

УДК 004.432.42Clojure
ББК 32.973-018.1
Э54

Эмерик, Чаз.

Э54 Программирование на Clojure. Практика применения Lisp в мире Java / Ч. Эмерик, Б. Карпер, К. Гранд ; пер. с англ. А. Н. Киселёва. — 2-е изд., эл. — 1 файл pdf : 817 с. — Москва : ДМК Пресс, 2023. — Систем. требования: Adobe Reader XI либо Adobe Digital Editions 4.5 ; экран 10". — Текст : электронный.

ISBN 978-5-89818-600-5

Почему многие выбирают Clojure? Потому что это функциональный язык программирования, не только позволяющий использовать Java-библиотеки, службы и другие ресурсы JVM, но и соперничающий с другими динамическими языками, такими как Ruby и Python.

Эта книга продемонстрирует вам гибкость Clojure в решении типичных задач, таких как разработка веб-приложений и взаимодействие с базами данных. Вы быстро поймете, что этот язык помогает устранить ненужные сложности в своей практике и открывает новые пути решения сложных проблем, включая многопоточное программирование.

Издание предназначено для программистов, желающих освоить всю мощь и гибкость функционального программирования.

УДК 004.432.42Clojure
ББК 32.973-018.1

Электронное издание на основе печатного издания: Программирование на Clojure. Практика применения Lisp в мире Java / Ч. Эмерик, Б. Карпер, К. Гранд ; пер. с англ. А. Н. Киселёва. — Москва : ДМК Пресс, 2015. — 816 с. — ISBN 978-5-97060-299-7. — Текст : непосредственный.

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.

В соответствии со ст. 1299 и 1301 ГК РФ при устранении ограничений, установленных техническими средствами защиты авторских прав, правообладатель вправе требовать от нарушителя возмещения убытков или выплаты компенсации.

ISBN 978-5-89818-600-5

© Chas Emerick, Brian Carper, and
Christophe Grand
© Оформление, перевод ДМК Пресс, 2015



Содержание

Предисловие к русскому изданию	15
Благодарности	16
Предисловие	17
Глава 1. Вниз по кроличьей норе	26
Почему Clojure?	26
Как получить Clojure	29
Интерактивная оболочка REPL для Clojure	30
Вам не придется путаться в частоколе скобок	34
Выражения, операторы, синтаксис и очередность	35
Гомоиконность	38
Механизм чтения	41
Скалярные литералы	43
Строки	43
Логические значения	43
nil	43
Знаки (characters)	44
Ключевые слова (keywords)	44
Символы (symbols)	46
Числа	46
Регулярные выражения	48
Комментарии	49
Пробелы и запятые	51
Литералы коллекций	51
Прочий синтаксический сахар механизма чтения	52
Пространства имен	53
Интерпретация символов	56
Специальные формы	57
Подавление вычислений: quote	59
Блоки кода: do	60
Определение переменных: def	61

Связывание локальных значений: let	62
Деструктуризация (let, часть 2).....	64
Деструктуризация упорядоченных коллекций	65
Деструктуризация ассоциативных массивов	69
Создание функций: fn.....	74
Деструктуризация аргументов функций	77
Литералы функций	80
Выполнение по условию: if	82
Организация циклов: loop и recur	83
Ссылки на переменные: var	85
Взаимодействие с Java: . и new	86
Обработка исключений: try и throw	86
Специализированная операция set!	87
Примитивы блокировок: monitor-enter и monitor-exit	87
Все вместе	87
eval.....	88
Это лишь начало.....	90

Часть I. ФУНКЦИОНАЛЬНОЕ И КОНКУРЕНТНОЕ ПРОГРАММИРОВАНИЕ 91

Глава 2. Функциональное программирование 92

Что подразумевается под термином «Функциональное программирование»?	93
О важности значений.....	94
О значениях	95
Сравнение значений изменяемых объектов	96
Важность выбора	101
Функции, как сущности первого порядка, и функции высшего порядка	103
Частичное применение	111
Композиция функций	116
Создание функций высшего порядка.....	120
Создание простейшей системы журналирования с применением композиции функций высшего порядка	121
Чистые функции	126
В чем преимущество чистых функций?	129
Функциональное программирование в реальном мире	132

Глава 3. Коллекции и структуры данных 134

Главенство абстракций над реализациями	135
Коллекции	139
Последовательности	142

Последовательности не являются итераторами	145
Последовательности не являются списками	146
Создание последовательностей	147
Ленивые последовательности	148
Удержание мусора	156
Ассоциативные коллекции	157
Берегитесь значения nil	161
Индексирование	162
Стек	164
Множество	165
Сортированные коллекции	166
Определение порядка с помощью компараторов и предикатов	168
Упрощенный доступ к коллекциям	173
Идиоматические приемы использования	175
Коллекции, ключи и функции высшего порядка	176
Типы структур данных	177
Списки	178
Векторы	179
Векторы как кортежи	180
Множества	181
Ассоциативные массивы	182
Ассоциативные массивы как специализированные структуры	183
Другие применения ассоциативных массивов	185
Неизменяемость и сохранность	189
Сохранность и совместное использование	190
Визуализация сохранности: списки	191
Визуализация сохранности: ассоциативные массивы (векторы и множества)	193
Очевидные преимущества	196
Переходные структуры данных	198
Метаданные	205
Включаем коллекции Clojure в работу	207
Идентификаторы и циклы	208
Думайте иначе: от императивного к функциональному	210
Вспоминаем классику: игра «Жизнь»	211
Генерация лабиринтов	220
Навигация, изменение и зипперы (zippers)	228
Манипулирование зипперами	229
Собственные зипперы	231
Зиппер Ариадны	232
В заключение	236

