

**УДК 004.438Common Lisp**  
**ББК 32.973.22**  
**С14**

С14 Сайбель П.  
Практическое использование Common Lisp / пер. с англ. А.Я. Отта. – М.: ДМК Пресс, 2017. – 488 с.: ил.

**ISBN 978-5-97060-538-7**

В отличие от основной массы литературы про Lisp, эта книга не просто рассказывает о ряде возможностей языка, предоставляя читателю самостоятельно осваивать их на практике. Здесь будут описаны все функции языка, которые понадобятся вам для написания реальных программ. Более трети книги посвящено разработке нетривиальных программ – статистического фильгра для спама, библиотеки для разбора двоичных файлов и сервера для трансляции музыки в формате MP3 через сеть, включающего в себя базу данных (MP3-файлов) и веб-интерфейс.

Издание предназначено для программистов различной квалификации, как уже использующих Lisp в своей работе, так и только знакомящихся с этим языком.

**УДК 004.438Common Lisp**  
**ББК 32.973.22**

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 1-59059-239-5 (анг.)  
ISBN 978-5-97060-538-7 (рус.)

© Peter Seibel  
© Оформление, перевод, ДМК Пресс, 2017

# Оглавление

От коллектива переводчиков . . . . .	13
<b>1. Введение: почему Lisp? . . . . .</b>	<b>14</b>
1.1. Почему Lisp? . . . . .	15
1.2. С чего всё началось . . . . .	18
1.3. Для кого эта книга? . . . . .	20
<b>2. Намылить, смыть, повторить: знакомство с REPL . . . . .</b>	<b>22</b>
2.1. Выбор реализации Lisp . . . . .	22
2.2. Введение в Lisp in a Box . . . . .	24
2.3. Освободите свой разум: интерактивное программирование . . . . .	25
2.4. Эксперименты в REPL . . . . .	26
2.5. «Hello, world» в стиле Lisp . . . . .	26
2.6. Сохранение вашей работы . . . . .	28
<b>3. Практикум: простая база данных . . . . .</b>	<b>33</b>
3.1. CD и записи . . . . .	34
3.2. Заполнение CD . . . . .	35
3.3. Просмотр содержимого базы данных . . . . .	36
3.4. Улучшение взаимодействия с пользователем . . . . .	37
3.5. Сохранение и загрузка базы данных . . . . .	40
3.6. Выполнение запросов к базе данных . . . . .	41
3.7. Обновление существующих записей — повторное использование where . . . . .	45
3.8. Избавление от дублирующего кода и большой выигрыш . . . . .	46
3.9. Об упаковке . . . . .	51
<b>4. Синтаксис и семантика . . . . .</b>	<b>52</b>
4.1. Зачем столько скобок? . . . . .	52
4.2. Вскрытие чёрного ящика . . . . .	53
4.3. S-выражения . . . . .	54
4.4. S-выражения как формы Lisp . . . . .	57
4.5. Вызовы функций . . . . .	58
4.6. Специальные операторы . . . . .	59
4.7. Макросы . . . . .	60
4.8. Истина, ложь и равенство . . . . .	61

4.9.	Форматирование кода Lisp . . . . .	63
<b>5.</b>	<b>Функции . . . . .</b>	<b>66</b>
5.1.	Определение новых функций . . . . .	66
5.2.	Списки параметров функций . . . . .	68
5.3.	Необязательные параметры . . . . .	68
5.4.	Остаточные (rest) параметры . . . . .	70
5.5.	Именованные параметры . . . . .	71
5.6.	Совместное использование разных типов параметров . . . . .	72
5.7.	Возврат значений из функции . . . . .	74
5.8.	Функции как данные, или функции высшего порядка . . . . .	75
5.9.	Анонимные функции . . . . .	77
<b>6.</b>	<b>Переменные . . . . .</b>	<b>80</b>
6.1.	Основы переменных . . . . .	80
6.2.	Лексические переменные и замыкания . . . . .	83
6.3.	Динамические (специальные) переменные . . . . .	84
6.4.	Константы . . . . .	89
6.5.	Присваивание . . . . .	90
6.6.	Обобщённое присваивание . . . . .	91
6.7.	Другие способы изменения «мест» . . . . .	92
<b>7.</b>	<b>Макросы: стандартные управляющие конструкции . . . . .</b>	<b>94</b>
7.1.	WHEN и UNLESS . . . . .	95
7.2.	COND . . . . .	97
7.3.	AND, OR и NOT . . . . .	97
7.4.	Циклы . . . . .	98
7.5.	DOLIST и DOTIMES . . . . .	99
7.6.	DO . . . . .	100
7.7.	Всемогущий LOOP . . . . .	102
<b>8.</b>	<b>Макросы: создание собственных макросов . . . . .</b>	<b>104</b>
8.1.	История Мака: обычная такая история . . . . .	104
8.2.	Время раскрытия макросов против времени выполнения . . . . .	106
8.3.	DEFMACRO . . . . .	107
8.4.	Пример макроса: do-primes . . . . .	108
8.5.	Макропараметры . . . . .	109
8.6.	Генерация раскрытия . . . . .	110
8.7.	Устранение протечек . . . . .	112
8.8.	Макросы, создающие макросы . . . . .	116
8.9.	Другой классический макрос, создающий макросы: ONCE-ONLY . . . . .	118
8.10.	Не только простые макросы . . . . .	118
<b>9.</b>	<b>Практикум: каркас для unit-тестирования . . . . .</b>	<b>119</b>
9.1.	Два первых подхода . . . . .	120
9.2.	Рефакторинг . . . . .	121

