

Содержание

Об авторе	11
О рецензенте	12
Предисловие	13
Глава 1. Разработка геопрограмм на Python	27
Python	27
Python 3	29
Разработка геопространственных программ	30
Сферы применения геоприложений	33
Анализ геоданных	33
Визуализация геоданных	35
Создание геопространственных мэшапов	37
Последние достижения	38
Заключение	41
Глава 2. Геоинформационные системы	42
Ключевые понятия ГИС	42
Географическое положение	42
Расстояние	46
Единицы измерения	48
Картографические проекции	50
Системы координат	56
Геодезические датумы	59
Географические фигуры	60
Форматы данных ГИС	62
Работа с данными ГИС вручную	64
Получение данных	65
Инсталляция библиотеки GDAL	65
Заключение	74
Глава 3. Библиотеки Python для геопрограммирования	75
Чтение и запись геоданных	75
Пакет GDAL/OGR	75
Инсталляция пакета GDAL/OGR	76
Концепция библиотеки GDAL	76
Пример использования	82
Концепция библиотеки OGR	85

Пример использования.....	86
Документация по GDAL/OGR.....	88
Работа с проекциями.....	89
Библиотека ruroj.....	89
Инсталляция библиотеки.....	89
Концепция библиотеки.....	91
Пример использования.....	93
Документация.....	94
Геоанализ и геообработка.....	95
Библиотека Shapely.....	95
Инсталляция библиотеки.....	95
Концепция библиотеки.....	97
Пример использования.....	99
Документация.....	100
Визуализация геоданных.....	101
Библиотека Mapnik.....	101
Инсталляция библиотеки.....	102
Концепция библиотеки.....	103
Пример использования.....	105
Документация.....	107
Заключение.....	107
Глава 4. Источники геоданных.....	109
Источники геоданных в векторном формате.....	110
Геоданные проекта OpenStreetMap.....	110
База данных TIGER.....	113
Геоданные веб-сайта Natural Earth.....	117
Географическая база данных GSHHG.....	119
Набор данных границ стран мира.....	121
Источники геоданных в растровом формате.....	122
Геоданные проекта Landsat.....	123
Геоданные веб-сайта Natural Earth.....	127
Геоданные проекта GLOBE.....	130
Национальный набор данных рельефа.....	132
Источники геоданных других типов.....	136
База данных сервера географических названий GEOnet.....	136
Данные информационной системы географических названий США.....	138
Выбор источника геоданных.....	140
Заклучение.....	140
Глава 5. Решение задач с геоданными на Python.....	142
Необходимые условия.....	142
Общие задачи с использованием геоданных.....	143

Задача: вычисление ограничительной рамки для всех стран мира.....	143
Задача: вычисление границы между Таиландом и Мьянмой.....	145
Задача: анализ высот на основе цифровой карты местности.....	147
Смена датумов и проекций.....	153
Задача: смена проекции для совмещения файлов фигур с географическими и UTM-координатами.....	153
Задача: перевод из одного датума в другой для совмещения свежих данных TIGER со старыми.....	157
Выполнение геопространственных расчетов.....	160
Задача: идентификация национальных парков внутри и в окрестностях городских агломераций.....	161
Конвертирование и стандартизация единиц геометрии и расстояния.....	166
Задача: вычисление длины границы между Таиландом и Мьянмой.....	166
Задача: нахождение точки в 132.7 км к западу от г. Шошоун, шт. Калифорния.....	173
Упражнения.....	174
Заключение.....	176
Глава 6. Пространственные базы данных.....	177
СУБД с поддержкой пространственных данных.....	177
Пространственные индексы.....	178
Знакомство с PostGIS.....	181
Инсталляция СУБД PostgreSQL.....	183
Инсталляция расширения PostGIS.....	184
Установка адаптера psycopg2.....	185
Настройка СУБД.....	186
Создание учетной записи пользователя Postgres.....	186
Создание базы данных.....	187
Разрешение доступа к базе данных.....	187
Включить поддержку пространственных данных.....	187
Использование расширения PostGIS.....	187
Документация по PostGIS.....	191
Продвинутый функционал PostGIS.....	191
Наиболее успешные практические приемы.....	192
Рекомендуем: используйте базу данных для отслеживания пространственных привязок.....	192
Рекомендуем: используйте для данных приемлемую пространственную привязку.....	194
Рекомендуем: избегайте динамических трансформаций внутри запроса.....	196
Рекомендуем: не создавайте геометрии внутри запроса.....	197
Рекомендуем: грамотно используйте пространственные индексы.....	198
Рекомендуем: учитывайте пределы оптимизатора запросов своей СУБД.....	198
Заключение.....	201

