



Содержание

Предисловие	12
Об авторе	17
Введение	18
О втором издании	18
Как организована эта книга	21
Об исходных текстах, приведенных в книге	23
«Путь Ruby»	24
Глава 1. Обзор Ruby	29
1.1. Введение в объектно-ориентированное программирование	30
1.2. Базовый синтаксис и семантика Ruby	35
1.3. ООП в Ruby	48
1.4. Динамические аспекты Ruby	57
1.5. Потренируйте свою интуицию: что следует запомнить	61
1.6. Жаргон Ruby	76
1.7. Заключение	79
Глава 2. Строки	80
2.1. Представление обычных строк	80
2.2. Альтернативная нотация для представления строк	81
2.3. Встроенные документы	83
2.4. Получение длины строки	83
2.5. Построчная обработка	83
2.6. Побайтовая обработка	84
2.7. Специализированное сравнение строк	84
2.8. Разбиение строки на лексемы	85
2.9. Форматирование строк	87
2.10. Строки в качестве объектов ввода/вывода	87
2.11. Управление регистром	88
2.12. Вычленение и замена подстрок	88
2.13. Подстановка в строках	90
2.14. Поиск в строке	91

2.15. Преобразование символов в коды ASCII и обратно	92
2.16. Явные и неявные преобразования	92
2.17. Дописывание в конец строки	94
2.18. Удаление хвостовых символов новой строки и прочих	94
2.19. Удаление лишних пропусков	95
2.20. Повтор строк	96
2.21. Включение выражений в строку	96
2.22. Отложенная интерполяция.....	96
2.23. Разбор данных, разделенных запятыми	97
2.24. Преобразование строки в число (десятичное или иное).....	98
2.25. Кодирование и декодирование строк в кодировке rot13.....	99
2.26. Шифрование строк	100
2.27. Сжатие строк	101
2.28. Подсчет числа символов в строке	101
2.29. Обращение строки.....	102
2.30. Удаление дубликатов	102
2.31. Удаление заданных символов	102
2.32. Печать специальных символов.....	102
2.33. Генерирование последовательности строк	103
2.34. Вычисление 32-разрядного CRC.....	103
2.35. Вычисление MD5-свертки строки	104
2.36. Вычисление расстояния Левенштейна между двумя строками	105
2.37. base64-кодирование и декодирование	106
2.38. Кодирование и декодирование строк (uuencode/uudecode).....	107
2.39. Замена символов табуляции пробелами и сворачивание пробелов в табуляторы	107
2.40. Цитирование текста.....	108
2.41. Заключение	109
Глава 3. Регулярные выражения	110
3.1. Синтаксис регулярных выражений	110
3.2. Компиляция регулярных выражений	112
3.3. Экранирование специальных символов	113
3.4. Якоря.....	113
3.5. Кванторы	114
3.6. Позитивное и негативное заглядывание вперед	116
3.7. Обратные ссылки.....	117
3.8. Классы символов.....	119
3.9. Обобщенные регулярные выражения	120

3.10. Сопоставление точки символу конца строки	121
3.11. Внутренние модификаторы	122
3.12. Внутренние подвыражения	122
3.13. Ruby и Oniguruma.....	123
3.14. Примеры регулярных выражений.....	129
3.15. Заключение	133
Глава 4. Интернационализация в Ruby.....	134
4.1. Исторические сведения и терминология	135
4.2. Кодировки в пост-ASCII мире.....	139
4.3. Справочники сообщений	150
4.4. Заключение	157
Глава 5. Численные методы.....	158
5.1. Представление чисел в языке Ruby.....	158
5.2. Основные операции над числами	159
5.3. Округление чисел с плавающей точкой	160
5.4. Сравнение чисел с плавающей точкой	162
5.5. Форматирование чисел для вывода.....	163
5.6. Вставка разделителей при форматировании чисел	163
5.7. Работа с очень большими числами	164
5.8. Использование класса BigDecimal	164
5.9. Работа с рациональными числами	166
5.10. Перемножение матриц	167
5.11. Комплексные числа	171
5.12. Библиотека mathn.....	172
5.13. Разложение на простые множители, вычисление НОД и НОК	172
5.14. Простые числа	173
5.15. Явные и неявные преобразования чисел.....	174
5.16. Приведение числовых значений	175
5.17. Поразрядные операции над числами	176
5.18. Преобразование системы счисления.....	177
5.19. Извлечение кубических корней, корней четвертой степени и т. д.	178
5.20. Определение порядка байтов	178
5.21. Численное вычисление определенного интеграла	179
5.22. Тригонометрия в градусах, радианах и градах	180
5.23. Неэлементарная тригонометрия.....	181
5.24. Вычисление логарифмов по произвольному основанию ...	182
5.25. Вычисление среднего, медианы и моды набора данных ...	182

5.26. Дисперсия и стандартное отклонение	183
5.27. Вычисление коэффициента корреляции	184
5.28. Генерирование случайных чисел	185
5.29. Кэширование функций с помощью метода memoize	186
5.30. Заключение	187
Глава 6. Символы и диапазоны	188
6.1. Символы.....	188
6.2. Диапазоны.....	192
6.3. Заключение	200
Глава 7. Дата и время	202
7.1. Определение текущего момента времени	203
7.2. Работа с конкретными датами (после точки отсчета)	203
7.3. Определение дня недели.....	204
7.4. Определение даты Пасхи.....	204
7.5. Вычисление n-ого дня недели в месяце	205
7.6. Преобразование из секунд в более крупные единицы.....	206
7.7. Вычисление промежутка времени, прошедшего от точки отсчета	207
7.8. Високосные секунды	207
7.9. Определение порядкового номера дня в году	208
7.10. Контроль даты и времени	208
7.11. Определение недели в году	209
7.12. Проверка года на високосность	210
7.13. Определение часового пояса.....	210
7.14. Манипулирование временем без даты.....	211
7.15. Сравнение моментов времени.....	211
7.16. Прибавление интервала к моменту времени.....	211
7.17. Вычисление разности между двумя моментами времени	212
7.18. Работа с конкретными датами (до точки отсчета).....	212
7.19. Взаимные преобразования объектов Date, Time и DateTime	213
7.20. Извлечение даты и времени из строки	214
7.21. Форматирование и печать даты и времени	215
7.22. Преобразование часовых поясов	216
7.23. Определение числа дней в месяце	216
7.24. Разбиение месяца на недели	216
7.25. Заключение	218

Глава 8. Массивы, хэши и другие перечисляемые структуры.....	219
8.1. Массивы.....	219
8.2. Хэши.....	242
8.3. Перечисляемые структуры в общем.....	252
8.4. Заключение.....	259
Глава 9. Более сложные структуры данных.....	260
9.1. Множества.....	260
9.2. Стеки и очереди.....	263
9.3. Деревья.....	268
9.4. Графы.....	274
9.5. Заключение.....	280
Глава 10. Ввод/вывод и хранение данных.....	281
10.1. Файлы и каталоги.....	282
10.2. Доступ к данным более высокого уровня.....	306
10.3. Библиотека KirbyBase.....	314
10.4. Подключение к внешним базам данных.....	317
10.5. Заключение.....	329
Глава 11. ООП и динамические механизмы в Ruby.....	330
11.1. Рутинные объектно-ориентированные задачи.....	331
11.2. Более сложные механизмы.....	356
11.3. Динамические механизмы.....	375
11.4. Заключение.....	395
Глава 12. Графические интерфейсы для Ruby.....	396
12.1. Ruby/Tk.....	397
12.2. Ruby/GTK2.....	409
12.3. FXRuby (FOX).....	422
12.4. QtRuby.....	436
12.5. Другие библиотеки для создания графических интерфейсов.....	446
12.6. Заключение.....	447
Глава 13. Поток в Ruby.....	448
13.1. Создание потоков и манипулирование ими.....	449
13.2. Синхронизация потоков.....	458
13.3. Заключение.....	473

Глава 14. Сценарии и системное администрирование	474
14.1. Запуск внешних программ	474
14.2. Флаги и аргументы в командной строке	479
14.3. Библиотека Shell	482
14.4. Переменные окружения	485
14.5. Сценарии на платформе Microsoft Windows	487
14.6. Моментальный инсталлятор для Windows	493
14.7. Библиотеки, о которых полезно знать	494
14.8. Работа с файлами, каталогами и деревьями	495
14.9. Различные сценарии	498
14.10. Заключение	502
Глава 15. Ruby и форматы данных	503
15.1. Разбор XML и REXML	503
15.2. RSS и Atom	508
15.3. Обработка изображений с помощью RMagick	512
15.4. Создание документов в формате PDF с помощью библиотеки PDF:Writer	521
15.5. Заключение	530
Глава 16. Тестирование и отладка	531
16.1. Библиотека Test::Unit	531
16.2. Комплект инструментов ZenTest	535
16.3. Работа с отладчиком Ruby	538
16.4. Использование irb в качестве отладчика	541
16.5. Измерение покрытия кода	542
16.6. Измерение производительности	543
16.7. Объекты печати	547
16.8. Заключение	548
Глава 17. Создание пакетов и распространение программ	550
17.1. Программа RDoc	550
17.2. Установка и подготовка пакета	555
17.3. RubyForge и RAA	559
17.4. Заключение	560
Глава 18. Сетевое программирование	561
18.1. Сетевые серверы	562
18.2. Сетевые клиенты	572
18.3. Заключение	591

Глава 19. Ruby и Web-приложения	592
19.1. Программирование CGI на Ruby	592
19.2. FastCGI	597
19.3. Ruby on Rails	599
19.4. Разработка Web-приложений с помощью Nitro.....	603
19.5. Введение в Wee	615
19.6. Разработка Web-приложений с помощью IOWA.....	617
19.7. Ruby и Web-сервер	622
19.8. Заключение	629
Глава 20. Распределенный Ruby	630
20.1. Обзор: библиотека drb.....	630
20.2. Пример: эмуляция биржевой ленты	633
20.3. Rinda: пространство кортежей в Ruby	636
20.4. Обнаружение сервисов в распределенном Ruby.....	640
20.5. Заключение	641
Глава 21. Инструменты разработки для Ruby	642
21.1. Система RubyGems.....	642
21.2. Программа Rake	644
21.3. Оболочка irb.....	647
21.4. Утилита ri	652
21.5. Поддержка со стороны редакторов.....	653
21.6. Интегрированные среды разработки	654
21.7. Заключение	656
Глава 22. Сообщество пользователей Ruby	657
22.1. Ресурсы в Web	657
22.2. Новостные группы и списки рассылки.....	658
22.3. Блоги и онлайн-журналы	658
22.4. Запросы на изменение Ruby	659
22.5. Каналы IRC	659
22.6. Конференции по Ruby.....	659
22.7. Локальные группы пользователей Ruby	660
22.8. Заключение	660
Алфавитный указатель	662



Предисловие

Предисловие ко второму изданию

В древнем Китае люди, в особенности философы, полагали, что под внешней оболочкой мира и любого существа скрыто нечто, чего нельзя ни объяснить, ни описать словами. Это нечто китайцы называли Дао, а японцы – До. На русский язык это слово можно перевести как Путь. Слово «до» входит в такие названия, как дзюдо, кендо, карате-до и айкидо. Это не просто боевые искусства, а целая философия и взгляд на жизнь.

Так и в языке программирования Ruby есть свои философия и способ мышления. Этот язык заставляет думать по-новому. Он помогает программистам получать удовольствие от своей работы. И не потому, что Ruby был создан в Японии, а потому, что программирование стало важной частью существования – по крайней мере, для некоторых людей, жизнь которых Ruby призван улучшить.

Как всегда, описать, что такое Дао, трудно. Я чувствую это, но не могу подыскать нужных слов даже на японском, моем родном языке. А вот один смельчак по имени Хэл Фултон попытался, и его первая попытка (первое издание этой книги) оказалась довольно удачной. Попытка номер два увенчалась еще лучшим результатом, чему немало способствовала помощь многих людей из сообщества пользователей Ruby. По мере того как Ruby набирает популярность (отчасти благодаря продукту Ruby on Rails), все важнее становится овладение секретами мастерства производительного программирования на этом языке. Надеюсь, что книга, которую вы держите в руках, поможет вам в решении этой задачи.

Удачной работы!
Юкихино «Мац» Мацумото
Август 2006 года, Япония

Предисловие к первому изданию

Вскоре после того как я впервые познакомился с компьютерами в начале 1980-х годов, меня заинтересовали языки программирования. С тех пор я буквально помешался на этой теме. Думаю, причина такого интереса в том, что языки программирования – это способ выражения мыслей. Они по сути своей предназначены для человека.

Но вопреки этому факту языки программирования почему-то всегда оказывались в большей степени машинно-ориентированными. Многие из них спроектированы с учетом удобства для компьютеров.

По мере того как компьютеры становятся мощнее и дешевле, ситуация постепенно меняется. Возьмем, к примеру, структурное программирование. Машине все равно, насколько хорошо структурирована программа: она просто исполняет ее команда за командой. Идеи структурного программирования обращены к людям, а не к машинам. То же относится и к объектно-ориентированному программированию.

Пришло время проектировать языки, удобные для людей!

В 1993 году я разговаривал со своим коллегой о сценарных языках, их выразительности и перспективах. Я считал, что программирование пойдет именно по этому пути и будет ориентироваться на человека. Но меня не вполне устраивали существующие языки, такие как Perl и Python. Я хотел видеть язык более мощный, чем Perl, и более объектно-ориентированный, чем Python. Найти идеальный вариант мне не удалось, поэтому остался один выход: изобрести свой собственный.

Ruby – не самый простой язык, но ведь и человеческая душа не проста. Ей равно нравятся простота и сложность. Она не приемлет ни слишком примитивных, ни чересчур заумных вещей. Она ищет равновесия.

Поэтому при проектировании ориентированного на человека языка – Ruby – я следовал принципу наименьшего удивления. Иными словами, работа шла под девизом: хорошо то, что не кажется мне странным. Вот почему я чувствую себя в своей родной стихии и испытываю радость, когда программирую на Ruby. А с момента выхода в свет первой версии в 1995 году многие программисты во всем мире разделили со мной эту радость.

Как всегда, хочу выразить величайшую благодарность всем членам сообщества, сложившегося вокруг Ruby. Они – причина его успеха.

Также я благодарен автору этой книги, Хэлу Фултону, за то, что он показал другим Путь Ruby. В книге объясняется философия, стоящая за языком Ruby. Это квинтэссенция моих мыслей и ощущений членов сообщества. Интересно, как Хэлу удалось прочитать мои мысли и раскрыть секрет Пути Ruby!.. Я никогда не встречался с ним лично, надеюсь, что скоро это все-таки произойдет.

В заключение хочу выразить надежду, что эта книга и сам язык Ruby помогут вам получить удовольствие и радость от программирования.

Юкихино «Мац» Мацумото
Сентябрь 2001, Япония



Благодарности

Благодарности за второе издание

Здравый смысл подсказывает, что второе издание требует в два раза меньше работы, чем первое. Но здравый смысл ошибается.

Хотя значительная часть текста книги перекочевала прямиком из первого издания, даже эту часть пришлось сильно править. К каждому предложению приходилось ставить вопрос: «Сохранилось ли в 2006 году то, что было верно в 2001-м?» И это только начало!

Короче говоря, я потратил много сотен часов на подготовку второго издания – примерно столько же, сколько ушло на первое. И тем не менее я «всего лишь автор».

Книга появляется на свет в результате усилий многих людей. Если говорить об издательстве, то я благодарен Дебре Вильямс Коули (Debra Williams Cauley), Сонглин Киу (Songlin Qiu) и Менди Фрэнк (Mandie Frank) за тяжелый труд и бесконечное терпение. Спасибо Джениль Бриз (Geneil Breeze) за неутомимое вычитывание и выпалывание сорняков из моего английского. Есть много других сотрудников, которых я не могу назвать, поскольку их работа проходила за кулисами и я никогда с ними не встречался.

За техническое редактирование отвечали главным образом Шашанк Дейт (Shashank Date) и Фрэнсис Хван (Francis Hwang). Они прекрасно справились со своей задачей – если остались какие-то ошибки, это всецело моя вина.

Большое спасибо людям, которые предлагали объяснения, писали код примеров и отвечали мне на многочисленные вопросы. Это сам Мац (Юкихино Мацумото), Дейв Томас (Dave Thomas), Кристиан Нойкирхен (Christian Neukirchen), Чед Фаулер (Chad Fowler), Дэниэл Бергер (Daniel Berger), Армин Перль (Armin Roehrl), Стефан Шмидль (Stefan Schmiedl), Джим Вайрих (Jim Weirich), Райан Дэвис (Ryan Davis), Джени У. (Jenny W.), Джим Фриз (Jim Freeze), Лайл Джонсон (Lyle Johnson), Мартин Де Мелло (Martin DeMello), Март Лоуренс (Mart Lawrence), Рон Джеффрис (Ron Jeffries), Тим Хантер (Tim Hunter), Чет Хендриксон (Chet Hendrickson), Натаниэль Талбот (Nathaniel Talbott) и Бил Клеб (Bill Kleb).

Особая благодарность активным помощникам. Эндрю Джонсон (Andrew Johnson) здорово обогатил мои знания о регулярных выражениях. Пол Бэтли (Paul Battley) предоставил обширный материал для главы об интернационализации. Масао Мута (Masao Mutoh) помог в написании той же главы, а также снабдил меня материалами по GTK. Остин Зиглер (Austin Ziegler) научил меня секретам

подготовки PDF-файлов. Калев Теннис (Kaleb Tennis) дал материал по Qt. Эрик Ходел (Eric Hodel) помог в описании продуктов Rinda и Ring, а Джеймс Бритт (James Britt) внес большой вклад в главу о разработке приложений для Web.

И еще раз выражаю искреннюю благодарность и восхищение Мацу – не только за помощь, но и прежде всего – за создание Ruby. Domo arigato gozaimasu*!

Еще раз хочу поблагодарить своих родителей. Они постоянно подбадривали меня и с нетерпением ждали выхода книги... Я еще сделаю из них программистов!

И опять – спасибо всем членам сообщества пользователей Ruby за неутомимую энергию и дух коллективизма. Особенно я благодарен читателям этой книги (обоих изданий). Я надеюсь, что вы найдете ее информативной, полезной и, быть может, даже увлекательной.

Благодарности за первое издание

Создание книги – плод усилий целого коллектива. Я не понимал этого в полной мере, пока не взялся за это дело сам. Рекомендую всем пройти подобное испытание, хотя оно и не из легких. Нет сомнений, что без помощи многих и многих людей книга не увидела бы свет.

Прежде всего, выражаю благодарность и восхищение Мацу (Юкихино Мацумото), который создал язык Ruby. Domo arigato gozaimasu*!

Спасибо Конраду Шнейкеры (Conrad Schneiker), который подал мне идею написать эту книгу и помог выработать ее общий план. Он же оказал мне неоценимую услугу, познакомив с языком Ruby в 1999 году.

Материалом для книги меня снабжали несколько человек. Первым хочу назвать Гая Хэрста (Guy Hurst), который написал значительную часть начальных глав, а также два приложения. Его помощь поистине бесценна.

Спасибо также другим помощникам, называя которых, я не придерживаюсь какого-то определенного порядка. Кэвин Смит (Kevin Smith) многое сделал для раздела главы 6, посвященного GTK, избавив меня от изучения сложной темы в условиях жесткого графика. Патрик Логан (Patrick Logan) пролил свет на графическую систему FOX GUI, описанную в той же главе. Чед Фаулер (Chad Fowler) в главе 9 углубился в тайны XML, а также помог при написании раздела о CGI.

Благодарю всех, кто правил корректуру, писал рецензии и помогал иными способами: Дона Мучоу (Don Muchow), Майка Стока (Mike Stok), Михо Огисима (Miho Ogishima) и уже упомянутых выше. Спасибо Дэвиду Эппштейну (David Eppstein), профессору математики, который ответил на вопросы по теории графов.

Одна из замечательных особенностей Ruby – поддержка со стороны сообщества пользователей. В списке рассылки и в конференции многие отвечали на мои вопросы, подавали идеи и всячески помогали. Опять же не придерживаясь определенного порядка, хочу упомянуть Дейва Томаса (Dave Thomas), Энди Ханта (Andy Hunt), Хи-Соб Парка (Hee-Sob Park), Майка Уилсона (Mike Wilson), Ави Брайанта (Avi Bryant), Ясуси Шоджи (Yasushi Shoji «Yashi»), Шуго Маэда (Shugo

* Огромное спасибо (яп.)

Maeda), Джима Вайриха (Jim Weirich) и Масаки Сукета (Masaki Suketa). Как это ни печально, но, скорее всего, кого-то я пропустил.

Очевидно, что книга никогда не вышла бы без помощи издателя. Многие работали над ней незримо для меня, но особенно я хотел бы поблагодарить Вильяма Брауна (William Brown), который тесно сотрудничал со мной и постоянно поощрял, и Скотта Мейера (Scott Meyer), который скрупулезно занимался объединением всех материалов. Других назвать не могу, потому что никогда о них не слышал. Но они знают себя сами.

Хочу поблагодарить своих родителей, которые следили за проектом издания, подбадривали меня и даже дали себе труд ради меня освоить азы программирования.

Один мой друг как-то сказал: «Если ты написал книгу, которую никто не читает, значит, ты не написал ее вовсе». Поэтому напоследок я хочу поблагодарить читателей. Эта книга для вас. Надеюсь, что она окажется полезной.



Об авторе

Хэл Фултон – обладатель двух ученых степеней по информатике, полученных в Университете штата Миссисипи. Он четыре года преподавал информатику в колледже, пока не переехал в Остин, штат Техас, для работы по контрактам (в основном с отделением компании IBM в Остине). Более 15 лет он работал с различными версиями ОС UNIX, в том числе AIX, Solaris и Linux. С языком Ruby впервые познакомился в 1999 году, а в 2001 приступил к работе над первым изданием этой книги – второй книги на английском языке, посвященной Ruby. Фултон присутствовал на шести конференциях по Ruby и проводил презентации на четырех из них, в частности на первой европейской конференции по языку Ruby, состоявшейся в Карлсруэ (Германия). Сейчас Хэл работает в компании Broadwing Communications, располагающейся в Остине, и занимается вопросами, связанными с большим хранилищем данных и относящимися к нему телекоммуникационными приложениями. В работе он использует язык C++, СУБД Oracle и, конечно, Ruby.

Фултон по-прежнему постоянно присутствует в списке рассылки и в IRC-канале, посвященном Ruby, а также участвует в нескольких разрабатываемых проектах на Ruby. Он член Ассоциации по вычислительной технике (ACM – Association for Computing Machinery) и компьютерного общества IEEE (Институт инженеров по электротехнике и электронике). В свободное от работы время увлекается музыкой, чтением, искусством и фотографией. Кроме всего прочего, Хэл член общества по изучению Марса и энтузиаст космических полетов. Мечтал бы когда-нибудь совершить такой полет. Проживает в Остине, штат Техас.



Введение

.....
*Путь, о котором можно поведать, –
не постоянный Путь.
Лао Цзы, «Дао де цзин»**

Эта книга называется «Путь Ruby»**. Название нуждается в некотором пояснении.

Я ставил себе целью выразить в этой книге философию языка Ruby, насколько это в моих силах. Ту же цель преследовали мои добровольные помощники. Успех должно разделить между всеми, а ошибки остаются моей и только моей виной.

Конечно, я не могу абсолютно точно сказать, в чем же состоит истинный дух Ruby. Эту задачу я оставляю Мацу, но подозреваю, что даже ему трудно будет облечь свои ощущения в слова.

Короче говоря, «The Ruby Way» – всего лишь книга, а Путь Ruby – удел создателя языка и сообщества в целом. Втиснуть его в рамки книги довольно трудно. И все-таки в этом введении я попытаюсь хотя бы отчасти передать неуловимый дух Ruby. Мудрый ученик не воспримет эту попытку как окончательный вердикт!

Не забывайте, что это второе издание. Большая часть введения сохранена, но впереди раздел «О втором издании», в котором описываются изменения и вновь включенный материал.

О втором издании

Все меняется, и Ruby – не исключение. Я пишу это введение в 2006 году, когда с момента выхода первого издания прошло уже почти пять лет. Настало время для обновления.

В это издание внесено немало исправлений и добавлено много нового материала. Прежняя глава 4 («Простые задачи, связанные с обработкой данных») превратилась в шесть глав, две из которых («Диапазоны и символы» и «Интернационализация в Ruby») совсем новые; в четыре остальных добавлены новые примеры и комментарии к ним. Сильно расширен материал о регулярных выражениях: теперь рассматриваются не только «классические» выражения, но и новая библиотека для их поддержки, Oniguruma.

Главы 8 и 9 раньше составляли одну главу. Она была разбита на две, поскольку из-за вновь добавленного материала оказалась слишком большой.

* Пер. В. Малявина. – *Прим. ред.*

** Оригинальное название книги. Дословный перевод «Путь Ruby», однако в русскоязычном издании было выбрано название «Программирование на языке Ruby». – *Прим. ред.*

Аналогичным образом главы 18, 19 и 20 образовались в результате разделения главы 9. Приложения удалены, чтобы освободить место для основного материала.

Появились также следующие новые главы:

- глава 15, «Форматы данных в Ruby». Здесь рассматриваются форматы XML, RSS, графические файлы, создание PDF-файлов и другие вопросы;
- глава 16, «Тестирование и отладка». Речь идет о тестировании, профилировании, отладке, анализе кода и тому подобных вещах;
- в главе 17, «Создание пакетов и распространение программ», обсуждаются, в частности, инструмент `setup.rb` и создание пакетов в формате `RubyGem`;
- глава 21, «Инструменты разработки для Ruby», знакомит с поддержкой Ruby в редакторах и интегрированных средах разработки (IDE), утилитой `ri` и форматом `RubyGem` с точки зрения пользователя;
- в главе 22, «Сообщество Ruby», приводятся основные Web-сайты, списки рассылки, форумы, конференции, IRC-каналы по Ruby и прочие дополнительные сведения.

В широком смысле все главы в этой книге «новые». Каждую из них я подверг тщательной ревизии, внес сотни мелких и десятки крупных изменений. Был исключен материал, утративший актуальность и показавшийся мне не слишком важным. Я учел изменения в самом языке Ruby, а также добавил в каждую главу новые примеры и комментарии.

Возможно, вас интересует, что было добавлено в старые главы. Отвечаю: уже упомянутая библиотека регулярных выражений `Oniguruma`; математические библиотеки и классы, включая `BigDecimal`, `mathn` и `matrix`; такие новые классы, как `Set` и `DateTime`.

В главу 10, «Ввод/вывод и хранение данных», я добавил материал о методе `readpartial`, неблокирующем вводе/выводе и классе `StringIO`. Также рассмотрены форматы CSV, YAML и `KirbyBase`. В ту часть главы 10, которая посвящена базам данных, включены сведения о СУБД `Oracle` и `SQLite`, интерфейсе `DBI`, а также об объектно-реляционном отображении (`Object-Relational Mappers – ORM`).

Глава 11, «ООП и динамические механизмы в Ruby», пополнилась информацией о таких недавно добавленных в язык конструкциях, как `initialize_copy`, `const_get`, `const_missing` и `define_method`. Также я рассматриваю делегирование и перенаправление.

Глава 12, «Графические интерфейсы для Ruby», была переработана целиком (в особенности разделы, посвященные `GTK` и `Fox`). Раздел по `QtRuby` – новый от начала до конца.

В главе 14, «Сценарии и системное администрирование», теперь обсуждаются моментальный инсталлятор для `Windows` и ряд аналогичных пакетов. Кроме того, улучшен код примеров.

В главе 18, «Сетевое программирование», появились разделы о вложениях в электронные письма и о взаимодействии с `IMAP`-сервером. Также рассматривается библиотека `OpenURI`.

В главе 19, «Ruby и Web-приложения», теперь рассматриваются продукты Ruby on Rails, Nitro, Wee, IOWA и другие инструменты для Web. Также уделено внимание инструментам WEBrick и в какой-то мере Mongrel.

В главу 20, «Распределенный Ruby», добавлен материал о системе Rinda – реализации пространства кортежей, написанной на Ruby. Тут же приводятся сведения о родственной системе Ring.

Так ли необходимы все эти добавления? Да, уверяю вас.

Напомню, кстати, что «The Ruby Way» – это вторая вышедшая на английском языке книга по языку Ruby; первой была знаменитая «Мотыга», или «Programming Ruby», Дэйва Томаса (Dave Thomas) и Энди Ханта (Andy Hunt). Моя книга была составлена так, чтобы не перекрывать, а дополнять свою предшественницу; это и стало одной из основных причин ее популярности.

Когда я приступал к работе над первым изданием, еще не было международных конференций по Ruby. Не было сайтов RubyForge, ruby-doc.org, не было Wiki-страницы rubygarden.org. Вообще в Сети не было почти ничего, кроме официального сайта Ruby. В архиве приложений Ruby насчитывалось всего несколько сотен программ.

В то время лишь немногие периодические издания (как обычные, так и онлайн-новые) знали о существовании этого языка. Когда где-то публиковалась статья о Ruby, мы все брали ее на заметку; о ней сообщалось в списке рассылки и там же проводилось обсуждение.

Многих привычных сегодня инструментов и библиотек еще не существовало. Все пока было впереди: и система RDoc, и пакет REXML для анализа XML-документов. Математическая библиотека была куда беднее нынешней. Поддержка баз данных была фрагментарной, а драйверы ODBC и вовсе отсутствовали. Tk был чуть ли не единственным графическим интерфейсом. Приложения для Web разрабатывались в виде низкоуровневых CGI-сценариев.

Еще не появился «моментальный» инсталлятор для Windows. Пользователям Windows приходилось выполнять компиляцию исходных текстов в среде Cygwin или с помощью minigw.

Системы RubyGem не было даже в примитивной форме. Процедура поиска и установки приложений проводилась вручную; для решения этой задачи использовались инструменты типа tar и make.

Никто слыхом не слыхал о Ruby on Rails. Никто (насколько мне известно) не употреблял термина «утипизация»*. Не было ни YAML для Ruby, ни системы Rake.

В то время мы пользовались версией Ruby 1.6.4 и считали ее безмерно крутой. Но Ruby 1.8.5 (с которой я обычно работаю сейчас) еще круче!

Был незначительно изменен синтаксис, но не настолько серьезно, чтобы об этом писать. По большей части речь идет о «граничных случаях», и теперь синтаксис в этих ситуациях выглядит более разумно. Ruby всегда отличали странности в отношении к необязательности скобок; в 98% случаев вы никакой раз-

* Речь идет о переводе выражения «duck typing». Его смысл и происхождение объясняются в разделе 1.6. Приношу извинения ревнителям чистоты русского языка за то, что не смог удержаться от игры слов и выдумал этот «термин» вместо «утиной типизации». – *Прим. перев.*

ницы не заметите, а если заметите, то, наверное, согласитесь, что язык стал более последовательным.

Изменилась семантика некоторых системных методов. Но несущественно. Например, раньше метод `Dir#chdir` не принимал в качестве параметра блок, но несколько лет назад это стало допустимо.

Некоторые системные методы объявлены устаревшими или переименованы. Метод `class` утратил свой псевдоним `type` (поскольку в Ruby мы обычно не говорим о типах объектов). Метод `intern` получил более понятное название `to_sym`, а метод `Array#indices` называется `Array#values_at`. Можно было бы продолжить, но думаю, что суть вы уловили.

Кроме того, было добавлено несколько новых системных методов, например `Enumerable#inject`, `Enumerable#zip` и `IO#readpartial`. Старая библиотека `futils` теперь называется `fileutils`, и у нее появилось собственное пространство имен `FileUtil`, тогда как раньше она добавляла методы в класс `File`.

Есть и еще много изменений. Но важно понимать, что все они вносились очень осторожно и аккуратно. Язык как был Ruby, так им и остался. Красота Ruby в небольшой степени обязана тому факту, что он изменяется медленно и обдуманно, ведомый мудростью Маца и других разработчиков.

Сегодня нет недостатка в книгах по Ruby. Публикуется больше статей, чем мы в состоянии переварить. Множатся руководства и документация в сети Web.

Появились новые инструменты и библиотеки. По разным причинам большая их часть – это каркасы и инструменты для разработки Web-приложений, средства для создания сетевых дневников (блогов), разметки, а также для объектно-реляционного отображения (ORM). Но есть и инструментарий для работы с базами данных, организации графических интерфейсов, математических расчетов, Web-сервисов, обработки изображений, управления версиями и т. д.

Поддержка Ruby в редакторах широко распространена и достигла немалой изощренности. Существуют интегрированные среды разработки (IDE), весьма полезные и зрелые; частично они перекрываются с конструкторами графических интерфейсов.

Нет сомнений и в том, что сообщество пользователей разрослось и изменилось. Сегодня Ruby никак не назовешь нишевым языком: им пользуются в НАСА, Национальной администрации по океану и атмосфере (NOAA), компании Motorola и во многих других крупных организациях. Он применяется для работы с графикой, доступа к базам данных, численного анализа, Web-приложений и в других областях. Короче говоря, Ruby стал весьма популярным языком.

Я работал над новой редакцией этой книги с любовью. Надеюсь, что она окажется вам полезной.

Как организована эта книга

Вряд ли вы станете изучать Ruby по этой книге. В ней не так уж много вводного и учебного материала. Если вы еще ничего не знаете о Ruby, то лучше начать с какой-нибудь другой книги.

Но программисты – народ упорный, и я допускаю, что научиться Ruby *только* по этой книге возможно. В главе 1, «Обзор Ruby», приводится краткое введение в язык и очень скромное руководство.

Также в главе 1 есть довольно полный перечень «скольких мест» (который трудно поддерживать в актуальном состоянии). Для разных читателей этот перечень полезен в разной мере, поскольку что для одного интуитивно очевидно, для другого выглядит странно.

В основном эта книга призвана отвечать на вопросы типа «Как сделать?». И потому вы, вероятно, многое будете пропускать. Я почту за честь, если кто-то прочтет книгу от корки до корки, но не надеюсь на это. Скорее я ожидаю, что вы будете искать в оглавлении темы, которые вас интересуют в конкретный момент. Впрочем, с момента выхода первого издания мне приходилось беседовать с разными людьми, и оказалось, что многие прочли книгу целиком. Более того, несколько человек писали мне, что выучили по ней Ruby. Что ж, все возможно!..

Некоторые рассматриваемые в книге вопросы могут показаться элементарными. Но ведь у разных людей и опыт разный; то, что очевидно одному, будет открытием для другого. Я старался сделать изложение как можно более полным. С другой стороны, было стремление уложиться в разумный объем (ясно, что эти цели противоречивы).

Можно назвать эту книгу «справочником наоборот». Вы ищете то, что нужно, не по имени класса или метода, а по функции или назначению. Например, в классе `String` есть несколько методов для манипулирования регистром букв: `capitalize`, `upcase`, `casecmp`, `downcase` и `swapcase`. В настоящем справочнике они встречались бы в алфавитном порядке, а в этой книге собраны в одном месте.

Конечно, в борьбе за полноту охвата материала я иногда сворачивал на путь, которому следуют справочные руководства. Во многих случаях я старался компенсировать это, предлагая не совсем обычные примеры или разнообразия их по сравнению со справочниками.

Я старался не перегружать код комментариями. Если не считать первой главы, то думаю, что достиг этой цели. Писатель может стать не в меру болтливым, но программист-то хочет видеть код (а если не хочет, то должен хотеть).

Иногда примеры выглядят искусственными, за что я приношу свои извинения. Проиллюстрировать какой-то прием или принцип в отрыве от реальной задачи бывает сложно. Но чем сложнее задача, чем выше ее уровень, тем большие усилия я прилагал к подысканию реального примера. Так, если речь идет о конкатенации строк, то, наверное, вы увидите безыскусный фрагмент кода с упоминанием пресловутых “foo” и “bar”, но когда рассматривается тема разбора XML-документа, будет приведен куда более содержательный и реалистичный пример.

Есть в этой книге два-три каприза, в которых хочу заранее сознаться. Во-первых, я всеми силами старался избегать «уродливых» глобальных переменных типа `$_` и ей подобных, пришедших из языка Perl. Они есть в Ruby и прекрасно работают, даже применяются в повседневной работе всеми или большинством программистов. Но почти всегда от их использования можно уйти, что я и позволил себе чуть ли не во всех примерах.

Другой каприз состоит в том, что я избегаю пользоваться обособленными выражениями, если у них нет побочных эффектов. В Ruby выражения – одна из основ языка, и это прекрасно; я старался извлечь из этой особенности максимум пользы. Но во фрагментах кода предпочитаю не употреблять выражения, которые просто возвращают никак не используемое значение. Например, для иллюстрации конкатенации строк достаточно было бы написать `"abc" + "def"`, но я в этом случае пишу что-то вроде `str = "abc" + "def"`. Кому-то это покажется излишеством, но выглядит естественным для программиста на языке C, привыкшего к тому, что бывают функции типа `void` и `non-void` (а также программисту на Pascal, мыслящему в терминах процедур и функций).

Третий каприз заключается в моем нежелании употреблять символ решетки для обозначения методов экземпляра. Многие поклонники Ruby считают, что я проявляю излишнюю болтливость, когда пишу «метод экземпляра `crypt` класса `String`», а не просто `String#crypt`, но я полагаю, что так никто не запутается. (На самом деле мне придется постепенно смириться с использованием такой нотации, так как ясно, что она уже никуда не исчезнет.)

Я старался давать ссылки на внешние ресурсы там, где это уместно. Ограничения по времени и объему не позволили мне включить в книгу все, что я хотел бы, но надеюсь, что это хотя бы отчасти компенсируется указаниями на то, где найти недостающую информацию. Из всех источников самым главным, наверное, следует считать архив приложений Ruby (Ruby Application Archive) в сети; вы не раз встретите ссылки на него.

В начале книги принято приводить соглашения об использовании шрифтов, применяемых для выделения кода, и о том, как отличить пример от обычного текста. Но я не стану оскорблять вас недоверием к вашим умственным способностям, – вы ведь и раньше читали техническую литературу.

Хочу подчеркнуть, что примерно 10% текста книги было написано другими людьми. И это не считая технического редактирования и корректуры!.. Вы просто обязаны прочитать благодарности, приведенные в этой (и любой другой) книге. Большинство читателей пропускают их. Прошу, прочтите прямо сейчас. Это будет так же полезно, как питание овощами.

Об исходных текстах, приведенных в книге

Все сколько-нибудь значительные фрагменты кода собраны в архив, который можно загрузить из сети. Этот архив есть на сайте www.awprofessional.com и на моем собственном сайте (www.rubyhacker.com).

Он предлагается в виде `tgz`-файла и в виде `zip`-файла. При именовании файлов в нем принято следующее соглашение: код, которому в тексте соответствует пронумерованный листинг, находится в файле с таким же именем (например, `listing14-1.rb`). Более короткие фрагменты именуются по номеру страницы, возможно, с добавленной буквой (например, `p260a.rb` и `p260b.rb`). Совсем короткие фрагменты, которые нельзя исполнить «вне контекста», в архиве обычно отсутствуют.

Конец ознакомительного фрагмента.
Приобрести книгу можно
в интернет-магазине
«Электронный универс»
e-Univers.ru