9.3.	Чиним возвращаемое значение . . . . .	122
9.4.	Улучшение отчёта . . . . .	123
9.5.	Выявление абстракций . . . . .	125
9.6.	Иерархия тестов . . . . .	126
9.7.	Подведение итогов . . . . .	127
<b>10.</b>	<b>Числа, знаки и строки . . . . .</b>	<b>129</b>
10.1.	Числа . . . . .	130
10.2.	Запись чисел . . . . .	131
10.3.	Базовые математические операции . . . . .	134
10.4.	Сравнение чисел . . . . .	135
10.5.	Высшая математика . . . . .	136
10.6.	Знаки (characters) . . . . .	136
10.7.	Сравнение знаков . . . . .	137
10.8.	Строки . . . . .	137
10.9.	Сравнение строк . . . . .	138
<b>11.</b>	<b>Коллекции . . . . .</b>	<b>140</b>
11.1.	Векторы . . . . .	140
11.2.	Подтипы векторов . . . . .	142
11.3.	Векторы как последовательности . . . . .	143
11.4.	Функции для работы с элементами последовательностей . . . . .	144
11.5.	Аналогичные функции высшего порядка . . . . .	146
11.6.	Работа с последовательностью целиком . . . . .	147
11.7.	Сортировка и слияние . . . . .	148
11.8.	Работа с частями последовательностей . . . . .	149
11.9.	Предикаты для последовательностей . . . . .	150
11.10.	Функции отображения последовательностей . . . . .	151
11.11.	Хэш-таблицы . . . . .	151
11.12.	Функции для работы с записями в хэш-таблицах . . . . .	153
<b>12.</b>	<b>Они называли его Lisp неспроста: обработка списков . . . . .</b>	<b>154</b>
12.1.	Списков нет . . . . .	154
12.2.	Функциональное программирование и списки . . . . .	157
12.3.	«Разрушающие» операции . . . . .	158
12.4.	Комбинирование утилизации с общими структурами . . . . .	160
12.5.	Функции для работы со списками . . . . .	162
12.6.	Отображение . . . . .	163
12.7.	Другие структуры . . . . .	165
<b>13.</b>	<b>Не только списки: другие применения cons-ячеек . . . . .</b>	<b>166</b>
13.1.	Деревья . . . . .	166
13.2.	Множества . . . . .	168
13.3.	Таблицы поиска: ассоциативные списки и списки свойств . . . . .	170
13.4.	DESTRUCTURING-BIND . . . . .	174

<b>14. Файлы и файловый ввод/вывод . . . . .</b>	<b>176</b>
14.1. Чтение данных из файлов . . . . .	176
14.2. Чтение двоичных данных . . . . .	178
14.3. Блочное чтение . . . . .	178
14.4. Файловый вывод . . . . .	179
14.5. Заккрытие файлов . . . . .	180
14.6. Имена файлов . . . . .	181
14.7. Как имена путей представляют имена файлов . . . . .	182
14.8. Конструирование имён путей . . . . .	184
14.9. Два представления для имён директорий . . . . .	186
14.10. Взаимодействие с файловой системой . . . . .	187
14.11. Другие операции ввода/вывода . . . . .	189
<b>15. Практика: переносимая библиотека файловых путей . . . . .</b>	<b>191</b>
15.1. API . . . . .	191
15.2. Переменная *FEATURES* и обработка условий при считывании . . . . .	192
15.3. Получение списка файлов в директории . . . . .	193
15.4. Проверка существования файла . . . . .	197
15.5. Проход по дереву каталогов . . . . .	198
<b>16. Переходим к объектам: обобщённые функции . . . . .</b>	<b>200</b>
16.1. Обобщённые функции и классы . . . . .	201
16.2. Обобщённые функции и методы . . . . .	203
16.3. DEFGENERIC . . . . .	204
16.4. DEFMETHOD . . . . .	205
16.5. Комбинирование методов . . . . .	207
16.6. Стандартный комбинатор методов . . . . .	209
16.7. Другие комбинаторы методов . . . . .	210
16.8. Мультиметоды . . . . .	212
16.9. Продолжение следует... . . . . .	214
<b>17. Переходим к объектам: классы . . . . .</b>	<b>215</b>
17.1. DEFCLASS . . . . .	215
17.2. Спецификаторы слотов . . . . .	216
17.3. Инициализация объекта . . . . .	217
17.4. Функции доступа . . . . .	220
17.5. WITH-SLOTS и WITH-ACCESSORS . . . . .	223
17.6. Слоты, выделяемые для классов . . . . .	225
17.7. Слоты и наследование . . . . .	226
17.8. Множественное наследование . . . . .	227
17.9. Правильный объектно-ориентированный дизайн . . . . .	230
<b>18. Несколько рецептов для функции FORMAT . . . . .</b>	<b>231</b>
18.1. Функция FORMAT . . . . .	232
18.2. Директивы FORMAT . . . . .	233
18.3. Основы форматирования . . . . .	235