Глава 7. Генерирование карт при помощи Python и библиотеки Mapnik	202
Введение в библиотеку Mapnik.....	202
Создание образца карты	209
Понятия библиотеки Mapnik.....	214
Источники данных	214
Правила, фильтры и стили	217
Символизаторы	220
Карты и слои.....	229
Визуализация карты	230
Заключение	232
Глава 8. Работа с пространственными данными	234
Описание приложения DISTAL	234
Проектирование и конструирование базы данных	238
Скачивание и импорт данных.....	242
Набор данных границ стран мира	242
Географическая база данных береговых линий GSHHG.....	243
Географические названия США	244
Географические названия остальных мест.....	246
Реализация приложения DISTAL.....	249
Сценарий «выбрать страну»	251
Сценарий «выбрать область»	253
Сценарий «показать результаты»	263
Использование приложения DISTAL	268
Заклучение	269
Глава 9. Совершенствование приложения DISTAL.....	270
Обработка линии антимериديана	270
Решение проблемы масштабирования	276
Производительность	280
Поиск проблемы.....	280
Улучшение производительности	282
Использование сегментов береговых линий.....	291
Анализ повышения производительности.....	292
Заклучение	293
Глава 10. Инструменты для разработки геопространственных веб-приложений	294
Инструментарий и методика для геопространственных веб-приложений	294
Веб-приложения	295
Веб-службы.....	300

Стэк «скользящей карты».....	305
Геопространственные веб-протоколы.....	306
Анализ трех конкретных инструментов	308
Протокол TMS.....	308
Библиотека OpenLayers.....	313
Модуль GeoDjango.....	317
Заключение	325

Глава 11. Собираем все вместе – полнофункциональная картографическая система

О системе ShapeEditor.....	326
Проектирование системы ShapeEditor.....	330
Импорт файла фигур.....	330
Выбор геообъекта.....	332
Правка геообъекта.....	334
Экспорт файла фигур.....	334
Необходимые компоненты	334
Настройка базы данных.....	335
Настройка проекта ShapeEditor.....	335
Определение приложений ShapeEditor	337
Создание общего приложения shared	337
Определение моделей данных.....	339
Объект Shapefile	339
Объект Attribute.....	340
Объект Feature	340
Объект AttributeValue.....	341
Файл models.py	341
Знакомство с подсистемой администрирования	344
Заключение	350

Глава 12. ShapeEditor – импорт и экспорт файлов фигур

Реализация режима просмотра списка файлов фигур	351
Импорт файлов фигур.....	355
Форма для импорта файлов фигур.....	355
Извлечение выгруженного файла фигур из архива	358
Импорт содержимого файла фигур.....	361
Очистка	368
Экспорт файлов фигур	369
Заключение	376

Глава 13. ShapeEditor – выбор и правка геообъектов

Выбор геообъекта для правки.....	378
Реализация сервера сборных цифровых карт.....	378

Отображение карты при помощи библиотеки OpenLayers	398
Перехват нажатий кнопкой мыши.....	404
Реализация режима просмотра «Найти геообъект»	406
Правка геообъектов.....	412
Добавление геообъектов.....	418
Удаление геообъектов.....	421
Удаление файлов фигур.....	423
Использование системы ShapeEditor	424
Дальнейшие усовершенствования и улучшения	424
Заключение	425
Глоссарий сокращений и основных терминов	427
Сокращения	427
Термины	430
Предметный указатель	436

Об авторе

Эрик Вестра уже больше 25 лет является профессиональным разработчиком программного обеспечения, который в течение последнего десятилетия работает почти исключительно с Python. Первоначальный интерес Эрика к проектированию графического интерфейса пользователя привел к разработке одной из самых продвинутых систем срочной курьерской доставки, используемых службами и компаниями курьерской доставки по всему миру. В последние годы Эрик участвует в разработке и реализации систем по подбору поставщиков для потребителей товаров и услуг по целому ряду регионов с широкой географией и с использованием разных систем обмена мгновенными сообщениями и платежных систем. Эта работа включает в себя создание геокодеров в реальном масштабе времени и режимов просмотра постоянно меняющихся данных географических карт. Эрик проживает в Новой Зеландии и работает на компании, расположенные по всему миру.

Кроме того, он является автором выпущенных в издательстве Packt книг *«Геоанализ на Python»* и *«Создание картографических приложений»* на основе QGIS, а также предстоящей публикации *«Модульное программирование на языке Python»*.

Я хотел бы поблагодарить Рут за ее великолепие, а моих дочерей – за их терпение. Без вас ничего из этого не было бы возможным.

