

# Содержание

Об авторе	14
<b>Предисловие</b>	15
<b>Часть I. Приступаем к работе</b>	19
<b>Глава 1. Предварительный обзор</b>	20
“Программирование на Python”: новелла	20
Постановка задачи	21
Шаг 1: представление записей	21
Использование списков	21
Использование словарей	26
Шаг 2: постоянное хранение записей	31
Использование форматированных файлов	31
Использование файлов pickle	36
Использование файлов pickle для отдельных записей	39
Использование баз данных shelve	40
Шаг 3: переход к ООП	44
Использование классов	45
Добавление поведения	46
Добавление наследования	47
Рефакторинг кода	48
Добавление постоянства	51
Другие системы управления базами данных	53
Шаг 4: добавление взаимодействия с консолью	55
Консольный интерфейс для хранилища shelve	55
Шаг 5: добавление графического пользовательского интерфейса	58
Основы графических пользовательских интерфейсов	58
Использование ООП для графических пользовательских интерфейсов	60
Получение пользовательского ввода	63
Графический пользовательский интерфейс для хранилища shelve	64
Шаг 6: добавление веб-интерфейса	70
Основы CGI	71
Запуск веб-сервера	74
Использование строк запросов и urllib	77
Форматирование текста ответа	78
Веб-интерфейс для хранилища shelve	79
Конец демонстрации	88

<b>Часть II. Системное программирование</b>	91
<b>Глава 2. Системные инструменты</b>	92
“ <code>os.path</code> – дорога к знанию”	92
Зачем использовать Python?	92
Следующие пять глав	93
Обзор разработки системных сценариев	95
Системные модули Python	95
Источники документации по модулям	96
Постраничное отображение документации	97
Специальный сценарий для постраничного отображения	98
Основы использования строковых методов	99
Другие концепции, связанные со строками, в Python 3.X:	
Unicode и тип <code>bytes</code>	102
Основы операций с файлами	103
Два способа использования программ	103
Руководства по библиотеке Python	105
Коммерческие справочники	105
Введение в модуль <code>sys</code>	106
Платформы и версии	106
Путь поиска модулей	106
Таблица загруженных модулей	108
Детали исключений	109
Другие инструменты, экспортируемые модулем <code>sys</code>	110
Введение в модуль <code>os</code>	110
Инструменты в модуле <code>os</code>	110
Административные инструменты	112
Константы переносимости	112
Общие инструменты <code>os.path</code>	113
Запуск команд оболочки из сценариев	115
Другие инструменты, экспортируемые модулем <code>os</code>	121
<b>Глава 3. Контекст выполнения сценариев</b>	124
“Я хотел бы иметь аргумент”	124
Текущий рабочий каталог	125
Текущий рабочий каталог, файлы и пути импортирования	125
Текущий рабочий каталог и командные строки	127
Аргументы командной строки	127
Передача аргументов командной строки	128
Переменные среды оболочки	131
Получение переменных оболочки	131
Изменение переменных оболочки	132
Особенности переменных оболочки: родители, <code>putenv</code> и <code>getenv</code>	134
Стандартные потоки ввода-вывода	135
Перенаправление потоков в файлы и программы	136
Перенаправление потоков и взаимодействие с пользователем	141

Перенаправление потоков в объекты Python	145
Служебные классы <code>io.StringIO</code> и <code>io.BytesIO</code>	148
Захват потока <code>stderr</code>	149
Синтаксис перенаправления в вызовах <code>print</code>	150
Другие варианты перенаправления: еще раз о функции <code>os.popen</code> и модуле <code>subprocess</code>	150
<b>Глава 4. Инструменты для работы с файлами и каталогами</b>	156
“Очистите свой жесткий диск за пять простых шагов!”	156
Инструменты для работы с файлами	156
Модель файловых объектов в Python 3.X	157
Использование встроенных файловых объектов	159
Двоичные и текстовые файлы	168
Низкоуровневые инструменты для работы с файлами из модуля <code>os</code>	179
Сканеры файлов	184
Инструменты для работы с каталогами	188
Обход одного каталога	188
Обход дерева каталогов	193
Обработка имен файлов Unicode в Python 3.X: <code>listdir</code> , <code>walk</code> , <code>glob</code>	198
<b>Глава 5. Системные инструменты для параллельного выполнения</b>	201
“Объясните обезьянам, что им делать”	201
Ветвление процессов	203
Комбинация <code>fork/exec</code>	206
Потоки	211
Модуль <code>_thread</code>	214
Модуль <code>threading</code>	225
Модуль <code>queue</code>	230
Предварительный обзор: графические пользовательские интерфейсы и потоки	234
Дополнительные сведения о глобальной блокировке интерпретатора	237
Завершение программ	240
Завершение с помощью модуля <code>sys</code>	240
Завершение с помощью модуля <code>os</code>	242
Коды завершения команд оболочки	243
Код завершения процесса и совместно используемое состояние	246
Код завершения потока и совместно используемое состояние	247
Взаимодействие между процессами	249
Анонимные каналы	251
Именованные каналы ( <code>fifo</code> )	262
Сокеты: первый взгляд	265
Сигналы	269
Модуль <code>multiprocessing</code>	272