18.4.	Директивы для знаков и целых чисел . . . . .	235
18.5.	Директивы для чисел с плавающей точкой . . . . .	237
18.6.	Директивы для английского языка . . . . .	238
18.7.	Условное форматирование . . . . .	240
18.8.	Итерация . . . . .	241
18.9.	Тройной прыжок . . . . .	243
18.10.	И многое другое... . . . . .	244
<b>19.</b>	<b>Обработка исключений изнутри: условия и перезапуск . . . . .</b>	<b>245</b>
19.1.	Путь языка Lisp . . . . .	247
19.2.	Условия . . . . .	247
19.3.	Обработчики условий . . . . .	248
19.4.	Перезапуск . . . . .	250
19.5.	Предоставление множественных перезапусков . . . . .	253
19.6.	Другие применения условий . . . . .	254
<b>20.</b>	<b>Специальные операторы . . . . .</b>	<b>257</b>
20.1.	Контроль вычисления . . . . .	257
20.2.	Манипуляции с лексическим окружением . . . . .	258
20.3.	Локальный поток управления . . . . .	261
20.4.	Раскрутка стека . . . . .	264
20.5.	Множественные значения . . . . .	268
20.6.	EVAL-WHEN . . . . .	270
20.7.	Другие специальные операторы . . . . .	273
<b>21.</b>	<b>Программирование по-взрослому: пакеты и символы . . . . .</b>	<b>275</b>
21.1.	Как процедура чтения использует пакеты . . . . .	275
21.2.	Немного про словарь пакетов и символов . . . . .	277
21.3.	Три стандартных пакета . . . . .	278
21.4.	Определение собственных пакетов . . . . .	279
21.5.	Упаковка библиотек для повторного использования . . . . .	282
21.6.	Импорт отдельных имён . . . . .	283
21.7.	Пакетная механика . . . . .	284
21.8.	Пакетные ловушки . . . . .	286
<b>22.</b>	<b>LOOP для мастеров с чёрным поясом . . . . .</b>	<b>289</b>
22.1.	Части LOOP . . . . .	289
22.2.	Управление итерированием . . . . .	290
22.3.	Подсчитывающие циклы (Counting Loops) . . . . .	290
22.4.	Организация циклов по коллекциям и пакетам . . . . .	292
22.5.	Equals-Then-итерирование . . . . .	293
22.6.	Локальные переменные . . . . .	294
22.7.	Деструктурирование переменных . . . . .	294
22.8.	Накопление значения . . . . .	295
22.9.	Безусловное выполнение . . . . .	296
22.10.	Условное выполнение . . . . .	297