О рецензенте

Лу Може получил образование в области информатики очень давно в Университете штата Мичиган, где учился использовать программное обеспечение для разработки циклотрона. Затем он в течение 34 лет проработал в компании IBM и потом продолжил карьеру, работая на несколько консалтинговых фирм, включая долгосрочное сотрудничество с железнодорожной отраслью индустрии. В настоящее время он консультирует компанию Keyhole Software, расположенную в Ливуд, шт. Канзас. Прошлой весной для этой компании он создал программный инструмент MockOlar для каркасного прототипирования на основе операций перетаскивания. Лу программирует на C++, Java и более новых языках и в настоящее время интересуется микросервисами, Docker, Node.js, NoSQL, геопространственными системами, функциональным программированием, мобильными и одностраничными веб-приложениями – любым новым языком или платформой. Лу иногда ведет блог, посвященный программным технологиям. Он является соавтором трех книг по информатике, написал два учебных руководства по XML для IBM DeveloperWorks и учебное руководство по LDAP для журнала WebSphere. Кроме того, Лу является соавтором нескольких сертификационных тестов по J2EE для IBM, а также рецензентом ряда других издателей.

Предисловие

С ростом популярности картографических веб-сайтов и пространственно ориентированных устройств и приложений методы разработки геопространственного программного обеспечения образовали быстро развивающуюся область информатики – геоинформатику. Будучи разработчиком на Python, вы не можете позволить себе отстать от новейших тенденций. В сегодняшнем информированном о местоположении мире каждый разработчик на Python может извлечь выгоду из понимания концепций и методики разработки геопространственных приложений.

Работа с геопространственными данными бывает непростой, потому что она связана с математическими моделями земной поверхности. Но поскольку Python – это мощный язык программирования с большим количеством высокоуровневых программных инструментов, он идеально подходит для разработки геопространственных приложений. Эта книга познакомит вас с инструментами языка Python, которые требуются для разработки геопространственных приложений. Она проведет вас по ключевым геопространственным понятиям, таким как географическое положение, расстояние, единицы измерения, картографические проекции, геодезические датумы и форматы геопространственных данных. Затем мы займемся изучением ряда программных библиотек Python и воспользуемся ими и общедоступными геопространственными данными для решения самых разнообразных задач. Книга предоставляет углубленный анализ методов хранения пространственных данных в базе данных и приемов использования пространственных баз данных в качестве инструментов для решения широкого диапазона геопространственных задач.

В ней подробно рассматриваются методы генерирования карт при помощи инструмента визуализации цифровых карт – программной библиотеки Mapnik. Кроме того, книга поможет вам создать высокотехнологичное геопространственное веб-приложение с функционалом редактирования карты на основе географического модуля GeoDjango для веб-платформы Django, программной библиотеки Mapnik и геопространственного расширения PostGIS для СУБД PostgreSQL. К концу книги вы научитесь интегрировать пространственный функционал в свои собственные приложения и создавать полнофункциональные картографические приложения с нуля.

Эта книга представляет собой практическое руководство, которое научит вас приемам получения доступа к геоданным, управления ими и их визуализации, используя широкий диапазон инструментов Python для разработки геоинформационных систем (ГИС).

О чем эта книга рассказывает

Глава 1 «Разработка геопрограмм на Python» предлагает обзор языка программирования Python и концепций, лежащих в основе процесса разработки геопро-

странственного приложения. Кроме того, будут затронуты основные прецеденты использования разработок геопространственных приложений, последние достижения и тенденции ближайшего будущего в данной области.

Глава 2 «Геоинформационные системы» посвящена ознакомлению с базовыми понятиями, включая географическое положение, расстояние, единицы измерения, картографические проекции, географические фигуры, геодезические датумы и форматы геоданных, и затем обсуждению процесса работы с геоданными в ручном режиме.

Глава 3 «Библиотеки Python для геопрограммирования» разбирает основные библиотеки Python, предназначенные для разработки геоприложений, в том числе функционал библиотек, способы их установки, важные понятия, необходимые для понимания работы библиотек, и способы их применения.

Глава 4 «Источники геоданных» посвящена исследованию главных общедоступных источников геоданных, характеристике имеющейся в распоряжении информации, используемым форматам данных и приемам импортирования данных после их скачивания.

Глава 5 «Решение задач с геоданными на Python» посвящена применению ранее представленных библиотек для выполнения различных задач с использованием геоданных, включая смену картографических проекций, импорт и экспорт данных, конвертирование и стандартизацию единиц геометрий и расстояний, выполнение геопространственных расчетов.

Глава 6 «Пространственные базы данных» вводит понятия, лежащие в основе пространственных баз данных, и затем подробно рассматривает расширение PostGIS для СУБД PostgreSQL с поддержкой пространственных данных, способы его установки и использования из программы на Python.

Глава 7 «Генерирование карт при помощи Python и библиотеки Mapnik» предлагает подробный анализ инструмента картографирования Mapnik и приемов его применения для генерирования разнообразных карт.

Глава 8 «Работа с пространственными данными» познакомит с процессом проектирования и реализации полнофункционального геоприложения под названием DISTAL, используя общедоступные геоданные, хранящиеся в пространственной базе данных.