Для чего нужен пакет multiprocessing?	273
Основы: процессы и блокировки	275
Инструменты ПРС: каналы, общая память и очереди	278
Запуск независимых программ	284
И многое другое	286
Для чего нужен пакет multiprocessing? Заключение	288
Другие способы запуска программ	288
Вызовы <code>os.spawn</code>	288
Вызов <code>os.startfile</code> в Windows	292
Переносимая инфраструктура для запуска программ	294
Обзор других системных инструментов	299
<b>Глава 6. Завершенные системные программы</b>	<b>300</b>
“Интенсивный поиск”	300
Игра “Найди самый большой файл Python”	301
Сканирование каталога стандартной библиотеки	301
Сканирование дерева каталогов стандартной библиотеки	302
Сканирование пути поиска модулей	303
Сканирование всего компьютера	305
Вывод имен файлов в формате Unicode	309
Разделение и объединение файлов	312
Переносимое разделение файлов	312
Переносимое объединение файлов	315
Варианты использования	318
Генерация веб-страниц для перенаправления	321
Файл шаблона страниц	322
Сценарий генерации страниц	323
Сценарий регрессионного тестирования	325
Запуск средства прогона тестов	328
Поиск в деревьях каталогов	333
<code>grep</code> , <code>glob</code> и <code>find</code>	333
Создание собственного модуля <code>find</code>	334
Удаление файлов с байт-кодом	338
Инструмент поиска в дереве, реализованный на Python	341
Модуль <code>visitor</code> : усовершенствованный обход каталогов	343
Редактирование файлов в деревьях каталогов ( <code>visitor</code> )	347
Глобальная замена в деревьях каталогов ( <code>visitor</code> )	349
Подсчет строк исходного кода ( <code>visitor</code> )	351
Изменение кода в копиях с помощью классов ( <code>visitor</code> )	352
Другие примеры использования <code>visitor</code> (внешние)	354
Воспроизведение медиафайлов	355
Модуль <code>webbrowser</code>	359
Модуль <code>mimetypes</code>	360
Запуск сценария	363
Автоматизированные инструменты запуска программ (внешние)	363

<b>Часть III. Построение графических пользовательских интерфейсов</b>	365
<b>Глава 7. Графические пользовательские интерфейсы</b>	366
“Здесь за тобой присматривают”	366
Темы, связанные с созданием графических пользовательских интерфейсов	367
Запуск примеров	368
Возможности разработки GUI на Python	369
Обзор библиотеки tkinter	374
Практические преимущества tkinter	374
Документация tkinter	375
Расширения tkinter	375
Структура tkinter	378
Восхождение по кривой обучения GUI	379
Сценарий “Hello World” в четыре строки (или меньше)	379
Основы применения tkinter	380
Создание виджетов	381
Диспетчеры геометрии	382
Запуск программ с GUI	382
Альтернативные способы применения tkinter	384
Основы изменения размеров виджетов	385
Конфигурирование параметров виджета и заголовка окна	387
Еще одна версия в память о старых временах	389
Упаковка виджетов без их сохранения	389
Добавление кнопок и обработчиков обратного вызова	391
Еще раз об изменении размеров виджетов: растягивание	393
Добавление пользовательских обработчиков обратного вызова	394
Обработчики обратного вызова на основе лямбда-выражений	395
Отложенные вызовы с использованием лямбда-выражений и объектных ссылок	396
Проблемы с областью видимости обработчиков обратных вызовов	398
Обработчики обратного вызова на основе связанных методов	404
Обработчики обратного вызова на основе вызываемых классов	405
Другие протоколы обратного вызова в tkinter	406
Привязка событий	407
Добавление нескольких виджетов	408
Еще раз об изменении размеров виджета: обрезка	409
Прикрепление виджетов к фреймам	410
Компоновка: порядок упаковки и прикрепление к сторонам	411
Еще раз о параметрах <code>expand</code> и <code>fill</code> метода <code>pack</code>	412
Позиционирование с использованием привязки вместо растягивания	413
Настройка виджетов с помощью классов	414
Стандартизация поведения и внешнего вида	415
Создание многократно используемых компонентов GUI с помощью классов	417
Прикрепление классов компонентов	419