22.11. Начальные установки и подытоживание . . . . .	299
22.12. Критерии завершения . . . . .	300
22.13. Сложим все вместе . . . . .	302
<b>23. Практика: спам-фильтр . . . . .</b>	<b>303</b>
23.1. Сердце спам-фильтра . . . . .	303
23.2. Тренируем фильтр . . . . .	307
23.3. Пословная статистика . . . . .	309
23.4. Комбинирование вероятностей . . . . .	311
23.5. Обратная функция chi-квадрат . . . . .	314
23.6. Тренируем фильтр . . . . .	314
23.7. Тестируем фильтр . . . . .	316
23.8. Набор вспомогательных функций . . . . .	318
23.9. Анализ результатов . . . . .	319
23.10. Что далее? . . . . .	321
<b>24. Практика. Разбор двоичных файлов . . . . .</b>	<b>323</b>
24.1. Двоичные файлы . . . . .	323
24.2. Основы двоичного формата . . . . .	324
24.3. Строки в двоичных файлах . . . . .	326
24.4. Составные структуры . . . . .	329
24.5. Проектирование макросов . . . . .	330
24.6. Делаем мечту реальностью . . . . .	331
24.7. Чтение двоичных объектов . . . . .	332
24.8. Запись двоичных объектов . . . . .	335
24.9. Добавление наследования и помеченных (tagged) структур . . . . .	336
24.10. Отслеживание унаследованных слотов . . . . .	338
24.11. Помеченные структуры . . . . .	340
24.12. Примитивные двоичные типы . . . . .	342
24.13. Стек обрабатываемых в данный момент объектов . . . . .	345
<b>25. Практика: разбор ID3 . . . . .</b>	<b>347</b>
25.1. Структура тега ID3v2 . . . . .	348
25.2. Определение пакета . . . . .	349
25.3. Типы целых . . . . .	350
25.4. Типы строк . . . . .	351
25.5. Заголовок тега ID3 . . . . .	355
25.6. Фреймы ID3 . . . . .	356
25.7. Обнаружение заполнителя тега . . . . .	358
25.8. Поддержка нескольких версий ID3 . . . . .	359
25.9. Базовые классы для фреймов разных версий . . . . .	361
25.10. Конкретные классы для фреймов разных версий . . . . .	362
25.11. Какие фреймы на самом деле нужны? . . . . .	363
25.12. Фреймы текстовой информации . . . . .	365
25.13. Фреймы комментариев . . . . .	367
25.14. Извлечение информации из тега ID3 . . . . .	368

<b>26. Практика. Веб-программирование с помощью AllegroServe . . . . .</b>	<b>373</b>
26.1. 30-секундное введение в веб-программирование на стороне сервера . . . . .	373
26.2. AllegroServe . . . . .	376
26.3. Генерация динамического содержимого с помощью AllegroServe . . . . .	379
26.4. Генерация HTML . . . . .	380
26.5. Макросы HTML . . . . .	383
26.6. Параметры запроса . . . . .	384
26.7. Cookies . . . . .	387
26.8. Небольшой каркас приложений . . . . .	390
26.9. Реализация . . . . .	391
<b>27. Практика: База данных для MP3 . . . . .</b>	<b>395</b>
27.1. База данных . . . . .	395
27.2. Определение схемы базы данных . . . . .	398
27.3. Вставка значений . . . . .	400
27.4. Выполнение запросов к базе данных . . . . .	401
27.5. Функции отбора . . . . .	404
27.6. Работа с результатами выполнения запросов . . . . .	406
27.7. Другие операции с базой данных . . . . .	408
<b>28. Практика. Сервер Shoutcast . . . . .</b>	<b>410</b>
28.1. Протокол Shoutcast . . . . .	410
28.2. Источники песен . . . . .	411
28.3. Реализация сервера Shoutcast . . . . .	414
<b>29. Практика. Браузер MP3-файлов . . . . .</b>	<b>420</b>
29.1. Списки песен . . . . .	420
29.2. Списки песен как источники песен . . . . .	422
29.3. Изменение списка песен . . . . .	426
29.4. Типы параметров запроса . . . . .	429
29.5. Шаблонный HTML . . . . .	430
29.6. Страница просмотра . . . . .	432
29.7. Плей-лист . . . . .	435
29.8. Находим плей-лист . . . . .	437
29.9. Запускаем приложение . . . . .	438
<b>30. Практика: библиотека для генерации HTML – интерпретатор . . . . .</b>	<b>439</b>
30.1. Проектирование языка специального назначения . . . . .	439
30.2. Язык FOO . . . . .	441
30.3. Экранирование знаков . . . . .	443
30.4. Вывод отступов . . . . .	445
30.5. Интерфейс HTML-процессора . . . . .	446
30.6. Реализация форматированного вывода . . . . .	447
30.7. Базовое правило вычисления . . . . .	450
30.8. Что дальше? . . . . .	453

---

<b>31. Практика: библиотека для генерации HTML – компилятор . . . . .</b>	<b>454</b>
31.1. Компилятор . . . . .	454
31.2. Специальные операторы FOO . . . . .	459
31.3. Макросы FOO . . . . .	464
31.4. Публичный интерфейс разработчика (API) . . . . .	467
31.5. Завершение работы . . . . .	468
<b>32. Заключение: что дальше? . . . . .</b>	<b>470</b>
32.1. Поиск библиотек Lisp . . . . .	470
32.2. Взаимодействие с другими языками программирования . . . . .	472
32.3. Сделать, чтобы работало; правильно, быстро . . . . .	473
32.4. Поставка приложений . . . . .	480
32.5. Что дальше? . . . . .	484