Глава 9 «Совершенствование приложения DISTAL» посвящена развитию функционала приложения из предыдущей главы для решения ряда задач, связанных с удобством использования и производительностью.

Глава 10 «Инструменты для разработки геопространственных веб-приложений» исследует понятия платформ веб-приложений, веб-служб, библиотек пользовательского интерфейса на JavaScript и скользящих карт. Данная глава познакомит с рядом стандартных веб-протоколов, используемых в геопространственных приложениях, и закончится обзором инструментария, при помощи которого в трех заключительных главах книги будет создано полнофункциональное картографическое приложение.

Глава 11 «Собираем все вместе – полнофункциональная картографическая система» знакомит с полноценным и высокотехнологичным веб-приложением

ShapeEditor, созданным с использованием геопространственного расширения PostGIS, библиотеки Mapnik и географического модуля GeoDjango. Работа начинается с проектирования законченного приложения и затем продолжается созданием моделей базы данных приложения ShapeEditor.

Глава 12 «ShapeEditor – импорт и экспорт файлов фигур» продолжает реализацию системы ShapeEditor, концентрируясь на отображении списка импортированных файлов фигур и подпрограммах импорта и экспорта файлов фигур через веб-браузер.

Глава 13 «ShapeEditor – выбор и правка геообъектов» завершает реализацию системы ShapeEditor добавлением подпрограмм, которые предлагают пользователю функционал для выбора и редактирования геообъектов внутри импортированного файла фигур. Он включает в себя создание специального сервера сборных цифровых карт и использование картографической библиотеки OpenLayers на JavaScript для визуализации геоданных на экране компьютера и взаимодействия с ними.

Что требуется для этой книги

Третье издание данной книги было расширено с целью поддержки Python 3, хотя, если хотите, вы можете продолжить пользоваться Python 2. Вам также понадобятся следующие инструменты и библиотеки, которые следует скачать и установить; подробные инструкции даны в соответствующих разделах книги:

- программный пакет **GDAL/OGR**;
- динамическая библиотека **GEOS**;
- библиотека Python **Shapely**;
- динамическая библиотека **Proj**;
- библиотека Python **pyproj**;
- СУБД **PostgreSQL** + программа администрирования **pgAdmin III**;
- геопространственное расширение **PostGIS** для СУБД PostgreSQL;
- адаптер СУБД PostgreSQL **psycopg2** для Python;
- динамическая библиотека **Mapnik**;
- веб-платформа **Django**;
- географический модуль **GeoDjango** для веб-платформы Django;
- картографическая библиотека **OpenLayers** на JavaScript.

Для кого эта книга

Эта книга предназначена для опытных разработчиков на языке Python, которые хотели бы освоить свободно распространяемый инструментарий и методику разработки геопространственных приложений с целью создания своих собственных геоприложений либо интеграции геопространственной технологии в свои существующие программы на Python.

Условные обозначения

В этой книге вы найдете ряд текстовых стилей, которые выделяют различные виды информации. Вот некоторые примеры этих стилей и объяснение их значения.

Фрагменты программного кода в тексте показаны следующим образом: «Набор данных `gdal.Dataset` представляет файл, который содержит данные в растровом формате».

Блок кода выглядит следующим образом:

```
import pyproj

lat1, long1 = (37.8101274, -122.4104622)
lat2, long2 = (37.80237485, -122.405832766082)

geod = pyproj.Geod(ellps="WGS84")
angle1, angle2, distance = geod.inv(long1, lat1, long2, lat2)

print("Расстояние равно {:.2f} метров".format(distance))
```


Когда требуется привлечь ваше внимание к конкретной части блока кода, соответствующие строки или элементы выделяются полужирным шрифтом:

```
for value in values:
    if value != band.GetNoDataValue():
        try:
            histogram[value] += 1
        except KeyError:
            histogram[value] = 1
```


Любой ввод команды из командной строки или вывод результатов их вычисления оформляется следующим образом:

```
% python3 calcBoundingBoxes.py
Afghanistan (AFG) lat=29.4061..38.4721, long=60.5042..74.9157
Albania (ALB) lat=39.6447..42.6619, long=19.2825..21.0542
Algeria (DZA) lat=18.9764..37.0914, long=-8.6672..11.9865
```

Новые термины и важные слова показаны полужирным шрифтом. Слова, которые вы видите на экране, например в меню или диалоговых окнах, выглядят в тексте следующим образом: «Нажмите на гиперссылке **Download Domestic Names**, чтобы скачать национальные географические названия».

 Предупреждения или важные примечания появляются в этом поле.

 Подсказки и приемы появляются тут.

 Дополнения к тексту оригинала книги.

Отзывы читателей

Отзывы наших читателей всегда приветствуются. Сообщите нам, что вы думаете об этой книге – что вам понравилось, а что нет. Обратная связь с читателями для нас очень важна, поскольку она помогает нам формировать названия книг, из которых вы действительно получите максимум полезного.

