

УДК 004.6
ББК 32.973.26
М57

Мински Я., Мадхавapedди А., Хикки Дж.
 М57 Программирование на языке OCaml / пер. с англ. А. Н. Киселева. – М.: ДМК Пресс, 2017. – 536 с.: ил.

ISBN 978-5-97060-561-5

Эта книга введет вас в мир OCaml, надежный язык программирования, обладающий большой выразительностью, безопасностью и быстродействием. Пройдя через множество примеров, вы быстро поймете, что OCaml – это превосходный инструмент, позволяющий писать быстрый, компактный и надежный системный код.

Вы познакомитесь с основными понятиями языка, узнаете о приемах и инструментах, помогающих превратить OCaml в эффективное средство разработки практических приложений. В конце книги вы сможете углубиться в изучение тонких особенностей инструментов компилятора и среды выполнения OCaml.

УДК 004.6
 ББК 32.973.26

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 978-1-449-32391-2 (анг.)

© 2014 Yaron Minsky, Anil Madhavapeddy,
 Jason Hickey

ISBN 978-5-97060-561-5 (рус.)

© Оформление, перевод, ДМК Пресс

Содержание

Вступление	14
Часть I. Основы языка	22
Глава 1. Введение	23
ОSam1 как калькулятор.....	23
Функции и автоматический вывод типов.....	25
Автоматический вывод типов.....	27
Автоматический вывод обобщенных типов.....	28
Кортежи, списки, необязательные значения и сопоставление с образцом.....	30
Кортежи.....	30
Списки.....	31
Необязательные значения.....	38
Записи и варианты.....	40
Императивное программирование.....	42
Массивы.....	42
Изменяемые поля записей.....	43
Ссылки.....	45
Циклы for и while.....	46
Законченная программа.....	48
Компиляция и запуск.....	48
Что дальше.....	49
Глава 2. Переменные и функции	50
Переменные.....	50
Сопоставление с образцом и let.....	53
Функции.....	54
Анонимные функции.....	55
Функции нескольких аргументов.....	57
Рекурсивные функции.....	59
Префиксные и инфиксные операторы.....	60
Объявление функций с помощью ключевого слова function.....	65
Аргументы с метками.....	66
Необязательные аргументы.....	69
Глава 3. Списки и образцы	77
Основы списков.....	77
Использование сопоставления с образцом для извлечения данных из списка.....	78
Ограничения (и благословения) сопоставления с образцом.....	80
Производительность.....	81
Определение ошибок.....	83

Эффективное использование модуля List	84
Другие полезные функции из модуля List	88
Хвостовая рекурсия	91
Компактность и скорость сопоставления с образцом	93
Глава 4. Файлы, модули программы	98
Программы в единственном файле	98
Программы и модули из нескольких файлов	101
Сигнатуры и абстрактные типы	103
Конкретные типы в сигнатурах	106
Вложенные модули	107
Открытие модулей	109
Подключение модулей	111
Типичные ошибки при работе с модулями	113
Несовпадение типов	113
Отсутствие определений	114
Несоответствие определений типов	114
Циклические зависимости	115
Проектирование с применением модулей	117
Старайтесь не экспортировать конкретные типы	117
Продумывайте синтаксис вызовов	117
Создавайте однородные интерфейсы	118
Определяйте интерфейсы до реализации	119
Глава 5. Записи	120
Сопоставление с образцом и полнота	122
Уплотнение полей	124
Повторное использование имен полей	125
Функциональные обновления	129
Изменяемые поля	131
Поля первого порядка	132
Глава 6. Варианты	137
Универсальные образцы и рефакторинг	139
Объединение записей и вариантов	141
Варианты и рекурсивные структуры данных	145
Полиморфные варианты	149
Пример: и снова о цветных терминалах	151
Когда следует использовать полиморфные варианты	157
Глава 7. Обработка ошибок	159
Типы возвращаемых значений с признаком ошибки	159
Кодирование ошибок в результате	160
Error и Or_error	161