Глава 4. Конкуренция и параллелизм	237
Сдвиг вычислений в пространстве и времени	238
delay	238
Механизм future	241
Механизм promise	243
Параллельная обработка по невысокой цене	246
Состояние и идентичность	250
Ссылочные типы	253
Классификация параллельных операций	255
Атомы	258
Уведомление и ограничение	261
Функции-наблюдатели	261
Функции-валидаторы	264
Ссылки	266
Программная транзакционная память	266
Механика изменения ссылок	268
Функция alter	271
Уменьшение конфликтов в транзакциях с помощью commute	273
Затирание состояния ссылки с помощью ref-set	279
Проверка локальной согласованности с помощью валидаторов	279
Острые углы программной транзакционной памяти	283
Функции с побочными эффектами строго запрещены	283
Минимизируйте продолжительность выполнения транзакций	284
Читающие транзакции могут повторяться	287
Искажение при записи	289
Переменные	291
Определение переменных	292
Приватные переменные	293
Строки документации	294
Константы	295
Динамическая область видимости	296
Переменные в языке Clojure не являются переменными в классическом понимании	303
Опережающие объявления	305
Агенты	307
Обработка ошибок в заданиях агентов	310
Режимы и обработчики ошибок в агентах	312
Ввод/вывод, транзакции и вложенная передача заданий	313
Сохранение состояний ссылок в журнале на основе агента	315
Использование агентов для распределения нагрузки	318
Механизмы параллельного выполнения в Java	328
Блокировки	329
В заключение	330

Часть II. СОЗДАНИЕ АБСТРАКЦИЙ	331
Глава 5. Макросы	332
Что такое макрос?	333
Чем не являются макросы	335
Что могут макросы, чего не могут функции?	336
Сравнение макросов и механизма eval в Ruby	339
Пишем свой первый макрос	341
Отладка макросов	343
Функции развертывания макросов	344
Синтаксис	346
Сравнение quote и syntax-quote	348
unquote и unquote-splicing	349
Когда следует использовать макросы	351
Гигиена	353
Генераторы символов во спасение	355
Предоставление пользователю права выбора имен	359
Двукратное вычисление	360
Распространенные идиомы и шаблоны макросов	362
Неявные аргументы: &env и &form	364
&env	364
&form	367
Вывод сообщений об ошибках в макросах	367
Сохранение определений типов, сделанных пользователем	370
Тестирование контекстных макросов	373
Подробности: -> и ->>	375
В заключение	379
Глава 6. Типы данных и протоколы	380
Протоколы	381
Расширение существующих типов	383
Определение собственных типов	389
Записи	392
Конструкторы и фабричные функции	396
Когда использовать ассоциативные массивы, а когда записи ...	398
Типы	399
Реализация протоколов	402
Встроенная реализация	403
Встроенные реализации интерфейсов Java	405
Определение анонимных типов с помощью reify	407
Повторное использование реализаций	408
Интроспекция протоколов	413
Пограничные случаи использования протоколов	415

Поддержка абстракций коллекций.....	417
В заключение	427
Глава 7. Мультиметоды.....	428
Основы мультиметодов	428
Навстречу иерархиям	431
Иерархии	434
Независимые иерархии.....	437
Сделаем выбор по-настоящему множественным!	441
Кое что еще	443
Множественное наследование	443
Интроспекция мультиметодов	445
type и class; или месть ассоциативного массива	446
Функции выбора не имеют ограничений.....	447
В заключение	449
Часть III. ИНСТРУМЕНТЫ, ПЛАТФОРМЫ И ПРОЕКТЫ	450
Глава 8. Создание и организация проектов на Clojure	451
География проекта	451
Определение и использование пространств имен.....	452
Пространства имен и файлы	461
Знакомство с classpath	465
Местоположение, местоположение, местоположение	467
Организация программного кода по функциональным признакам ...	469
Основные принципы организации проектов	471
Сборка	472
Предварительная компиляция.....	473
Управление зависимостями	476
Модель Maven управления зависимостями	477
Артефакты и координаты.....	477
Репозитории	479
Зависимости.....	480
Инструменты сборки и шаблоны настройки.....	483
Maven.....	484
Leiningen	488
Настройка предварительной компиляции	491
Сборка гибридных проектов.....	493
В заключение	496
Глава 9. Java и взаимодействие с JVM.....	497
JVM – основа Clojure.....	498
Использование классов, методов и полей Java	499

Удобные утилиты взаимодействий	503
Исключения и обработка ошибок	506
Отказ от контролируемых исключений	509
with-open, прощай finally	510
Указание типов для производительности	512
Массивы	518
Определение классов и реализация интерфейсов	519
Экземпляры анонимных классов: proху	520
Определение именованных классов	523
gen-class	524
Аннотации	532
Создание аннотированных тестов для JUnit	533
Реализация конечных точек веб-службы JAX-RS	534
Использование Clojure из Java	537
Использование классов, созданных с помощью deftype и defrecord	541
Реализация интерфейсов протоколов	544
Сотрудничество	546

Глава 10. REPL-ориентированное программирование

Интерактивная разработка	547
Постоянное изменяющееся окружение	552
Инструменты	554
Оболочка REPL	555
Интроспекция пространств имен	557
Eclipse	560
Emacs	563
clojure-mode и paredit	564
inferior-lisp	565
SLIME	567
Отладка, мониторинг и исправление программ в REPL во время эксплуатации	570
Особые замечания по поводу «развертываемых» оболочек REPL	574
Ограничения при переопределении конструкций	576
В заключение	579

Часть IV. ПРАКТИКУМ

Глава 11. Числовые типы и арифметика

Числовые типы в Clojure	581
В Clojure предпочтение отдается 64-битным (или больше) представлениям	583
Clojure имеет смешанную модель числовых типов	583

Рациональные числа	586
Правила определения типа результата	587
Арифметика в Clojure	589
Ограниченная и произвольная точность	589
Неконтролируемые операции	593
Режимы масштабирования и округления в операциях с вещественными числами произвольной точности.....	595
Равенство и эквивалентность	597
Идентичность объектов (identical?).....	597
Равенство ссылок (=).....	598
Числовая эквивалентность (==)	600
Эквивалентность может защитить ваш рассудок.....	601
Оптимизация производительности операций с числами	603
Объявление функций, принимающих и возвращающих значения простых типов	604
Ошибки и предупреждения, вызванные несоответствием типов	608
Используйте простые массивы осмысленно.....	610
Механика массивов значений простых типов	613
Автоматизация указания типов в операциях с многомерными массивами	618
Визуализация множества Мандельброта в Clojure	620
Глава 12. Шаблоны проектирования	629
Внедрение зависимостей.....	631
Шаблон Стратегия (Strategy)	636
Цепочка обязанностей (Chain of Responsibility)	638
Аспектно-ориентированное программирование	642
В заключение	647
Глава 13. Тестирование.....	648
Неизменяемые значения и чистые функции	648
Создание фиктивных значений.....	649
clojure.test	651
Определение тестов	653
«Комплекты» тестов	656
Крепления (fixtures)	658
Расширение HTML DSL.....	662
Использование контрольных проверок.....	668
Предусловия и постусловия	670
Глава 14. Реляционные базы данных	673
clojure.java.jdbc	673
Подробнее о with-query-results	678

Транзакции.....	680
Пулы соединений	681
Korma.....	682
Вступление	683
Запросы	685
Зачем использовать предметно-ориентированный язык?	686
Hibernate	689
Настройка	690
Сохранение данных.....	694
Выполнение запросов.....	695
Избавление от шаблонного кода	695
В заключение	698
Глава 15. Нереляционные базы данных.....	699
Настройка CouchDB и Clutch	700
Простейшие CRUD-операции	701
Представления.....	703
Простое представление (на JavaScript)	704
Представления на языке Clojure	706
_changes: использование CouchDB в роли очереди сообщений	711
Очереди сообщений на заказ	713
В заключение	717
Глава 16. Clojure и Веб.....	718
«Стек Clojure»	718
Основа: Ring.....	720
Запросы и ответы.....	721
Адаптеры	724
Обработчики	725
Промежуточные функции	727
Маршрутизация запросов с помощью Compojure	729
Обработка шаблонов.....	743
Enlive: преобразование HTML с применением селекторов.....	745
Попробуем воду	746
Селекторы	748
Итерации и ветвление	750
Объединяем все вместе.....	752
В заключение	756
Глава 17. Развертывание веб-приложений на Clojure.....	758
Веб-архитектура Java и Clojure	758
Упаковка веб-приложения	762
Сборка .war-файлов с помощью Maven	764



Сборка .war-файлов с помощью Leiningen	767
Запуск веб-приложений на локальном компьютере	769
Развертывание веб-приложения	770
Развертывание приложений на Clojure с помощью	
Amazon Elastic Beanstalk	771
За пределами развертывания простых веб-приложений	775
Часть V. РАЗНОЕ	776
Глава 18. Выбор форм определения типов	777
Глава 19. Внедрение Clojure	780
Только факты.....	780
Подчеркните особую продуктивность	782
Подчеркните широту сообщества.....	784
Будьте благоразумны	786
Глава 20. Что дальше?	788
(dissoc Clojure 'JVM).....	788
ClojureCLR.....	788
ClojureScript	789
4Clojure	790
Overtone	790
core.logic	791
Pallet	792
Avout	793
Clojure на платформе Heroku	793
Об авторах	795
Иллюстрация на обложке	796
Предметный указатель	797