

ОГЛАВЛЕНИЕ

Предисловие.....	15
Благодарности	15
Дэвид Фланаган (David Flanagan)	15
Юкихиро Мацумото (Yukihiro Matsumoto)	16
Способы оформления, принятые в этой книге	16
Глава 1. Введение	17
1.1. Экскурсия по Ruby.....	18
1.1.1. Объектная ориентированность Ruby.....	19
1.1.2. Блоки и итераторы	19
1.1.3. Выражения и операторы Ruby	21
1.1.4. Методы	22
1.1.5. Присваивания	23
1.1.6. Суффиксы и префиксы, состоящие из знаков пунктуации	24
1.1.7. Регулярные выражения и числовые диапазоны	25
1.1.8. Классы и модули	26
1.1.9. Сюрпризы Ruby.....	28
1.2. Опробование Ruby	29
1.2.1. Интерпретатор Ruby.....	29
1.2.2. Отображение вывода	30
1.2.3. Интерактивный Ruby с irb.....	31
1.2.4. Просмотр документации по Ruby с помощью ri.....	32
1.2.5. Управление пакетом программ Ruby с помощью gem	32
1.2.6. Дополнительные учебные пособия по Ruby	34
1.2.7. Источники Ruby.....	34
1.3. О книге.....	34
1.3.1. Как читать эту книгу.....	36
1.4. Решение головоломки Судоку на Ruby	36
Глава 2. Структура и выполнение Ruby-программ	45
2.1. Лексическая структура	46
2.1.1. Комментарии.....	46
2.1.2. Литералы	48
2.1.3. Знаки пунктуации	49
2.1.4. Идентификаторы	49

2.1.5. Ключевые слова.....	51
2.1.6. Разделители.....	52
2.2. Синтаксическая структура.....	55
2.2.1. Блочные структуры языка Ruby.....	56
2.3. Структура файла.....	57
2.4. Кодировка программы.....	58
2.4.1. Объявление кодировки программы.....	59
2.4.2. Кодировка исходного кода и установленная по умолчанию внешняя кодировка.....	60
2.5. Выполнение программы.....	61
Глава 3. Типы данных и объекты	63
3.1. Числа.....	64
3.1.1. Целочисленные литералы.....	65
3.1.2. Литералы чисел с плавающей точкой.....	66
3.1.3. Арифметика, используемая в Ruby.....	66
3.1.4. Двоичное представление чисел с плавающей точкой и ошибки округления.....	68
3.2. Текст.....	69
3.2.1. Строковые литералы.....	69
3.2.2. Символьные литералы.....	78
3.2.3. Строковые операторы.....	79
3.2.4. Получение доступа к символам и подстрокам.....	81
3.2.5. Выполнение итераций в отношении строк.....	83
3.2.6. Кодировка строк и многобайтовые символы.....	84
3.3. Массивы.....	90
3.4. Хэши.....	93
3.4.1. Хэш-литералы.....	94
3.4.2. Хэш-коды, равенство и изменяющиеся ключи.....	94
3.5. Диапазоны.....	95
3.5.1. Проверка принадлежности к диапазону.....	97
3.6. Обозначения.....	98
3.7. True, False и Nil.....	99
3.8. Объекты.....	100
3.8.1. Ссылки на объекты.....	100
3.8.2. Продолжительность существования объекта.....	101
3.8.3. Идентичность объекта.....	102
3.8.4. Класс объекта и тип объекта.....	102
3.8.5. Равенство объектов.....	104
3.8.6. Объект Order.....	107
3.8.7. Преобразование объектов.....	108
3.8.8. Копирование объектов.....	112
3.8.9. Маршализация (Marshaling) объектов.....	113

3.8.10. Замораживание объектов	113
3.8.11. Пометка объектов	114
Глава 4. Выражения и операторы	115
4.1. Простые литералы и литералы ключевых слов	116
4.2. Ссылки на переменные	117
4.2.1. Неинициализированные переменные	117
4.3. Ссылки на константы	118
4.4. Вызовы методов.....	120
4.5. Присваивания	123
4.5.1. Присваивание значений переменным	124
4.5.2. Присваивание значений константам	125
4.5.3. Присваивание значений атрибутам и элементам массива	126
4.5.4. Сокращенная запись присваивания.....	127
4.5.5. Параллельное присваивание.....	129
4.6. Операторы	133
4.6.1. Унарные операторы + и -.....	135
4.6.2. Возведение в степень: **.....	136
4.6.3. Арифметические операторы: +, -, *, / и %	136
4.6.4. Сдвиг и добавление: << и >>	136
4.6.5. Дополнение, объединение, пересечение: ~, &, и ^	137
4.6.6. Сравнение: <, <=, >, >= и <=>.....	138
4.6.7. Равенство: ==, !=, =~, !~ и ===	139
4.6.8. Булевы операторы: &&, , !, and, or, not	140
4.6.9. Диапазоны и триггеры: .. и	143
4.6.10. Условный оператор: ?:.....	145
4.6.11. Операторы присваивания.....	147
4.6.12. Оператор defined?.....	147
4.6.13. Операторы-модификаторы	149
4.6.14. Что не относится к операторам.....	149
Глава 5. Инструкции и управляющие структуры	151
5.1. Условия.....	152
5.1.1. If	152
5.1.2. Работа if в качестве модификатора.....	155
5.1.3. Unless.....	157
5.1.4. Case	158
5.1.5. Оператор ?:.....	162
5.2. Циклы	162
5.2.1. While и until.....	162
5.2.2. While и until в качестве модификаторов	163
5.2.3. Цикл for-in	165
5.3. Итераторы и перечисляемые объекты.....	166
5.3.1. Числовые итераторы.....	168
5.3.2. Перечисляемые объекты	168

5.3.3. Создание собственных итераторов.....	170
5.3.4. Нумераторы	172
5.3.5. Внешние итераторы	174
5.3.6. Итерация и параллельное изменение	177
5.4. Блоки.....	178
5.4.1. Синтаксис блока.....	178
5.4.2. Значение блока	179
5.4.3. Блоки и область видимости переменных	180
5.4.4. Передача аргументов блоку.....	181
5.5. Изменение хода работы программы.....	184
5.5.1. Return	184
5.5.2. Break	186
5.5.3. Next	188
5.5.4. Redo	190
5.5.5. Retry	191
5.5.6. Throw и catch.....	192
5.6. Исключения и их обработка	194
5.6.1. Классы и объекты исключений.....	194
5.6.2. Выдача исключений с помощью raise	197
5.6.3. Обработка исключений с помощью rescue.....	198
5.6.4. Предложение else	203
5.6.5. Предложение ensure.....	204
5.6.6. Rescue в определениях метода, класса и модуля	206
5.6.7. Rescue в качестве модификатора инструкции.....	206
5.7. BEGIN и END	207
5.8. Потоки, нити (fibers) и продолжения (continuations).....	208
5.8.1. Потоки для параллельного выполнения	209
5.8.2. Нити для сопрограмм	209
5.8.3. Продолжения.....	215

Глава 6. Методы, proc- и lambda-объекты и замкнутые выражения 217

6.1. Определение простых методов	219
6.1.1. Значение, возвращаемое методом.....	219
6.1.2. Методы и обработка исключений	220
6.1.3. Вызов метода для объекта.....	221
6.1.4. Определение синглтон-методов	221
6.1.5. Отмена определения методов.....	222
6.2. Имена методов	223
6.2.1. Методы-операторы.....	224
6.2.2. Псевдонимы методов.....	224
6.3. Методы и круглые скобки	226
6.3.1. Необязательные скобки.....	226
6.3.2. Обязательные скобки	227

6.4. Аргументы метода.....	228
6.4.1. Параметры по умолчанию	229
6.4.2. Список аргументов переменной длины и массивы.....	230
6.4.3. Отображение аргументов на параметры.....	232
6.4.4. Использование хэшей для поименованных аргументов.....	232
6.4.5. Блоки-аргументы.....	233
6.5. Proc и lambda.....	237
6.5.1. Создание Proc-объектов.....	237
6.5.2. Вызов объектов Proc и Lambda.....	240
6.5.3. Аридность Proc.....	241
6.5.4. Идентичность Proc-объектов.....	242
6.5.5. Чем lambda-объекты отличаются от proc-объектов.....	242
6.6. Замкнутые выражения.....	246
6.6.1. Замкнутые выражения и совместно используемые переменные.....	247
6.6.2. Замкнутые выражения и связывания.....	249
6.7. Объекты класса Method	250
6.7.1. Несвязанные объекты метода.....	251
6.8. Функциональное программирование	252
6.8.1. Применение функции к перечисляемым объектам.....	252
6.8.2. Составление функций.....	254
6.8.3. Частично применяемые функции.....	255
6.8.4. Функции, обладающие мемоизацией.....	256
6.8.5. Классы Symbol, Method и Proc	257
Глава 7. Классы и модули	261
7.1. Определение элементарного класса.....	262
7.1.1. Создание класса.....	263
7.1.2. Создание экземпляра класса Point.....	263
7.1.3. Инициализация класса Point.....	264
7.1.4. Определение метода to_s.....	265
7.1.5. Методы доступа и атрибуты	266
7.1.6. Определение операторов.....	268
7.1.7. Доступ к массивам и хэшам с помощью метода []	271
7.1.8. Перечисление координат	271
7.1.9. Равенство точек.....	272
7.1.10. Упорядочение Point-объектов.....	274
7.1.11. Изменяющийся Point-объект.....	275
7.1.12. Быстрое и простое создание изменяющихся классов.....	276
7.1.13. Метод класса	278
7.1.14. Константы	280
7.1.15. Переменные класса	280
7.1.16. Переменные экземпляра класса	281
7.2. Область видимости методов: открытые, защищенные и закрытые методы..	283
7.3. Подклассы и наследование.....	286

7.3.1. Наследуемые методы.....	287
7.3.2. Переопределение методов	288
7.3.3. Дополнение поведения путем выстраивания цепочки.....	290
7.3.4. Наследование методов класса	291
7.3.5. Наследование и переменные экземпляра.....	291
7.3.6. Наследование и переменные класса.....	292
7.3.7. Наследование констант	293
7.4. Создание и инициализация объектов.....	294
7.4.1. New, allocate и initialize.....	294
7.4.2. Фабричные методы	295
7.4.3. Dup, clone и initialize_copy	296
7.4.4. Marshal_dump и marshal_load.....	298
7.4.5. Шаблон Синглтон (Singleton)	299
7.5. Модули	301
7.5.1. Модули как пространства имен.....	301
7.5.2. Использование модулей в качестве миксинов	304
7.5.3. Включаемые модули пространства имен	306
7.6. Загрузка и востребование модулей	307
7.6.1. Путь загрузки.....	308
7.6.2. Выполнение загруженного кода	310
7.6.3. Автоматически загружаемые модули	311
7.7. Синглтон-методы и обособленные классы (eigenclass).....	312
7.8. Поиск метода.....	314
7.8.1. Поиск метода класса.....	315
7.9. Поиск констант.....	317
Глава 8. Отражение и метапрограммирование	319
8.1. Типы, классы и модули	321
8.1.1. Предки и модули.....	321
8.1.2. Определение классов и модулей	323
8.2. Вычисление строк и блоков	323
8.2.1. Связывания и метод eval.....	323
8.2.2. Instance_eval и class_eval.....	324
8.2.3. Instance_exec и class_exec	325
8.3. Переменные и константы.....	325
8.3.1. Запрос, установка и проверка переменных	326
8.4. Методы	328
8.4.1. Вывод имен и проверка наличия методов.....	328
8.4.2. Получение объектов метода.....	329
8.4.3. Вызов Method-объектов	329
8.4.4. Методы определения, отмены определения и присвоения псевдонимов.....	330
8.4.5. Обработка неопределенных методов	332
8.4.6. Установка видимости метода.....	333

8.5. Перехватчики	333
8.6. Трассировка	335
8.7. Модули ObjectSpace и GC	337
8.8. Создание своих собственных управляющих структур	338
8.8.1. Отложенное и повторяющееся выполнение: after и every	338
8.8.2. Безопасное применение потоков путем синхронизации блоков	340
8.9. Ненайденные методы и константы	341
8.9.1. Работа const_missing с кодовыми константами Юникода	341
8.9.2. Отслеживание вызовов методов с помощью метода method_missing	342
8.9.3. Объекты, синхронизированные благодаря делегированию	344
8.10. Динамически создаваемые методы	345
8.10.1. Определение методов с помощью class_eval	346
8.10.2. Определение методов с помощью define_method	347
8.11. Выстраивание цепочки псевдонимов	348
8.11.1. Отслеживание загрузки файлов и определение новых классов	349
8.11.2. Выстраивание цепочки методов для обеспечения безопасности работы потоков	351
8.11.3. Выстраивание цепочки методов для осуществления отслеживания	354
8.12. Предметно-ориентированные языки	356
8.12.1. Простой вывод XML с помощью метода method_missing	356
8.12.2. Проверяемый вывод XML-кода с помощью генерации метода	358
Глава 9. Платформа Ruby	363
9.1. Строки	364
9.1.1. Форматирование текста	369
9.1.2. Упаковка и распаковка двоичных строк	370
9.1.3. Строки и кодировки	371
9.2. Регулярные выражения	371
9.2.1. Литералы регулярных выражений	371
9.2.2. Фабричные методы регулярных выражений	373
9.2.3. Синтаксис регулярных выражений	374
9.2.4. Определение соответствия шаблону с использованием регулярных выражений	378
9.3. Числа и математические операции	385
9.3.1. Числовые методы	385
9.3.2. Модуль Math	386
9.3.3. Десятичная арифметика	387
9.3.4. Комплексные числа	388
9.3.5. Рациональные числа	388
9.3.6. Векторы и матрицы	389
9.3.7. Случайные числа	390
9.4. Работа с датой и временем	390
9.5. Коллекции	393
9.5.1. Перечисляемые объекты	393
9.5.2. Массивы	401

9.5.3. Хэши	408
9.5.4. Наборы	414
9.6. Файлы и каталоги	419
9.6.1. Имена файлов и каталогов	420
9.6.2. Вывод содержимого каталогов	422
9.6.3. Проверка файлов	423
9.6.4. Создание, удаление и переименование файлов	425
9.7. Ввод и вывод данных	426
9.7.1. Открытие потоков	426
9.7.2. Потоки и кодировки	429
9.7.3. Чтение из потока	431
9.7.4. Запись в поток	435
9.7.5. Методы произвольного доступа	437
9.7.6. Закрытие, сброс и тестирование потоков	437
9.8. Работа в сети	439
9.8.1. Самый простой клиент	439
9.8.2. Самый простой сервер	440
9.8.3. Датаграммы	441
9.8.4. Более сложный пример клиента	442
9.8.5. Мультиплексированный сервер	443
9.8.6. Извлечение веб-страниц	445
9.9. Потоки и параллельные вычисления	446
9.9.1. Время существования потоков	448
9.9.2. Потоки и переменные	449
9.9.3. Диспетчеризация потоков	451
9.9.4. Состояния потоков	452
9.9.5. Составление списка потоков и их группы	455
9.9.6. Примеры организации потоков	455
9.9.7. Исключение потоков и взаимная блокировка	458
9.9.8. Очередь, и очередь определенного размера	461
9.9.9. Переменные условий и очереди	462
Глава 10. Среда окружения Ruby	465
10.1. Вызов Ruby-интерпретатора	466
10.1.1. Наиболее востребованные ключи	467
10.1.2. Ключи, связанные с предупреждениями и выдачей информации	468
10.1.3. Ключи, относящиеся к кодировке	468
10.1.4. Ключи, связанные с обработкой текста	469
10.1.5. Ключи разного назначения	470
10.2. Высокоуровневое окружение	471
10.2.1. Предопределенные модули и классы	471
10.2.2. Высокоуровневые константы	472
10.2.3. Глобальные переменные	474

10.2.4. Предопределенные глобальные функции.....	478
10.2.5. Глобальные функции, определенные пользователями	480
10.3. Сокращения для удобства извлечения данных и составления отчетов	481
10.3.1. Функции ввода.....	481
10.3.2. Нерекомендуемые функции извлечения данных.....	482
10.3.3. Функции составления отчетов	482
10.3.4. Сокращения, используемые в однострочных сценариях	483
10.4. Обращение к операционной системе.....	483
10.4.1. Вызов команд операционной системы.....	484
10.4.2. Процессы и ветвления	485
10.4.3. Отлавливание сигналов	487
10.4.4. Прерывание работы программ.....	487
10.5. Безопасность	488
10.5.1. Помеченные данные	489
10.5.2. Ограниченное выполнение и уровни безопасности.....	489