Функция bind и другие идиомы обработки ошибок	163
Исключения	165
Вспомогательные функции для возбуждения исключений	167
Обработчики исключений	169
Восстановление работоспособности после исключений	169
Перехват определенных исключений	170
Трассировка стека	172
От исключений к типам с информацией об ошибках и обратно	174
Выбор стратегии обработки ошибок	175
Глава 8. Императивное программирование	177
Пример: императивные словари	177
Элементарные изменяемые данные	182
Данные в формах, подобных массивам	182
Изменяемые поля записей и объектов и ссылочные ячейки	183
Внешние функции	184
Циклы for и while	184
Пример: двусвязные списки	186
Изменение списка	188
Итеративные функции	189
Отложенные вычисления и другие благоприятные эффекты	190
Мемоизация и динамическое программирование	192
Ввод и вывод	200
Терминальный ввод/вывод	200
Форматированный вывод с помощью printf	202
Файловый ввод/вывод	204
Порядок вычислений	207
Побочные эффекты и слабый полиморфизм	209
Ограничение значений	210
Частичное применение и ограничение значения	211
Ослабление ограничения значений	212
В заключение	215
Глава 9. Функторы	216
Простейший пример	216
Более практичный пример: вычисления с применением интервалов	218
Создание абстрактных функторов	222
Совместно используемые ограничения	223
Деструктивная подстановка	225
Использование нескольких интерфейсов	227
Расширение модулей	231
Глава 10. Модули первого порядка	235
Приемы работы с модулями первого порядка	235

Пример: фреймворк обработки запросов	241
Реализация обработчика запросов	242
Диспетчеризация запросов по нескольким обработчикам	244
Загрузка и выгрузка обработчиков запросов.....	248
Жизнь без модулей первого порядка	252
Глава 11. Объекты	253
Объекты OCaml	253
Полиморфизм объектов.....	255
Неизменяемые объекты	257
Когда следует использовать объекты.....	258
Подтипизация	259
Подтипизация в ширину	259
Подтипизация в глубину	260
Вариантность	261
Сужение	265
Подтипизация и рядный полиморфизм	267
Глава 12. Классы	269
Классы в OCaml	269
Параметры класса и полиморфизм.....	270
Типы объектов и интерфейсы.....	272
Функциональные итераторы.....	274
Наследование	276
Типы классов	277
Открытая рекурсия	278
Скрытые методы.....	280
Бинарные методы.....	281
Виртуальные классы и методы.....	285
Создание простых фигур	285
Инициализаторы.....	288
Множественное наследование.....	288
Как выполняется разрешение имен	289
Примеси	290
Отображение анимированных фигур.....	293
Часть II. Инструменты и технологии	295
Глава 13. Отображения и хэш-таблицы	296
Отображения	297
Создание отображений с компараторами	298
Деревья.....	301
Полиморфные компараторы.....	302
Множества	304

Соответствие интерфейсу Comparable.S	304
Хэш-таблицы	307
Соответствие интерфейсу Hashable.S	310
Выбор между отображениями и хэш-таблицами.....	311
Глава 14. Анализ командной строки.....	315
Простейший анализ командной строки	315
Анонимные аргументы	316
Определение простых команд.....	317
Выполнение простых команд	317
Типы аргументов	319
Определение собственных типов аргументов	320
Необязательные аргументы и аргументы по умолчанию.....	321
Последовательности аргументов	324
Добавление поддержки передачи именованных флагов в командной строке.....	325
Группировка подкоманд	327
Расширенное управление парсингом.....	329
Типы в основе Command.Spec	330
Объединение фрагментов спецификаций	331
Интерактивный запрос ввода.....	333
Добавление аргументов с метками в функции обратного вызова.....	335
Автодополнение командной строки средствами Bash	336
Создание фрагментов автодополнения	336
Установка фрагмента автодополнения	337
Альтернативные парсеры командной строки.....	338
Глава 15. Обработка данных JSON	339
Основы JSON	339
Парсинг данных в формате JSON с помощью Yojson	340
Выборка значений из структур JSON	343
Конструирование значений JSON	346
Использование нестандартных расширений JSON	348
Автоматическое отображение JSON в типы OCaml.....	350
Основы ATD.....	350
Аннотации ATD	351
Компиляция спецификаций ATD в код на OCaml.....	352
Пример: запрос информации об организации в GitHub	353
Глава 16. Парсинг с помощью OCamllex и Menhir	357
Лексический анализ и парсинг	358
Определение парсера.....	360
Описание грамматики.....	360
Парсинг последовательностей	362
Определение лексического анализатора.....	364

