зарегистрированные продукты» (Registered Products) и найдите ссылку на бонусный контент рядом с соответствующим продуктом. Перейдите по этой ссылке, чтобы получить доступ к бонусным материалам.

Часть I

Начальные знания

Первое, что вам необходимо выучить, – это все. Знаю, звучит пугающе, но, как я уже упомянул в предисловии, эта книга будет обучать вас лишь трем навыкам. В каждом упражнении вы станете подкреплять каждый из навыков при выполнении других задач. Я могу сказать вам: «сделать копию команды `cat`», — но что вы будете делать в действительности, так это развивать свою креативность. Я могу приказать «создать структуру данных связанного списка», но на самом деле вы примените процессы структурированного обзора кода в своей программистской практике. Секрет этой книги заключается в том, чтобы использовать проекты и упражнения как средства для изучения трех важных практик: движения, креативности и качества.

В идеале в этих трех концептах нет ничего сверхъестественного. Движение – это просто шаги, которые вы совершаете при создании чего-либо. Креативность – всего лишь способ создания и воплощения идей в жизнь. Используя качество, вы убеждаетесь, что эти воплощения не мусор. Самое же приятное в применении. Как вы примените движение к навыкам личностного развития? Как поймете, создали вы качественное программное обеспечение или

нет? Как возьмете идею и сделаете ее реальностью? Все три принципа взаимосвязаны, так как вам необходимо движение, чтобы развить креативность и обеспечить качество, что также требует креативности, поскольку движение не длится постоянно. Это красивый порочный круг.

Процесс обучения по этой книге таков.

1. В каждой части книги я буду давать вам цель работы над движением, креативностью или качеством. Обычно это будет две концепции за раз; иногда только одна. Например, в части II вы станете работать над креативностью, на протяжении 45 минут осуществляя простой взлом. Вы также проанализируете свое начальное движение, и если вам окажется трудно сдвинуться с места, значит, вы не были достаточно креативными.
2. В начале каждого упражнения вам будет даваться наводка или цель, над которой нужно подумать во время выполнения упражнения. Каждая из них предложит сосредоточиться на одном или нескольких аспектах вашей работы. Упражнение 4 в части II просто дает задачу реализовать что-то, а затем в упражнении 5 уже нужно перечислить тормозящие вас факторы и попытаться устранить их или уменьшить их влияние. Другие упражнения советуют осмотреть свое рабочее место и убрать все, что может вас отвлечь. В каждом упражнении вы станете размышлять над этими целями, а затем работать над упражнением, пытаясь сосредоточиться на конкретной задаче.
3. В конце каждого упражнения есть практические задания для дополнительной работы. Они могут быть непосредственно связаны с проектом или просто относиться к проблемам движения, креативности и качества, с которыми вы имеете дело в упражнении.
4. Некоторые упражнения имеют повышенный уровень сложности. Это означает, что вам дается описание инструмента для реализации, обычно основанного на существующем инструменте Unix, а затем предлагается реализовать его, но никакого кода не предоставляется. Возможно, сначала вам нужно будет изучить небольшие фрагменты образца кода, но, как правило, в этих упражнениях Python не используется. Решения доступны в интернете в проекте Git на Github по адресу bit.ly/lmpthwsolve.
5. Другие упражнения будут представлены в виде описаний чего-либо, что вам необходимо реализовать, основываясь на моем коде. В этих упражнениях будут объясняться определенные вещи (например,

алгоритм), а затем вам придется реализовать их как можно точнее и найти как можно больше ошибок. Обычно такие упражнения нацелены на повышение качества, так что вам предложат пройти автоматизированные тесты, отследить частоту ошибок и решить дополнительные задания.

6. Наконец, вы будете использовать записную книжку, чтобы делать заметки и отслеживать показатели, которые пригодятся вам для совершенствования своего способа работать. Крайне важно относиться к этому как к дневнику, личной учетной записи, содержащей историю вашего развития, которой вы ни с кем не должны делиться, в особенности с вашим менеджером. Такая информация может быть использована для извлечения выгоды из вас как работника, поэтому храните ее в надежном месте.

Ваша цель при чтении этой книги состоит не только в том, чтобы сделать пару копий некоторых инструментов Unix. Ваша цель – использовать эти небольшие проекты Unix, чтобы сосредоточиться на аспектах вашей способности работать над проектами более крупными.

Что, если я ненавижу твое дурацкое персональное движение, Зед?

Никаких проблем. Эта книга задумывалась как нечто, что позволит вам расти и совершенствоваться. Если вы не в полной мере готовы анализировать то, как вы работаете, тогда отложите книгу на время. Можете просто выполнить все задания самостоятельно когда хотите, а затем вернуться и приступить к проектам с ограничениями в собственном движении. Каждое упражнение автономно и обеспечивает персональное развитие. Делайте, что можете, и возвращайтесь, когда вам понадобится понять, как вы работаете.

Что, если окажется, что у меня все плохо?

Весьма вероятно, так и произойдет. Но мой метод заключается в том, чтобы помочь вам понять, почему у вас все плохо и что нужно сделать, чтобы это исправить. Тогда вопрос развития сведется лишь к объемам проделанной работы. Держите свой дневник в тайне, и никто не узнает, насколько у вас все плохо. Тогда совсем скоро вы узнаете, каков ваш уровень и на что нужно

обратить внимание. Больше никакого гадания, мошенник вы или действительно способны выполнять работу. Вы объективно оцените свои сильные и слабые стороны и перестанете беспокоиться о том, каково ваше место в этом мире.

Однако, скорее всего, вы не так ужасны, как думаете. Предназначение этой книги – быть личным курсом по улучшению объективной оценки своих навыков. Это означает, что вы должны сосредоточиться не на том, насколько вы хороши в чем-либо, но на том, насколько вы растете. Если ваш результат в конкретном упражнении вас расстраивает, нужно взять перерыв и подумать, что можно улучшить. Вы также должны взглянуть на это упражнение в контексте всех остальных, ранее выполненных, и сделать объективный вывод о том, насколько вы стали лучше. Сосредоточение внимания на улучшении помогает мыслить объективно (не положительно или отрицательно) и продолжать учиться.