Отзыв по общим вопросам можно отправить по адресу feedback@packtpub.com, упомянув заголовок книги в теме вашего электронного сообщения.

Если речь о теме, в которой у вас есть экспертные знания, и вы интересуетесь написанием либо содействием в написании книги, то обратитесь к нашему перечню авторов по адресу www.packtpub.com/authors.

Служба поддержки

Теперь, когда вы являетесь довольным владельцем книги издательства Packt, мы предложим вам ряд возможностей с целью помочь вам получать максимум от своей покупки.

Скачивание исходного кода примеров

Вы можете скачать файлы с кодом примеров из вашего аккаунта по адресу <http://www.packtpub.com/> для всех книг издательства Packt Publishing, которые вы приобрели. Если вы купили эту книгу в другом месте, то можно посетить <http://www.packtpub.com/support> и зарегистрироваться там, чтобы получить файлы прямо по электронной почте.

Загрузить файлы с примерами программного кода можно, выполнив следующие шаги:

1. Войдите на наш веб-сайт или зарегистрируйтесь там, используя ваш адрес электронной почты и пароль.
2. Наведите указатель мыши на вкладку **SUPPORT** вверху страницы.
3. Щелкните по разделу **Code Downloads & Errata**, посвященному примерам программного кода и опечаткам.
4. Введите название книги в **Поле поиска**.
5. Выберите книгу, для которой вы хотели бы скачать файлы с примерами.
6. Из раскрывающего меню выберите место, где вы купили эту книгу.
7. Щелкните по **Code Download**, чтобы скачать примеры.

Кроме того, файлы примеров можно скачать, нажав на кнопку **Code Files** на странице книги на веб-сайте издательства Packt Publishing. Доступ к этой странице можно получить, введя наименование книги в **Поле поиска**. Обращаем внимание, что для этого вы должны войти в ваш аккаунт в издательстве Packt.

Скачав файл, пожалуйста, убедитесь, что вы разархивировали или извлекли папку, воспользовавшись последней версией указанных ниже архиваторов:

- WinRAR / 7-Zip для Windows;
- Zipag / iZip / UnRarX для Mac;
- 7-Zip / PeaZip для Linux.

Помимо этого, пакет примеров программного кода, прилагаемый к данной книге, размещен на GitHub по адресу <https://github.com/PacktPublishing/Python-Geospatial-Development-Third-Edition>. Мы также располагаем другими пакетами примеров программного кода, которые можно выбрать из нашего богатого каталога книг и видео, предлагаемого на странице <https://github.com/PacktPublishing/>. Можете убедиться сами!

Опечатки

Несмотря на то что мы приняли все меры, чтобы гарантировать корректность нашего информационного материала, ошибки действительно случаются. Если вы найдете ошибку в одной из наших книг, возможно, в тексте или коде, то мы будем благодарны, если вы сообщите об этом нам. Поступая так, вы можете уберечь других читателей от разочарования и помочь нам улучшать последующие версии этой книги. Если вы найдете какие-либо опечатки, пожалуйста, сообщите о них, посетив страницу <http://www.packtpub.com/submit-errata>, выбрав вашу книгу, нажав на ссылку формы **errata submission form** для предоставления информации об опечатке и введя ваши данные об опечатках. Как только ваши данные будут проверены, сообщение будет принято к рассмотрению, и данные об опечатке будут загружены на наш веб-сайт или добавлены к любому списку из существующих списков опечаток в разделе Errata под соответствующим заголовком.

Чтобы просмотреть ранее предоставленные опечатки, перейдите на страницу <https://www.packtpub.com/books/content/support> и в поле поиска введите название книги. Запрошенная информация появится под разделом Errata.

Нарушение авторских прав

Пиратство защищенного авторским правом материала в Интернете является хронической проблемой во всех средствах массовой информации. В издательстве Packt мы очень серьезно относимся к защите нашего авторского права и лицензий. Если вы сталкиваетесь с какими-либо недопустимыми копиями наших работ в какой-либо форме в Интернете, пожалуйста, просим вас незамедлительно предоставить нам адрес размещения или название веб-сайта, чтобы мы могли добиваться правовой защиты.

Пожалуйста, свяжитесь с нами по электронному адресу copyright@packtpub.com со ссылкой на предполагаемый пиратский материал.

Мы ценим вашу помощь в защите наших авторов и в наших усилиях предоставлять вам ценный информационный материал.

Вопросы

Если у вас есть вопрос по каким-либо аспектам этой книги, то вы можете связаться с нами по электронному адресу questions@packtpub.com, и мы приложим все усилия, чтобы решить ваш вопрос.

Комментарий переводчика

Весь материал книги приведен в соответствие с последними действующими версиями библиотек (время перевода книги – август-сентябрь 2016 г.), дополнен свежей информацией и протестирован в среде Windows 10 и Fedora 24. При тестировании программного кода за основу взят Python версии 3.5.2.