Вступление.....	364
Регулярные выражения.....	365
Лексические правила	366
Рекурсивные правила	367
Объединяем все вместе	368

Глава 17. Сериализация данных с применением

s-выражений	371
Основы использования.....	372
Преобразование типов OCaml в s-выражения	374
Формат Sexpr	376
Сохранение инвариантов	377
Вывод информативных сообщений об ошибках.....	380
Директивы sexpr-преобразований	382
sexpr_opaque.....	383
sexpr_list	384
sexpr_option	385
Определение значений по умолчанию.....	385

Глава 18. Конкурентное программирование

с помощью Async	388
Основы Async	389
Ivar и uproc	393
Примеры: эхо-сервер.....	395
Усовершенствование эхо-сервера	399
Пример: поиск определений с помощью DuckDuckGo.....	401
Обработка URI	402
Парсинг строк JSON.....	402
Выполнение запроса HTTP	403
Обработка исключений.....	406
Мониторы.....	408
Пример: обработка исключений при работе с DuckDuckGo.....	410
Тайм-ауты, отмена и выбор.....	413
Работа с системными потоками	416
Защищенность данных в потоках и блокировки	419

Часть III. Система времени выполнения..... 421

Глава 19. Интерфейс внешних функций..... 422

Пример: интерфейс к терминалу	423
Простые скалярные типы языка C.....	427
Указатели и массивы.....	429
Выделение памяти для указателей.....	430
Использование представлений для отображения составных значений.....	431

Структуры и объединения	432
Определение структуры	432
Добавление полей в структуры	433
Незавершенные определения структур	433
Определение массивов	437
Передача функций в код на С	438
Пример: быстрая сортировка в командной строке	439
Дополнительная информация о взаимодействии с кодом на С	441
Организация структур в памяти	442
Глава 20. Представление значений в памяти	444
Блоки и значения OCaml	445
Различение целых чисел и указателей во время выполнения	445
Блоки и значения	447
Целые числа, символы и другие простые типы	448
Кортежи, записи и массивы	448
Вещественные числа и массивы	449
Варианты и списки	450
Полиморфные варианты	452
Строковые значения	453
Нестандартные блоки памяти	454
Управление внешней памятью средствами Bigarray	454
Глава 21. Сборка мусора	456
Алгоритм сборки мусора	456
Сборка мусора с разделением на поколения	457
Быстрая вспомогательная куча	457
Выделение памяти во вспомогательной куче	458
Основная куча долгоживущих блоков	459
Выделение памяти в основной куче	460
Стратегии распределения памяти	461
Маркировка и сканирование кучи	462
Компактификация кучи	463
Указатели между поколениями	464
Подключение функций-финализаторов к значениям	467
Глава 22. Компиляторы: парсинг и контроль типов	470
Обзор инструментов компилятора	470
Парсинг исходного кода	472
Синтаксические ошибки	473
Автоматическое оформление отступов в исходном коде	473
Автоматическое создание документации на основе интерфейсов	475
Препроцессинг исходного кода	477
Использование Camlp4 в интерактивной оболочке	479

Запуск Camlp4 из командной строки	480
Препроцессинг сигнатур модулей	482
Дополнительные источники информации о Camlp4	483
Статическая проверка типов	483
Демонстрация типов, выводимых компилятором	484
Вывод типов	486
Модули и отдельная компиляция	491
Упаковка модулей вместе	493
Сокращение путей к модулям в сообщениях об ошибках	495
Типизированное синтаксическое дерево	496
Использование ocp-index для поддержки автодополнения	496
Непосредственное исследование типизированного синтаксического дерева	497
Глава 23. Компиляторы: байт-код и машинный код	501
Нетипизированная lambda-форма	501
Оптимизация сопоставлений с образцом	501
Оценка производительности сопоставления с образцом	504
Переносимый байт-код	506
Компиляция и компоновка байт-кода	507
Выполнение байт-кода	508
Встраивание байт-кода OCaml в программы на C	509
Компиляция быстрого машинного кода	511
Исследование ассемблерного кода	511
Отладка двоичных выполняемых файлов	515
Профилирование машинного кода	519
Встраивание машинного кода в программы на C	521
Сводка по расширениям имен файлов	522
Алфавитный указатель	523