

УДК 004.438Erlang
ББК 32.973.26-018.2

T56

Томпсон С., Чезарини Ф.

Программирование в Erlang/ Пер. с англ. Холомьёва А. О. – М.: ДМК Пресс, 2019. – 488 с.: ил. (Серия "Функциональное программирование")

ISBN 978-5-97060-721-3

Эта книга познакомит вас с Erlang, языком программирования, который идеально подходит для создания параллельных отказоустойчивых приложений, предназначенных для работы в режиме реального времени. С появлением многоядерных процессоров, предоставляющих новые средства масштабирования распараллеленных приложений, Erlang становится все более популярным. Независимо от уровня предварительной подготовки, вы сможете научиться написанию сложных параллельных приложений.

Книга написана лидерами международного сообщества Erlang. В ней вы найдете подробное описание правильно построенных списков, рекурсии, отладки, сетевого и параллельного программирования, а также многих других аспектов синтаксиса и семантики языка Erlang.

УДК 004.438Erlang
ББК 32.973.26-018.2

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-0-596-51818-9 (анг.)

ISBN 978-5-97060-721-3 (рус.)

Copyright © Francesco Cesarini and
Simon Thompson. All rights reserved

© Перевод на русский язык, оформление,
ДМК Пресс

Оглавление

Предисловие к русскому изданию	14
Вступительное слово	15
Предисловие	16
Франческо: Почему Erlang?	16
Саймон: Почему Erlang?	17
Для кого эта книга?	17
Как читать эту книгу	17
Условные обозначения	19
Использование кода из примеров	20
Safari Books Online	20
Как связаться с нами	20
Благодарности	21
Благодарности от российского издательства	22
1 Введение	23
1.1 Почему Erlang?	23
1.2 История Erlang	25
1.3 Особенности Erlang	26
Высокоуровневые конструкции	26
Параллельные вычисления и передача сообщений	27
Масштабируемые, безопасные и эффективные параллельные вычисления	28
Применение в системах реального времени	29
Надёжность	29
Распределённые вычисления	30
Интеграция и открытость	31
1.4 Erlang и многоядерные процессоры	31
1.5 Erlang на практике	32
Коммутатор ATM AXD301	33
CouchDB	34
Erlang и C++	35
1.6 Как применять Erlang?	36

2	Основы Erlang	38
2.1	Целые числа	38
2.2	Интерпретатор Erlang	39
2.3	Действительные числа	39
	Арифметические операции	40
2.4	Атомы	41
2.5	Логические значения	43
2.6	Кортежи	44
2.7	Списки	45
	Символы и строки	46
	Атомы и строки	46
	Создание и обработка списков	47
	Функции и операции, определённые на списках	48
2.8	Сравнение термов	51
2.9	Переменные	53
2.10	Сложные структуры данных	55
2.11	Сопоставление с образцом	57
2.12	Функции	62
2.13	Модули	65
	Компиляция и виртуальная машина Erlang	66
	Директивы модулей	66
2.14	Упражнения	68
3	Последовательное программирование в Erlang	70
3.1	Условные выражения	71
	case-выражение	71
	Область видимости переменных	74
	if-выражение	75
3.2	Охранные выражения	76
3.3	Встроенные функции	78
	Извлечение элементов объекта и формирование запросов к объектам	79
	Преобразование типов	80
	Словарь процессов	81
	Метапрограммирование	81
	Процессы, порты, распределённые программы и системная информация	82
	Ввод-вывод	83
3.4	Рекурсия	85
	Хвостовая рекурсия	90
	Оптимизация хвостовой рекурсии	93
	Итерационные функции против рекурсивных	94

3.5	Ошибки времени выполнения	95
3.6	Обработка ошибок	98
	С помощью try ... catch	98
	С помощью catch	102
3.7	Библиотечные модули	106
	Документация	106
	Полезные модули	107
3.8	Отладчик	109
3.9	Упражнения	111
4	Параллельное программирование	117
4.1	Создание процессов	118
4.2	Передача сообщений	120
4.3	Приём сообщений	122
4.4	Выборочный приём сообщений	124
4.5	Пример эхо-процесса	128
4.6	Зарегистрированные процессы	130
4.7	Время задержки	133
4.8	Анализ производительности	135
4.9	Каркас процессов	136
4.10	Хвостовая рекурсия и утечки памяти	137
4.11	Один случай из практики параллельного программирования	138
4.12	Состояние гонки, взаимная блокировка, голодание процессов	140
4.13	Менеджер процессов	142
4.14	Упражнения	143
5	Шаблоны проектирования процессов	145
5.1	Модель клиент/сервер	146
	Пример модели клиент/сервер	147
5.2	Пример шаблона процессов	152
5.3	Конечный автомат	154
	Пример конечного автомата	154
	Одноместный семафор	156
5.4	Менеджер и обработчик событий	159
	Пример обобщённого менеджера событий	160
	Обработчики событий	163
5.5	Упражнения	165
6	Обработка ошибок в процессах	167
6.1	Соединение процессов и сигналы выхода	167
	Перехват сигналов выхода	170
	Функции наблюдения за процессами	172

Функции останова	174
Встроенные функции и терминология	174
Семантика распространения сигналов выхода	176
6.2 Построение надёжных систем	177
Наблюдение за клиентами	178
Пример процесса-наблюдателя	181
6.3 Упражнения	183
7 Записи и макросы	186
7.1 Записи	186
Записи, первое знакомство	187
Применение записей	188
Функции и сопоставление с образцом	189
Записи в интерпретаторе	190
Реализация записей	191
Встроенные функции для работы с записями	192
7.2 Макросы	193
Простые макросы	193
Параметризованные макросы	194
Отладка и макросы	195
Заголовочные файлы	196
7.3 Упражнения	197
8 Обновление приложений	200
8.1 Обновление модулей	200
8.2 За кулисами	203
Загрузка кода	206
Сервер кода	207
Очищение модулей	209
8.3 Обновление процессов	209
8.4 Файл .erlang	214
8.5 Упражнения	214
9 Новые типы данных и высокоуровневые выражения	216
9.1 Функциональное программирование	216
9.2 Тип fun и функции высшего порядка	217
Функция как аргумент	217
Определение функций на лету: fun-выражения	219
Функция как результат	220
Использование определённых функций	221
Функции и переменные	222
Стандартные функции высшего порядка	223

Ленивые вычисления и списки	224
9.3 Генераторы списков	225
Первый пример	225
Обобщённые генераторы списков	225
Несколько генераторов	227
Стандартные функции	227
9.4 Двоичные данные и сериализация	228
Двоичные данные	228
Битовый синтаксис	230
Битовое сопоставление с образцом	232
Генераторы двоичных данных	233
Пример: декодирование сегментов TCP	234
Битовые операции	235
Сериализация	236
9.5 Ссылки	238
9.6 Упражнения	239
10 ETS- и Dets-таблицы	241
10.1 ETS-таблицы	241
Реализация и оптимальный выбор типа таблиц	242
Создание таблиц	244
Работа с таблицей	245
Пример: построение предметного указателя, первая часть	246
Обход таблицы	249
Пример: построение предметного указателя, вторая часть	249
Извлечение данных: match	251
Извлечение данных: select	253
Другие операции с таблицами	254
Записи и ETS-таблицы	255
Визуализация таблиц	256
10.2 Dets-таблицы	256
10.3 Пример: база данных абонентов мобильной связи	259
Интерфейс базы данных	260
Сервер базы данных	266
10.4 Упражнения	271
11 Распределённое программирование	273
11.1 Распределённые приложения в Erlang	273
11.2 Распределённые приложения в Erlang: основы	275
Имена узлов и область видимости	277
Взаимодействие и безопасность	278
Взаимодействие и сообщения	280

Соединение узлов	281
Удалённый вызов процедур	284
Модуль gpc	286
Ключевые модули для распределённого программирования	287
11.3 Процесс ermd	288
Распределённый Erlang за межсетевым экраном	289
11.4 Упражнения	290
12 Поведения OTP	291
12.1 Введение в поведения OTP	291
12.2 Обобщённый сервер	294
Запуск сервера	294
Передача сообщений	296
Завершение сервера	298
Полный текст примера	299
Тестирование gen_server	302
12.3 Наблюдатель	305
Спецификация наблюдателя	306
Спецификация дочерних процессов	306
Пример	308
Динамические дочерние процессы	309
12.4 Приложения	309
Структура директорий	310
Resource-файл	312
Запуск и завершение приложений	313
Менеджер приложений	316
12.5 Управление релизами	316
12.6 Другие поведения и источники для самостоятельного изучения	319
12.7 Упражнения	320
13 Начала работы с Mnesia	322
13.1 Для чего подходит Mnesia	322
13.2 Настройка Mnesia	324
Задание схемы	324
Запуск Mnesia	325
Таблицы Mnesia	325
13.3 Транзакции	328
Запись	329
Чтение и удаление	329
Индексация	330
Грязные операции	332
13.4 Разрыв соединения	334

13.5	Дополнительные источники информации	335
13.6	Упражнения	335
14	Создание GUI-приложений средствами wxErlang	337
14.1	wxWidgets	337
14.2	wxErlang. Порт wxWidgets для Erlang	338
	Объекты и типы	339
	Обработка событий, идентификаторы объектов и типы событий	339
	Соберём всё вместе	341
14.3	Первый пример: микроблог	342
14.4	Мини-блог	344
14.5	Установка и запуск wxErlang	348
14.6	Упражнения	350
15	Работа с сокетами	351
15.1	User Datagram Protocol	351
15.2	Transmission Control Protocol	355
	Пример обмена данными через TCP	356
15.3	Модуль inet	360
15.4	Дополнительные источники информации	361
15.5	Упражнения	362
16	Взаимодействие с другими языками программирования	364
16.1	Обзор средств взаимодействия	364
16.2	Взаимодействие с Java	366
	Узлы и почтовые ящики	366
	Представление типов Erlang	367
	Обмен сообщениями	367
	Соберём всё вместе: и вновь RPC	368
	Взаимодействие	369
	Мелким шрифтом	370
	Но и это ещё не всё	370
16.3	Узлы C	371
	Дополнительные возможности	375
16.4	Вызов Erlang из командной строки UNIX: erl_call	375
16.5	Порты	375
	Команды управления	377
	Обмен данными	378
16.6	Библиотечная поддержка обмена данными	379
	Работаем с Ruby: electricity	380
16.7	Подключаемые драйверы и FFI	381
16.8	Упражнения	382

17 Отладка приложений в Erlang	383
17.1 Введение	383
17.2 Трассировочные встроенные функции	385
Трассировочные флаги для процессов	386
Флаги наследования	388
Сборка мусора и отсчёты времени	389
17.3 Трассировка вызовов с помощью функции <code>trace_pattern</code>	391
17.4 Трассировщик <code>dbg</code>	394
Первые шаги	394
Трассировка и профилирование функций	397
Трассировка локальных и глобальных вызовов функций	398
Распределённая трассировка	399
Перенаправление вывода	400
17.5 Спецификация сопоставления: <code>fun</code> -синтаксис	404
Создание спецификаций функцией <code>fun2ms</code>	404
Разница между спецификациями для <code>ets</code> и <code>dbg</code>	412
17.6 Устройство спецификаций сопоставления	412
Голова	413
Условные выражения	414
Тело спецификации	417
Сохранение спецификаций сопоставления	421
17.7 Дополнительная литература	422
17.8 Упражнения	422
18 Типы и документация	424
18.1 Типы в Erlang	424
Пример: записи с типизированными полями	424
Нотация типов в Erlang	425
18.2 <code>TypeEr</code>	427
Dialyzer: утилита для статической проверки кода Erlang	430
18.3 Создание документации с помощью EDoc	431
Документация к модулю <code>usr_db</code>	432
Запуск EDoc	434
Типы в EDoc	436
Дополнительные возможности EDoc	437
18.4 Упражнения	439
19 EUnit и разработка через тестирование	440
19.1 Разработка через тестирование	440
19.2 EUnit	441
Как пользоваться EUnit	442
Функциональное тестирование: сериализация дерева	442

19.3 Инфраструктура EUnit	446
Макрос assert	446
Функции для генерации тестов	446
Представление тестов в EUnit	447
19.4 Тестирование приложений с состоянием	448
Окружения: инициализация и очистка	448
19.5 Тестирование параллельных программ	449
19.6 Упражнения	450
20 Стиль и эффективность	451
20.1 Приложения и модули	451
Библиотеки	452
Грязный код	452
Интерфейсы	453
Возвращаемые значения	453
Внутренние структуры данных	455
20.2 Процессы и параллельные вычисления	456
20.3 Стилиевые соглашения	460
20.4 Стратегии разработки	465
20.5 Эффективность	467
Последовательное программирование	467
Списки	469
Хвостовая и нехвостовая рекурсия	471
Параллельное программирование	471
20.6 И наконец...	473
Приложение	475
Основы работы с Erlang	475
Установка	475
Запуск интерпретатора	475
Средства разработки	477
Редакторы	477
Другие средства	478
Дополнительные источники информации	479
Об авторах	481