Расширение классов компонентов	420
Автономные контейнерные классы	421
Конец введения в tkinter	424
Сравнение Python/tkinter и Tcl/Tk	426
<b>Глава 8. Обзор tkinter, часть 1</b>	428
“Виджеты, гаджеты, графические интерфейсы и тому подобное...”	428
Темы, рассматриваемые в главе	428
Настройка внешнего вида виджетов	429
Окна верхнего уровня	433
Виджеты Toplevel и Tk	434
Протоколы окон верхнего уровня	435
Диалоговые окна	440
Стандартные (общие) диалоговые окна	440
Модуль dialog в старом стиле	452
Специальные диалоговые окна	453
Привязка к событиям	457
Другие события, привязываемые с помощью bind	461
Виджеты Message и Entry	463
Виджет Message	463
Виджет Entry	463
Компоновка форм для ввода	465
“Переменные” tkinter и альтернативные способы компоновки форм	468
Флажки, переключатели и ползунки	471
Флажки	471
Переключатели	475
Ползунки	481
Три способа выполнения кода GUI	485
Прикрепление к фреймам	485
Независимые окна	490
Выполнение программ	491
Изображения	498
Использование изображений с кнопками	501
Просмотр и обработка изображений с помощью PIL	504
Основы PIL	504
Отображение изображений других типов с помощью PIL	505
Создание миниатюр изображений с помощью PIL	509
<b>Глава 9. Обзор tkinter, часть 2</b>	518
“Меню дня: ветчина, ветчина и ветчина”	518
Меню	518
Меню окон верхнего уровня	519
Меню на основе виджетов Frame и Menubutton	523
Окна с меню и панелями инструментов	528
Виджеты Listbox и Scrollbar	533
Работа со списками выбора	534
Работа с полосами прокрутки	536

Компоновка полос прокрутки	537
Виджет Text	538
Работа с виджетом Text	541
Добавление операций редактирования текста	544
Unicode и виджет Text	550
Более сложные операции с виджетом Text и тегами	560
Виджет Canvas	561
Базовые операции с холстом	562
Работа с виджетом Canvas	562
Прокручиваемые холсты	566
Прокручиваемые холсты и миниатюры изображений	569
Использование событий холста	572
Сетки	576
Зачем нужны сетки?	576
Основы сеток: снова о формах ввода	577
Сравнение методов grid и pack	578
Комбинирование grid и pack	580
Реализация возможности растягивания виджетов, размещенных в сетке	582
Компоновка более крупных таблиц с помощью grid	586
Инструменты синхронизации, потоки и анимация	594
Использование потоков с графическими пользовательскими интерфейсами на базе tkinter	596
Использование метода after	598
Простые приемы анимации	601
Дополнительные темы, связанные с анимацией	606
Конец обзора	607
Другие виджеты и их параметры	607
<b>Глава 10. Приемы реализации графических пользовательских интерфейсов</b>	609
“Создание улучшенной мышеловки”	609
GuiMixin: общие подмешиваемые классы инструментов	610
Функции построения виджетов	610
Подмешиваемые служебные классы	612
GuiMaker: автоматизация создания меню и панелей инструментов	615
Протоколы подклассов	619
Классы GuiMaker	620
Код самотестирования GuiMaker	620
BigGui: демонстрационная клиентская программа	622
ShellGui: графические пользовательские интерфейсы для инструментов командной строки	625
Обобщенный графический интерфейс для инструментов оболочки	625
Классы инструментальных наборов для конкретных приложений	627
Добавление графических пользовательских интерфейсов к инструментам командной строки	629

GuiStreams: перенаправление потоков данных в виджеты	636
Использование перенаправления для сценариев упаковки	640
Динамическая перезагрузка обработчиков	641
Помещение в оболочки интерфейсов окон верхнего уровня	643
Графические пользовательские интерфейсы, потоки управления и очереди	647
Помещение данных в очереди	649
Помещение обратных вызовов в очереди	653
Другие способы добавления графического пользовательского интерфейса в код без GUI	660
Отображение окон GUI по требованию	661
Добавление графического пользовательского интерфейса как отдельной программы: сокет (повторная встреча)	663
Добавление графического пользовательского интерфейса как отдельной программы: каналы команд	668
Инструменты запуска PyDemos и PyGadgets	676
Панель запуска PyDemos (в основном внешняя)	677
Панель запуска PyGadgets	681
<b>Глава 11. Примеры реалистичных программ с графическим пользовательским интерфейсом</b>	685
“Python и открытый код”	685
Примеры в других главах	686
Стратегия, принятая в главе	687
PyEdit: программа/объект текстового редактора	687
Запуск PyEdit	688
Исходный код PyEdit	694
PyPhoto: программа просмотра и изменения размеров изображений	713
Запуск PyPhoto	714
Исходный код PyPhoto	717
PyView: программа просмотра изображений и примечаний	723
Запуск PyView	723
Исходный код PyView	727
PyDraw: программа рисования и перемещения графики	733
Запуск PyDraw	733
Исходный код PyDraw	734
PyClock: виджет аналоговых/цифровых часов	741
Краткий урок геометрии	741
Запуск PyClock	745
Исходный код PyClock	748
PyTicTacToe: виджет игры в крестики-нолики	754
Запуск PyTicTacToe	754
Исходный код PyTicTacToe (внешний)	756
Что дальше?	758
Предметный указатель	759