Книга содержит много аббревиатур и технических терминов из разных областей науки. Для удобства большинство аббревиатур кратко определено в сносках, а для некоторых терминов в силу отсутствия единой терминологии приведены соответствующие варианты наименований или пояснения. В конце книги основные термины и аббревиатуры собраны в *гlossarii*.

Поскольку в оригинале книги и, главное, во всех инструментальных средах программирования для отделения целой части числа от дробной используется не запятая, а точка, и чтобы не вносить путаницу при работе с книгой, в переводе оставлена форма записи чисел как есть, без изменений.

Книга может быть интересной широкому кругу специалистов, в том числе начинающим аналитикам данных, преподавателям, студентам, а также всем, кто интересуется геопрограммированием.

Далее приведены особенности установки и работы некоторого используемого программного обеспечения.

Установка и настройка дистрибутива Anaconda

Anaconda — это полностью свободный дистрибутив Python (предназначенный в том числе для коммерческого использования и повторного распространения). Он содержит более 400 самых популярных библиотек Python для вычислений в области естественных наук, математики, инженерии и анализа данных.

Установка дистрибутива в Windows выполняется стандартным образом после загрузки с сайта бинарного файла дистрибутива. В Linux сначала необходимо скачать установщик — скриптовый файл с расширением `.sh` для оболочки Bash (https://www.continuum.io/downloads#_unix). На примере Anaconda 4.2.0 (с Python 3.5.2) для 64-разрядных операционных систем команда установки выглядит так: `bash Anaconda3-4.2.0-Linux-x86_64.sh`.

Отметим, что инсталляция вступит в силу после того, как вы закроете и заново откроете окно терминала.

Для обновления дистрибутива Anaconda следует набрать в терминале следующую команду:

```
conda update conda
```

Удаление дистрибутива Anaconda:

```
rm -rf ~/anaconda
```

Более подробная информация по деинсталляции Anaconda содержится по указанной выше ссылке.

Чтобы удостовериться, что дистрибутив Anaconda установлен успешно, воспользуйтесь следующей командой:

```
conda --version
```

В результате будет выведен номер установленной версии.

В случае если вы в основном работаете в среде Python 3, но иногда возникает необходимость переключиться в среду Python 2 и использовать ее для работы с библиотеками, которые предназначены только для Python 2, то Anaconda предлагает функционал создания и активации новой среды, куда можно установить нужную версию языка. В нижеследующей таблице приведено несколько команд, которые можно выполнить прямо в Spyder во встроенном окне командной строки (**Инструменты** ⇨ **Открыть командную строку**):

conda create -n py27 python=2.7.12 anaconda	Установить другую версию Python (2.7.12) в новую среду с именем py27 (имя может быть любым)
conda info --envs	Проверить, что среда с именем py27 установлена (команда выводит список всех сред, при этом активная среда выделяется знаком *)
source activate py27 (Linux, OSX), activate py27 (Windows)	Переключиться в среду py27 с другой версией Python (команда activate добавляет в начало строки путь к среде py27)
deactivate	Деактивировать текущую среду и вернуться к среде по умолчанию
python --version	Проверить, что среда использует нужную версию Python
conda remove -n py27 --all	Удалить среду py27 (после выполнения команды выполнить conda clean --lock)
conda install --name py27 scipy	Установить библиотеку scipy в среду py27
conda remove --name py27 scipy	Удалить библиотеку scipy в среде py27
conda clean --lock	Очистить блокировку, если произошел сбой при установке среды (иногда сперва требуется удалить conda в Диспетчере задач)

Установка и настройка инструментальной среды Spyder

Spyder (в октябре 2016 г. появилась версия 3.0.0 с локализованным на русский язык графическим интерфейсом) — это инструментальная среда для научных вычислений для языка Python (Scientific PYthon Development EnviRonment) для Windows, Mac OS X и Linux. Это простая, легковесная и бесплатная интерактив-

ная среда разработки на Python, которая предлагает функционал, аналогичный среде разработки на MATLAB, включая готовые к использованию виджеты PyQt5 и PySide: редактор исходного кода, редактор массивов данных NumPy, редактор словарей, консоли Python и IPython и многое другое.

Для пользователей Windows хорошая новость заключается в том, что Spyder уже включен в состав дистрибутива Anaconda Python, и исполнимый файл находится в папке C:\Users\[ИМЯ_ПОЛЬЗОВАТЕЛЯ]\Anaconda3\Scripts\spyder.exe. Чтобы установить среду Spyder без установки Anaconda, необходимо скачать соответствующий файл whl (<https://pypi.python.org/pypi/spyder>) и установить, как объяснено в разделе *Установка библиотек Python из whl-файла* ниже. Чтобы установить среду Spyder в Ubuntu Linux, используя официальный менеджер пакетов, нужна всего одна команда:

```
sudo apt-get install spyder3
```

Чтобы установить с использованием менеджера пакетов pip:

```
sudo apt-get install python-qt5 python-sphinx
sudo pip3 install spyder
```

И чтобы обновить:

```
sudo pip3 install -U spyder
```

Установка среды Spyder в Fedora 24 приведена ниже в разделе *«Подготовка среды Python 3 в ОС Fedora 24»*. Во всех вышеперечисленных случаях речь идет о версии Spyder для Python 3 (на момент инсталляции это был Python 3.5.2). Чтобы установить версию Spyder для Python 2, нужно просто поменять spyder3 на spyder.

Настройка среды Spyder с Python 3 для работы с Python 2

При инсталляции дистрибутива Anaconda3 по умолчанию базовым является интерпретатор Python 3 (в данном случае версии 3.5.2). В случае если нужно на время переключиться в режим работы в интерпретаторе Python версии 2 (в данном случае версии 2.7.12), существуют два варианта: дополнительно установить дистрибутив Anaconda для Python 2 либо выполнить небольшую настройку инструментальной среды Spyder, по умолчанию работающей на основе Python 3.

Для такой настройки нужно в основном меню выбрать **Инструменты** и затем **Параметры**. В открывшемся окне **Параметры** выбрать **Интерпретатор Python**. В разделе **Интерпретатор Python** с двумя переключателями установить переключатель **Использовать следующий интерпретатор Python** и затем нажать кнопку напротив текстового поля, чтобы выбрать путь к интерпретатору Python 2.7.12, который находится в папке C:\Users\[имя_пользователя]\Anaconda3\envs\py27, где py27 — это имя среды, созданной вами для Python 2.7.12 (см. раздел *«Настройка дистрибутива Anaconda»* ранее). Если открыть новую консоль (**Консоли** ⇌ **Открыть консоль Python**), то в строке приветствия будет указана нужная версия Python.

Теперь в этой консоли можно набирать команды и вставлять целые программы, однако запускать программы из рабочего окна редактора Spyder не получится. Для

сценария, работающего с Python 2, требуется еще одна, и последняя, настройка. В основном меню нужно выбрать **Запуск**, затем **Настроить** и в разделе **Консоль** выбрать второй переключатель **Выполнить во внешнем системном терминале** (то есть в новой специально выделенной консоли). Теперь этот конкретный сценарий будет выполняться в консоли с Python 2.

Закончив работу с интерпретатором Python 2.7.12, следует не забыть вернуться к исходному интерпретатору Python 3.5.2. Для этого нужно в окне **Интерпретатор Python** просто установить переключатель **По умолчанию** (тот же, что и для Spyder).

Напомним, что месторасположение базового интерпретатора следующее: C:\Users\[ИМЯ_ПОЛЬЗОВАТЕЛЯ]\Anaconda3\python.exe. Разумеется, такой режим работы вносит некоторые неудобства, но, по крайней мере, он не такой громоздкий, как установка еще одной версии Anaconda 4.2.0, но уже с Python 2.7.12 в качестве базового интерпретатора.

Установка библиотек Python из whl-файла

Библиотеки для Python можно разрабатывать не только на чистом Python. Довольно часто библиотеки пишутся на C (динамические библиотеки), и для них пишется обертка Python, или же библиотека пишется на Python, но для оптимизации узких мест часть кода пишется на C. Такие библиотеки получаются очень быстрыми, однако библиотеки с вкраплениями кода на C программисту на Python тяжелее установить ввиду банального отсутствия соответствующих знаний либо необходимых компонентов и настроек в рабочей среде (в особенности в Windows). Для решения описанных проблем разработан специальный формат (файлы с расширением .whl) для распространения библиотек, который содержит заранее скомпилированную версию библиотеки со всеми ее зависимостями. Формат whl поддерживается всеми основными платформами (Mac OS X, Linux, Windows).

Установка производится с помощью менеджера пакетов pip. В отличие от обычной установки командой `pip3 install <имя_пакета>`, вместо имени пакета указывается путь к whl-файлу `pip3 install <путь/к/whl_файлу>`. Например:

```
pip3 install C:\temp\networkx-1.11-py2.py3-none-any.whl
```

Откройте окно командной строки и при помощи команды `cd` перейдите в каталог, где размещен ваш whl-файл. В Anaconda по умолчанию подобные файлы лежат в каталоге Scripts. Просто скопируйте туда ваш whl-файл. В этих случаях полный путь указывать не понадобится. Например:

```
pip3 install networkx-1.11-py2.py3-none-any.whl
```

При выборе пакета важно, чтобы разрядность устанавливаемой библиотеки и разрядность интерпретатора совпадали. Пользователи Windows могут брать whl-файлы с сайта <http://www.lfd.uci.edu/~gohlke/pythonlibs/>. Библиотеки там постоянно обновляются и в архиве содержатся все, какие только могут понадобиться.

Подготовка среды Python 3 в ОС Fedora 24

Действие	Команда
Обновление ОС Fedora 24 (http://losst.ru/nastrojka-fedora-24-posle-ustanovki)	\$ dnf update \$ dnf upgrade
Интерпретатор языка Python 3 и менеджер пакетов	\$ dnf install python3 \$ dnf -y install python3-tools \$ pip3 install --upgrade pip
Оболочка Python IDLE3	\$ dnf install idle3-tools
Основные научные библиотеки	\$ dnf install python3-numpy python3-matplotlib python3-scipy python3-pandas python3-sympy python3-scikit-learn
Записные книжки Jupyter	\$ pip3 install jupyter --upgrade
Инструментальная среда Spyder (файл whl в каталоге пакетов Python https://pypi.python.org/pypi/spyder)	\$ dnf install python3-spyder или \$ dnf install PyQt5 (pip3 install PyQt5) \$ sudo dnf install python3-devel \$ pip3 install psutil \$ sudo pip3 install /путь/к/файлу/spyder-3.0.0-ру3-none-any.whl
СУБД PostgreSQL, геопространственного расширения PostGIS, адаптера базы данных psycopg2 и администратора БД pgadmin3	\$ sudo dnf list postgresql* \$ sudo dnf install postgresql postgresql-contrib postgresql-server postgis \$ sudo dnf install python3-psycopg2 \$ sudo dnf install pgadmin3

Установка геопространственных библиотек в ОС Fedora 24

Библиотека GDAL

Скачать файл RPM из каталога библиотек RPM <https://www.rpmfind.net/linux/rpm2html/search.php?query=gdal-python3>:

```
$ dnf install /путь/к/файлу/gdal-python3-2.1.1-1.fc26.x86_64.rpm
```

или

```
$ dnf install gdal gdal-devel
```

Библиотека pyproj

Скачать файл RPM из каталога библиотек RPM <https://www.rpmfind.net/linux/rpm2html/search.php?query=python3-pyproj&submit=Search+...&system=&arch=>:

```
$ dnf install /путь/к/файлу/python3-pyproj-1.9.5.1-2.fc24.x86_64.rpm
```

Библиотека Shapely

Скачать файл RPM из каталога библиотек RPM <https://www.rpmfind.net/linux/rpm2html/search.php?query=python3-shapely&submit=Search+...&system=&arch=>:

```
$ dnf install /путь/к/файлу/python3-shapely-1.5.16-1.fc24.x86_64.rpm
```

Библиотека *Mapnik*

Автор книги будет использовать библиотеку *Mapnik* в главах 3, 7, 8, 9 и 13. Стоит отметить, что версия 3 библиотеки *Mapnik*, являясь мощным средством картографирования, пока работает только в Linux и Mac OS X, и разработка бинарников для Windows стоит в ближайших планах разработчиков. В связи с этим примеры программ в этих главах тестировались в ОС Fedora 24. Кроме того, предыдущие издания этой книги вышли в 2010 и 2013 гг., и с тех пор парадигма написания кода с использованием библиотеки *Mapnik* поменялась с процедурного на декларативный на основе XML и Geojson. Тем не менее в книге авторские примеры программного кода оставлены без изменений, а альтернативные реализации на XML приведены в комментариях и в прилагаемых к книге примерах.

Скачать файл RPM из каталога библиотек RPM <https://www.rpmfind.net/linux/rpm2html/search.php?query=mapnik&submit=Search+...&system=&arch=>:

```
$ dnf install /путь/к/файлу/mapnik-3.0.10-3.fc24.x86_64.rpm
$ dnf install /путь/к/файлу/python3-mapnik-0.1-7.20160202git1cb6851.fc24.x86_64.rpm
```

Запись справочной информации по библиотеке *mapnik* в HTML-файл:

```
$ python3
>>> import pydoc
>>> import mapnik
>>> pydoc.writedoc("mapnik")
>>> exit()
```

Протокол настройки СУБД PostgreSQL в ОС Fedora 24

```
[root@localhost ~]$ sudo postgresql-setup --initdb --unit postgresql
[root@localhost ~]$ sudo systemctl start postgresql
[root@localhost ~]$ sudo -i -u postgres
-bash-4.3$ psql
psql (9.5.4)
Введите "help", чтобы получить справку.
postgres=# alter user postgres password 'geo';
ALTER ROLE
postgres=# \c
Вы подключены к базе данных "postgres" как пользователь "postgres".
postgres=# create table test (id int);
CREATE TABLE
postgres=# \d
                Список отношений
  Схема | Имя | Тип | Владелец
-----+-----+-----+-----
 public | test | таблица | postgres
(1 строка)
postgres=# drop table test;
DROP TABLE
postgres=# \d
```

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru