

Содержание

Осваиваем язык Julia	12
Об авторе	13
О рецензентах.....	14
Предисловие	15
О чем рассказывает эта книга.....	16
Что вам потребуется для работы	17
Для кого эта книга	17
Соглашения.....	18
Обратная связь.....	19
Скачивание исходного кода программ	19
Ошибки и опечатки	19
Нарушение авторских прав	19
Вопросы	20
Readme от автора.....	20
Комментарий переводчика	20
Выполнение примеров программного кода на Julia.....	21
Установка среды разработки Julia	22
Установка пакетов в среду разработки Julia	23
Установка и удаление IDE Juno/Atom	25
Установка и работа с плагином для Eclipse.....	25
Работа с Julia в облаке JuliaBox.....	26
Установка и удаление дистрибутива Anaconda Python	26
Запуск сервера записных книжек Jupyter	27
Факультативные настройки среды	27
Среда разработки Julia	29
Введение.....	29
Философия.....	30
Роль в науке о данных и в области больших данных	31
Сопоставление с другими языками	32
Характеристики.....	34
Начало работы.....	35
Исходные тексты Julia.....	36
Сборка из исходников	37
Изучение стека исходного кода	41
Интегрированная среда разработки Juno	42

Плагин JuliaDT для среды программирования Eclipse	43
Среда программирования IJulia.....	44
Краткий обзор языка Julia	46
Julia через консоль	47
Установка некоторых пакетов.....	50
Мои эталонные испытания.....	52
Управление пакетами	53
Вывод перечня пакетов, добавление и удаление.....	53
Выбор и ознакомление с пакетами	54
Как деинсталлировать Julia	58
Добавление незарегистрированного пакета	59
Особенности языка Julia.....	59
Параллельная обработка	59
Множественная диспетчеризация	60
Гомоиконные макрокоманды	60
Межязыковое взаимодействие.....	60
Заключение	61
Разработка на Julia	62
Целые числа, биты, байты и логические значения.....	62
Целое	63
Логические и арифметические операторы.....	65
Логический тип.....	65
Массивы	66
Операции на матрицах.....	68
Поэлементные операции	68
Простая марковская цепь – кошки и мышки.....	69
Символьный и строковый типы	70
Символьный тип	70
Строковый тип.....	71
Пример: игра «Быки и коровы»	74
Вещественные, комплексные и рациональные числа	76
Вещественные	77
Рациональные числа	78
Комплексные числа.....	78
Составные типы.....	82
Дополнительно о матрицах	82
Векторизованный и девекторизованный программный код	83
Многомерные массивы	83
Разреженные матрицы	85
Массивы и таблицы данных.....	86

Словари, множества и прочее	87
Словари	87
Множества	89
Другие структуры данных	90
Заключение	91
Типы и диспетчеризация	92
Функции	92
Объекты первого класса	92
Передача аргументов	96
Область видимости	100
Задача о восьми ферзях	103
Система типов Julia	104
Обзор рационального типа	104
Тип данных для транспортных средств	107
Перечисляемый тип	116
Множественная диспетчеризация	118
Параметрические типы	119
Операции преобразования и приведения	121
Модуль для фиксированного 3D-вектора	122
Заклучение	124
Функциональная совместимость	125
Взаимодействие с другими программными средами	125
Обращение к программам на C и Fortran	126
Язык Python	131
Некоторые другие языки	133
Программный интерфейс Julia	133
Обращение к API из C	134
Метапрограммирование	136
Символические имена	136
Макрокоманды	138
Тестирование	141
Макрокоманда <code>enum</code>	143
Объект Task и многозадачность	145
Параллельные операции	146
Распределенные массивы	148
Простая модель MapReduce	151
Исполнение команд	152
Запуск команд на исполнение	153
Работа с файловой системой	155

Перенаправление ввода-вывода и конвейеры	157
Заключение	160
Работа с данными	161
Базовая система ввода-вывода	161
Терминальный ввод-вывод	161
Дисковые файлы	163
Обработка текста	165
Двоичные файлы	167
Структурированные наборы данных	169
Файлы CSV и DLM	169
Файлы HDF5	173
Файлы XML	176
DataFrames и RDatasets	179
Пакет DataFrames	179
Таблицы данных DataFrame	180
Пакет RDatasets	183
Статистика	187
Простые статистики	188
Выборки и их оценка	190
Модуль Pandas	190
Отдельные темы	192
Временные ряды	192
Распределения вероятностей	195
Проверка статистических гипотез	197
Обобщенная линейная модель	199
Заключение	202
Научное программирование	203
Линейная алгебра	204
Система уравнений	204
Разложение матриц	206
Собственные значения и собственные векторы	207
Матрицы специального вида	210
Обработка сигналов	211
Частотный анализ	211
Фильтрация и сглаживание	211
Цифровые фильтры	214
Обработка изображений	214
Дифференциальные уравнения	217
Решение обыкновенных дифференциальных уравнений	217

Нелинейные обыкновенные дифференциальные уравнения	219
Дифференциальные уравнения в частных производных	222
Оптимизационные задачи	224
Пакет JuMP	225
Пакет Optim	227
Пакет Nlopt	229
Стохастические задачи	233
Стохастическое моделирование	233
Байесовские методы и марковские процессы	237
Заключение	243
Графика	244
Базовая графика в Julia	245
Построение текстовых графиков	245
Пакет Cairo	246
Пакет Winston	248
Визуализация данных	251
Пакет Gadfly	252
Пакет Compose	257
Графические движки	259
Пакет PyPlot	259
Пакет Gaston	262
Пакет PGFPlots	264
Использование сети Интернет	267
Пакет Bokeh	267
Пакет Plotly	268
Растровая графика	271
Возвращаясь к пакету Cairo	272
Возвращаясь к пакету Winston	273
Пакеты Images и ImageView	274
Заклучение	276
Базы данных	278
Общий обзор баз данных	278
Вот две таблетки: красная и синяя. Выбирай!	279
Взаимодействие с базами данных	280
Другие соображения	282
Реляционные базы данных	283
Создание и загрузка	283
Нативные интерфейсы	286
Программный интерфейс ODBC	289

Другие методы взаимодействия.....	293
Интерфейс DBI.....	294
Пакет PyCall	296
Стандарт взаимодействия JDBC	298
Хранилища данных NoSQL.....	299
Системы «ключ-значение»	300
Документоориентированные хранилища данных	303
Взаимодействие с RESTful.....	306
Данные в формате JSON.....	307
Интернет-СУБД.....	308
Графовые системы хранения данных	311
Заключение	314
Сетевое взаимодействие	315
Сокеты и серверы.....	315
Стандартные порты	315
Сокеты UDP и TCP в Julia	316
«Зазеркальный» эхо-сервер.....	317
Именованные каналы	320
Работа в сети Интернет.....	321
Веб-служба на основе TCP	321
Группа пакетов JuliaWeb	323
Сервер цитат	326
Технология WebSocket	328
Обмен сообщениями.....	332
Электронная почта	332
Социальная сеть Twitter	333
СМС-сообщения	335
Облачные службы.....	338
Введение в веб-службы Amazon.....	339
Пакет AWS.jl	340
Платформа Google Cloud	344
Заклучение	347
Работа с Julia	348
Внутреннее устройство	348
Язык Fentolisp	349
Программный интерфейс Julia.....	350
Генерация машинных кодов	352
Советы относительно производительности	355
Наиболее успешная практика.....	355

Профилирование.....	357
Статический анализ кода	359
Отладка.....	361
Разработка пакета	363
Анатомия.....	363
Классификация.....	366
Использование Git.....	367
Публикация.....	369
Сообщества программистов	370
Классификации	371
Группа пакетов JuliaAstro.....	371
Группа пакетов JuliaGPU	378
Что не вошло в книгу?.....	381
Заключение	382

Символическая математика с Julia 384

Введение.....	384
О пакете.....	385
Уравнения	386
Элементарная алгебра	386
Решение уравнений.....	387
Построение графиков выражений	390
Пределы.....	392
Производные	393
Экстремумы	394
Интегрирование.....	394
Применения	395

Глоссарий основных терминов и сокращений 398

Предметный указатель 406

Осваиваем язык Julia

Julia – это хорошо структурированный язык программирования с большим быстродействием, устраняющий классическую проблему выполнения анализа на одном языке и трансляции его результатов на второй с целью повышения производительности. Эта книга поможет вам развить и усовершенствовать свои навыки программирования на Julia для решения задач автоматизации, возникающих в реальной жизни.

Книга начинается с небольшого инструктажа по поводу инсталляции и выполнения Julia в разных операционных средах. Затем вы сравните самые разные способы работы с языком и подробно изучите его ключевой функционал, разбирая практические примеры, построенные на основе пошагового принципа. Пользуясь простыми статистическими и аналитическими показателями, вы откроете для себя быстродействие языка, его реальную мощь, которая делает его особенно полезным в высокоинтенсивных вычислительных задачах, и отметите, что язык Julia способен сотрудничать с внешними процессами, получая значительное улучшение качества графики и визуализации данных. Наконец, вы займетесь метапрограммированием и узнаете, как оно укрепляет мощь языка и формирует его сетевую и распределенную вычислительную среду.

Для кого эта книга написана

Это практическое руководство предназначено для специалистов в области науки о данных. Книга предполагает наличие некоторых навыков работы с Julia и навыков программирования на скриптовом языке, таком как Python и R, либо на компилируемом языке, таком как C или Java.

Чему вы научитесь, прочитав эту книгу

- Инсталлировать среду разработки и выполнять ее сборку и настройку под вашу операционную среду.
- Создавать проекты в области науки о данных в рамках полного цикла ETL, анализа и визуализации данных.
- Понимать систему типов и принципы множественной диспетчеризации для получения большей отдачи от программирования на Julia.
- Взаимодействовать с файлами и таблицами данных с целью изучения простых статистических и аналитических показателей.
- Отображать графики и визуальные данные с целью проведения на Julia имитационного моделирования.
- Использовать Julia для взаимодействия с базами данных SQL и NoSQL.
- Работать с распределенными системами в веб-среде и в облаке.
- Разрабатывать свои собственные программные пакеты и участвовать в деятельности сообщества программистов на Julia в качестве соавтора.

Об авторе

Малкольм Шеррингтон работает в сфере информационных технологий более 35 лет. Он имеет степени в области математики, химии и инженерно-технических наук и читал лекции в двух различных университетах Великобритании, а также работал в аэрокосмической и медицинской отраслях экономики. В настоящее время руководит собственной компанией в финансовом секторе, с определенными интересами к высокоэффективным вычислениям и приложениям на основе графического процессора и параллельных вычислений.

Будучи деятельным специалистом, Малкольм начал программировать научные задачи на Fortran и C, совершенствуя навыки программирования на Ada и Common LISP, и недавно занялся обработкой данных и аналитикой на Perl, Python и R.

Малкольм Шеррингтон является организатором Лондонской ассоциации программистов на Julia. Кроме того, он является соорганизатором британской meetup-группы по высокопроизводительным вычислениям и финансовым технологиям и лондонской meetup-группы финансовых аналитиков.

Я бы хотел посвятить эту книгу памяти моей покойной жены, Хэйзел Шеррингтон, без поддержки которой я бы не взялся осваивать Julia и которой не суждено было увидеть подтверждения своей продуктивности.

Кроме того, хочу выразить особую благодарность Барбаре Доре и Джеймсу Уэймсу за их существенную помощь и материальную поддержку при подготовке этой книги.

О рецензентах

Гурурагав Гопал в настоящее время работает консультантом по управлению рисками в стартапах. Ранее он работал в Paterson Securities в качестве разработчика в области финансовой аналитики и консультанта по трейдингу. Кроме того, занимал должность консультанта по анализу данных и был связан с организацией электронной торговли. Он преподавал студентам и аспирантам в Технологическом университете VIT в Веллуре (Индия), специализируясь на распознавании образов, машинном обучении и больших данных. Как научный сотрудник связан с несколькими исследовательскими организациями, а именно IFMR и NAL. Кроме того, выступил рецензентом книги *Learning Data Mining with R* («Изучение интеллектуального анализа данных при помощи R»), вышедшей в издательстве Packt Publishing, а также нескольких журналов и конференций.

Он имеет степень бакалавра в области электротехники и электроники, а также степень магистра в области информатики и инженерии. Позже он выполнил свою курсовую работу от Института финансового управления и исследования (IFMR), Индия, по финансовой инженерии и управлению рисками, и с тех пор был занят в финансовой отрасли. Завоевал множество наград и имеет несколько международных публикаций.

Гурурагав Гопал интересуется программированием, обучением и выполняет консалтинговые услуги, а в свободное время слушает музыку.

По вопросам профессиональных консультаций к нему можно обратиться на его странице в соцсети LinkedIn (in.linkedin.com/in/gururaghavg).

Жуо СЛ – свободный разработчик из Китая с десятилетним опытом разработок в Linux, на C, C++, Java и Perl. Любит принимать участие в деятельности сообщества программистов (разумеется, включая сообщество программистов на Julia), в том числе в качестве соавтора. Ведет свой личный веб-сайт на <http://kdr2.com>, где о Жуо можно узнать еще больше.

Дэн Власюк – автор различных пакетов Julia, включая пакеты TimeSeries и Quandl; основатель группы JuliaQuant на веб-сайте Github, связанной с программными пакетами в области количественных финансов.

Предисловие

Julia – относительно молодой язык программирования. Первоначальные проектные работы по языку Julia начались в Массачусетском технологическом институте (MIT)¹ в августе 2009 года, и к февралю 2012-го он стал общедоступным. Заслуга по созданию этого языка в основном принадлежит трем разработчикам: Стефану Карпински, Джеффу Безансону и Виралу Шаху. Все трое, наряду с Аланом Эдельманом, до сих пор активно развивают проект Julia в MIT, где в настоящее время проводится ряд курсов по Julia, многие из которых доступны в Интернете.

Первоначально разработчики задумывали Julia как язык для научного программирования с быстродействием, достаточным для восполнения потребности в моделировании на интерактивном языке с последующей неизбежной переработкой кода на компилирующем языке, таком как C или Fortran. В то время главные языки для научного программирования, как, например, MATLAB и Mathematica, были защищены правами интеллектуальной собственности и до сих пор остаются относительно медленными. При этом имеются клоны этих языков в области открытого программного обеспечения, например GNU Octave и Scilab, но они работают еще медленнее. Когда язык Julia был запущен, сообщество программистов увидело в нем замену MATLAB, однако это не совсем верно. Несмотря на то что синтаксис Julia похож на MATLAB настолько, что любой компетентный в MATLAB специалист может легко изучить Julia, он не разрабатывался как клон. Это язык с более развитым функционалом и со многими существенными отличиями, которые будут подробно рассмотрены далее.

Период с 2009 года засвидетельствовал рост двух новых вычислительных дисциплин: больших данных / облачных вычислений и науки о данных. Обработка больших данных на Hadoop традиционно рассматривается как область программирования на Java, поскольку Hadoop выполняется в виртуальной машине Java. Разумеется, существует возможность обрабатывать большие данные при помощи языков программирования помимо тех, которые основаны на Java и используют парадигму jar-файлов с потоковой передачей, и тут Julia может применяться по аналогии с тем, как это делается на C++, C# и Python.

Появление науки о данных возвестило о начале использования языков программирования, которые просты для аналитиков, обладающих некими навыками программирования, но не являющихся специалистами в этой области. Два языка, развитие которых ускорилося, чтобы заполнить этот пробел, – R и Python. Оба они относительно стары и уходят своими корнями в 1990-е годы. Однако популярность обоих демонстрирует стремительный рост, по иронии, примерно с того

¹ Массачусетский технологический институт – университет и исследовательский центр, расположенный в Кембридже (штат Массачусетс, США). Также известен как Массачусетский институт технологий и Массачусетский технологический университет.

времени, когда общественности был представлен язык Julia. Тем не менее, даже имея такую признанную и солидную оппозицию, этот язык взволновал научное программистское сообщество и продолжает проводить рейды в этом направлении.

Цель этой книги состоит в том, чтобы охватить все аспекты языка Julia, которые делают его привлекательным для аналитиков данных. Язык развивается быстро. Двоичные дистрибутивы доступны для Linux, Mac OS X и Windows, но они отстают от текущих исходников. Поэтому, чтобы выполнять при помощи Julia серьезную работу, важно понимать, как получать и собирать рабочую систему из исходных текстов. Кроме того, для Julia доступны интерактивные среды разработки (IDE), и в данной книге будут проанализированы IDE Jupyter, Juno и плагин JuliaDT для среды Eclipse.

О чем рассказывает эта книга

Глава 1 «Среда разработки Julia» рассказывает о том, как запустить дистрибутив Julia и привести его в состояние готовности к работе. Важно уметь получать самые последние исходные тексты и собирать систему с нуля, а также находить и устанавливать надлежащие пакеты и при необходимости их удалять.

Глава 2 «Разработка на Julia» содержит краткий обзор части стандартных синтаксических конструкций языка. Julia – язык новый, но он отнюдь не покажется новым читателям с компетенцией в MATLAB, R или Python, поэтому цель главы состоит в том, чтобы с помощью примеров кратко довести до читателей информацию о Julia и направить их к онлайн-источникам. Кроме того, важно понимать разницу работы через консоль и с интегрированными средами разработки Jupyter/Juno/JuliaDT.

Глава 3 «Типы и диспетчеризация» посвящена системе типов Julia и показывает, каким образом она предоставляет разработчику мощные методы посредством ее de facto функциональной системы диспетчеризации.

Глава 4 «Функциональная совместимость» касается методов, посредством которых Julia может взаимодействовать с операционной системой и другими языками программирования. Эти методы являются в основном нативными для Julia. Глава заканчивается введением в параллелизм, который будет рассмотрен подробнее в главе 9.

Глава 5 «Работа с данными» рассматривает выполняемую аналитиком данных процедуру – от источника данных до результатов анализа. Большинство проектов начинается с данных, которые нужно прочесть, очистить и отобрать. Об этом и идет речь в главе. Далее описываются простые статистические и аналитические показатели.

Глава 6 «Научное программирование» фактически освещает главную причину для программирования на Julia. Мощь языка заключена в его быстродействии в сочетании с простотой разработки на языке сценариев, что делает его особенно ценным при решении задач с процессами, вычислительно ограниченными возможностями ЦПУ. В главе рассматриваются различные подходы, используемые при решении математических и естественнонаучных задач.

Глава 7 «Графика» описывает тот аспект, в котором Julia часто не выдерживает сравнения с другими альтернативными языками, такими как MATLAB и R. Действительно, более ранние версии языка имели достаточно ограниченную поддержку графики, но теперь дело обстоит иначе. В главе представлено большое разнообразие сложных подходов к визуализации графики на экране и сохранению ее в дисковых файлах.

Глава 8 «Базы данных» рассказывает о взаимодействии языка Julia с базами данных. В базе данных могут храниться данные для анализа – либо там требуется сохранить результаты анализа. В этой главе будет проанализировано несколько подходов к хранению данных в хранилищах SQL и NoSQL. Они не встроены в язык, а скорее полностью опираются на сторонние пакеты, поэтому в ближайшем будущем могут быть усовершенствованы.

Глава 9 «Сетевое взаимодействие» касается аспектов работы с распределенными источниками данных. В науке о данных большие данные и облачные системы приобретают все большую популярность, и в этой главе рассказывается о сетевом программировании на уровне сокета и взаимодействии через сеть Интернет. Кроме того, она посвящена анализу работы Julia в веб-службах Amazon и с вычислительным сервером Google.

Глава 10 «Работа с Julia» предоставляет читателям дополнительную информацию и побуждает продолжить работу, участвуя в совместной разработке на Julia. Можно попробовать свои силы в качестве соавтора, вносящего свой вклад в существующий пакет, либо просто примкнуть к одному из сообществ программистов на Julia.

Что вам потребуется для работы

Разработка на Julia может выполняться в любой из известных вычислительных операционных систем: Linux, OS X и Windows. В целях углубленного исследования языка читатель может захотеть получить последние версии и собрать язык из исходников под Linux. Однако для того, чтобы начать работать с языком в любой из трех операционных платформ, предусмотрена простая и удобная установка языка с использованием двоичного дистрибутива. Кроме того, можно отдельно скачать и установить интегрированную среду разработки (IDE) Juno и плагин для среды разработки Eclipse.

Некоторые примеры в более поздних главах, касающиеся поддержки баз данных, сетевого взаимодействия и облачных служб, потребуют установки дополнительных компонентов и использования дополнительных ресурсов, и вопросы их получения будут обсуждаться в соответствующих частях книги.

Для кого эта книга

Эта книга не является введением в программирование, и потому предполагается, что читатель знаком с понятийным аппаратом по крайней мере одного языка программирования. Для тех, кто знаком с языками сценариев, такими как Python, R

и MATLAB, эта задача не является трудной – как, впрочем, и для тех, кто использует похожие языки: C, Java и C#.

Однако для аналитика данных, возможно с компетенцией в методах аналитики на основе электронных таблиц, таких как Excel, или статистических пакетов, таких как SPSS и Stata, значительная часть текста должна оказаться полезной.

Соглашения

В этой книге используется несколько разных стилей оформления, каждый из которых имеет свое назначение. Ниже приведены примеры.

Фрагменты программного кода в тексте, названия таблиц баз данных, папок и файлов, расширения файлов, пути, фиктивные URL, данные, вводимые пользователем, и дескрипторы Twitter выделяются моноширинным шрифтом: «Папка test содержит программный код, который иллюстрирует, как писать тестовые сценарии и применять систему Base.Test».

Блок кода выглядит следующим образом:

```
function isAdmin2(_mc::Dict{ASCIIString,UserCreds}, _name::ASCIIString)
    check_admin::Bool = false;
    try
        check_admin = _mc[_name].admin
    catch
        check_admin = false
    finally
        return check_admin
    end
end
```

А ввод или вывод командной строки записывается следующим образом:

```
julia> include("asian.jl")
julia> run_asian()
```

Новые термины и важные слова выделены *курсивом*. Элементы интерфейса, системные сообщения и клавиши оформляются полужирным шрифтом: «**404, Страница не найдена**».

Сочетания клавиш, которые следует нажимать одновременно, оформляются с помощью знака + (плюс), например: «нажать сочетание клавиш **Ctrl+P**».

Последовательность команд обозначается знаком стрелки, например: «**File** → **Settings** (Файл → Настройки)».

- ☑ Предупреждения или важные примечания приводятся в отдельном текстовом блоке.
- 💡 Подсказки и приемы обозначены таким символом.
- ⚙ Так оформляются дополнения к тексту оригинала книги.

Обратная связь

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге — что понравилось или, может быть, не понравилось. Благодаря обратной связи мы получаем возможность выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв прямо на нашем сайте www.dmkpress.com: зайдите на страницу книги и оставьте комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы специалист в конкретной области и заинтересованы в написании новой книги, заполните на нашем сайте форму, размещенную на странице http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Скачивание исходного кода программ

Скачать сопроводительные файлы к книгам издательства «ДМК Пресс» можно на сайте www.dmkpress.com или www.дмк.рф. Зайдите на страницу с описанием интересующей вас книги и просмотрите список доступных материалов к ней.

Ошибки и опечатки

Хотя мы стремимся удовлетворить требования самых взыскательных читателей, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите нам о ней. Таким образом, вы избавите других читателей от досадных недоразумений и поможете нам улучшить последующие издания книги.

Если вам встретятся какие-либо ошибки в тексте, пожалуйста, сообщите об этом главному редактору по адресу dmkpress@gmail.com. Мы учтем ваши пожелания при выпуске следующих тиражей.

Нарушение авторских прав

Пиратство в Интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Packt очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы обнаружите в Интернете незаконно размещенные публикации любой нашей книги, пожалуйста, сообщите нам об этом с указанием ссылки на веб-страницу, чтобы мы могли применить соответствующие санкции.

Ссылки на подозрительные материалы направляйте по адресу dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов — это позволяет нам предоставлять читателям качественные материалы.

Вопросы

Если у вас имеются какие-либо вопросы, связанные с этой книгой, вы можете обратиться к нам по электронному адресу dmkpress@gmail.com, и мы приложим все усилия, чтобы помочь вам решить проблему.

Readme от автора

По ходу изложения я неоднократно отмечал, что Julia находится в состоянии разработки и даже еще не достиг уровня версии v1.0.

Одна из целей написания данной книги состояла в том, чтобы помочь вам почувствовать себя комфортно во время изучения среды разработки на Julia и при взаимодействии с сообществом программистов на Julia.

Хотя на момент перевода книги приводимый в ней программный код работал в версии v0.4 (и был адаптирован к последней версии v0.4.6) на разных платформах, в следующих версиях могут возникнуть расхождения.

Проблемы могут быть вызваны:

1. Не рекомендованными для использования (устаревшими) функциями или синтаксическими конструкциями. Например, к последней версии v0.4.x был пересмотрен синтаксис для массивов `Any[]`, кортежей, функции `convert()` и пр.
2. Включением пакетов в Базу (стандартную библиотеку).

Например, функции `DateTime` были включены в состав Базы для версии v0.4. Распределенные массивы были изъяты из Базы и существуют в виде пакета `DisributedArrays.jl`, который необходимо включать в среду в версиях v0.4 и выше.

3. Изменениями в пакетах.

Разработчикам пакетов необходимо некоторое время, чтобы принять к сведению изменения в функционале Базы и модифицировать свой программный код соответствующим образом. В некоторых случаях результатом таких изменений может стать изменение поведения всего пакета.

Кроме того, пакеты могут быть заменены на другие, программный код перераспределен либо пакет реструктурирован, в том числе путем создания подмодулей, изъятия либо переименования функций, модификации структуры типов.

Поэтому, если имеются проблемы с программным кодом, загляните в раздел `issues` стандартной библиотеки или конкретного пакета на Github либо попробуйте найти решение в Интернете. И наконец, задайте вопрос/заявите о проблеме в разделе `issues` на Github разработчика или на сайте вопросов и ответов для программистов Stackoverflow (<http://stackoverflow.com/questions/tagged/julia-lang?page=1&sort=newest&pagesize=15>).

Скорее всего, вы будете не первым, кто с этим столкнулся.

Комментарий переводчика

На сегодняшний день книга «Осваиваем язык Julia» представляет собой наиболее полное описание возможностей программирования на Julia. Год с момента изда-

ния оригинала книги, посвященной такому молодому и изменяющемуся языку как Julia, – это очень большой срок. Поэтому сразу стоит отметить, что весь материал книги приведен в соответствие с последней версией языка (v0.4.6 на июль 2016 года; в оригинале же рассматривалась стабильная на тот момент версия 0.3.x) и дополнен свежей информацией.

Книга содержит много аббревиатур и технических терминов из разных областей науки. Для удобства большинство аббревиатур расшифровывается в сносках, а для некоторых наименований в силу отсутствия единой терминологии приведены соответствующие варианты или пояснения.

В целом в книге принят сбалансированный подход к теории и программированию. Разумеется, можно найти более подробное изложение аспектов науки о данных и на основе других языков программирования. Однако основная задача книги заключается в том, чтобы объяснить читателю, каким образом можно использовать мощь языка Julia в научных вычислениях, в частности обращаясь к методам науки о данных.

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

Выполнение примеров программного кода на Julia

Приведенные в книге примеры протестированы в операционных системах Windows 8.1/10 и Linux/Lubuntu 16.4.

Прилагаемые к книге адаптированные и скорректированные примеры программного кода должны находиться в подпапке `julia_projects` домашней папки пользователя (`/home/julia_projects` или `C:\Users\[ИМЯ_ПОЛЬЗОВАТЕЛЯ]\julia_projects`). Ниже приведена структура папки с проектами:

Alice	данные для примеров из папки <code>code</code> и записных книжек Jupyter
code	консольные программы
data	данные для примеров из папки <code>code</code> и записных книжек Jupyter
gaston	скорректированный графический пакет Gaston
images	файлы изображений из книги и полученные в результате выполнения программ
Глава 01..Глава 10	записные книжки (практически полностью дублируют папку <code>code</code>)
Полезная информация	дополнительные примеры кода, записные книжки и пр.

Рекомендуем всегда начинать консольную сессию с команды

```
julia> cd(joinpath(homedir(), "julia_projects"))
```

или аналогичной ей, чтобы сделать папку с проектами Julia текущей. Впрочем, эту команду можно прописать в файле `.juliarc.jl`, который находится в папке `homedir()` (`C:\Users\[ИМЯ_ПОЛЬЗОВАТЕЛЯ]` в Windows; образец файла имеется в папке с прилагаемыми примерами), и тогда эта команда будет исполняться при каждом открытии консоли интерпретатора REPL.

В простейшем случае работу большинства приводимых в книге примеров можно проверить в окне консоли интерпретатора REPL, просто копируя в него фрагменты программного кода из буфера обмена (вставка правой кнопкой мыши возле подсказки `julia>`). В Windows вместо консоли Julia можно использовать консоль, например, Cmder (<http://cmder.net/>) с расширенными возможностями, включая эмуляцию команд Unix; она идет в минимальной и полной версиях и не требует установки (Julia запускается изнутри консоли).

Как вариант, можно перенести все примеры в корневую папку пользователя; скажем, в Windows ее расположение будет таким: `C:\Users\[ИМЯ_ПОЛЬЗОВАТЕЛЯ]\julia_projects`. В этом случае для выполнения примеров сначала нужно перейти в папку `;cd julia_projects` и затем включать оттуда примеры в среду Julia для исполнения, используя команду `include` – к примеру, `include("code/asian_option.jl")`.

Остальные подразумевают использование интегрированной среды разработки Juno/Atom, плагина JuliaDT для интегрированной среды разработки Eclipse, локального сервера записных книжек Jupyter и перспективной облачной веб-службы JuliaBox (<https://www.julibox.org/>), куда можно загружать локальные записные книжки и файлы с программным кодом на Julia для просмотра и исполнения.

В режиме разработки целесообразно в строке запуска консоли Julia прописать следующие переключатели:

- `-q, --quiet` – «тихий» запуск, то есть без стартового заголовка;
- `--depwarn={yes|no|error}` – активировать/деактивировать депрекационные предупреждения (`error` делает их ошибками);
- `--precompiled={yes|no}` – использовать прекомпилированный код из системного образа при наличии такового.

В следующей версии (0.5.0) предполагается новый переключатель:

- `--compilecache={yes|no}` – активировать/деактивировать обновляющую (`incremental`) прекомпиляцию модулей.

В целом строка должна иметь следующий вид:

```
julia.exe -q --precompiled=yes --depwarn=no
```

Установка среды разработки Julia

В Windows надо просто скачать установщик, соответствующий разрядности вашей операционной системы, и установить (<http://julialang.org/downloads/>). В Linux/Ubuntu сначала нужно выполнить небольшую подготовительную работу. Через терминал следует установить компиляторы:

```
sudo apt-get install gcc
sudo apt-get install g++
sudo apt-get install gfortran
```

Компоновщик:

```
sudo apt-get install cmake
```

И Git-клиент (факультативно):

```
sudo apt-get install git
```

А затем приступить собственно к установке Julia. Для операционных систем Ubuntu предусмотрен *Персональный архив пакетов (PPA)*, который предоставляет возможность автоматического обновления программного обеспечения до самой свежей стабильной версии, в данном случае языка Julia. Чтобы воспользоваться PPA и установить Julia в Ubuntu версий 12.04 и выше, в терминале следует выполнить следующие команды:

```
sudo add-apt-repository ppa:staticfloat/juliareleases
sudo add-apt-repository ppa:staticfloat/julia-deps
sudo apt-get update
sudo apt-get install julia
```

С другой стороны, можно установить ночные сборки, выполнив следующие команды:

```
sudo apt-add-repository ppa:staticfloat/julianightlies
sudo apt-add-repository ppa:staticfloat/julia-deps
sudo apt-get update
sudo apt-get install julia
```

Сборка новых версий происходит каждую ночь. Если вы уже установили Julia и желаете обновиться до последней версии, выполните следующее:

```
sudo apt-get update
sudo apt-get upgrade
```

Установка пакетов в среду разработки Julia

После установки Julia следует обновить пакеты:

```
Pkg.update()
```

И затем установить ряд пакетов, которые будут использоваться в примерах:

```
Pkg.add("ASCIIPLOTS")
```

Кроме упомянутого пакета `ASCIIPLOTS` аналогичным образом следует установить следующие пакеты: `IJulia`, `Atom`, `Winston`, `Gadfly`, `PyPlot`, `HDF5`, `Match` и некоторые другие. Как правило, при установке пакетов автоматически устанавливаются другие необходимые пакеты, например `DataStructures`, `DataFrames`, `DataArrays`, `Distributions`, `StatsBase`, `StatsFuns`, `Compose` и др.

Если по какой-то причине в работе пакета возникнет сбой, то, как правило, его повторная сборка решает проблему:

```
Pkg.build("ZMQ")
```

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru