

**УДК 004.432.42**  
**ББК 32.973.2-018**  
**О49**

**О49 Крис Окасаки**  
 Чисто функциональные структуры данных / Пер. с англ. Г.К.Бронников,  
 – М.: ДМК Пресс, 2016. — 252 с.: ил.  
**ISBN 978-5-97060-233-1**

Большинство книг по структурам данных предполагают использование императивного языка программирования, например, C/C++ или Java. Однако реализации структур данных на таких языках далеко не всегда хорошо переносятся на функциональные языки программирования, такие как Стандартный ML, Haskell или Scheme. В этой книге структуры данных описываются с точки зрения функциональных языков, в ней содержатся примеры и предлагаются подходы к проектированию, которые могут использоваться разработчиками при создании их собственных структур данных. Книга включает в себя как классические структуры данных, к примеру, красно-чёрные деревья и биномиальные очереди, так и некоторые новые структуры данных, созданные специально для функциональных языков. Весь исходный код приводится на Стандартном ML и Haskell, причём большинство программ нетрудно адаптировать для других функциональных языков программирования.

Это издание представляет собой справочное руководство для профессиональных программистов, работающих с функциональными языками, и может также использоваться в качестве учебника для самостоятельного изучения.

УДК 004.432.42  
 ББК 32.973.2-018

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 всё равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несёт ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-0-5216635-0-2 (англ.)

© Cambridge University Press, 1998

© Г.К. Бронников, перевод, 2015

ISBN 978-5-97060-233-1 (рус.)

© Оформление, ДМК Пресс, 2016

# Оглавление

От редактора перевода	8
Предисловие	9
1. Введение	11
1.1. Функциональные и императивные структуры данных . . . . .	11
1.2. Энергичное и ленивое вычисление . . . . .	12
1.3. Терминология . . . . .	13
1.4. Наш подход . . . . .	14
1.5. Обзор книги . . . . .	15
2. Устойчивость	17
2.1. Списки . . . . .	17
2.2. Двоичные деревья поиска . . . . .	21
2.3. Примечания . . . . .	26
3. Знакомые структуры данных в функциональном окружении	27
3.1. Левоориентированные кучи . . . . .	27
3.2. Биномиальные кучи . . . . .	31
3.3. Красно-чёрные деревья . . . . .	35
3.4. Примечания . . . . .	40
4. Ленивое вычисление	41
4.1. \$-запись . . . . .	41
4.2. Потоки . . . . .	44
4.3. Примечания . . . . .	46
5. Основы амортизации	49
5.1. Методы амортизированного анализа . . . . .	49
5.2. Очереди . . . . .	52
5.3. Биномиальные кучи . . . . .	55
5.4. Расширяющиеся кучи . . . . .	57
5.5. Парные кучи . . . . .	64
5.6. Плохие новости . . . . .	66
5.7. Примечания . . . . .	67

6.	Амортизация и устойчивость при ленивом вычислении	69
6.1.	Трассировка вычисления и логическое время	69
6.2.	Сочетание амортизации и устойчивости	71
6.2.1.	Роль ленивого вычисления	71
6.2.2.	Общая методика анализа ленивых структур данных	72
6.3.	Метод банкира	74
6.3.1.	Обоснование метода банкира	75
6.3.2.	Пример: очереди	77
6.3.3.	Наследование долга	81
6.4.	Метод физика	82
6.4.1.	Пример: биномиальные кучи	84
6.4.2.	Пример: очереди	86
6.4.3.	Сортировка слиянием снизу вверх с совместным использованием	88
6.5.	Ленивые парные кучи	94
6.6.	Примечания	95
7.	Избавление от амортизации	98
7.1.	Расписания	99
7.2.	Очереди реального времени	101
7.3.	Биномиальные кучи	105
7.4.	Сортировка снизу вверх с расписанием	110
7.5.	Примечания	114
8.	Ленивая перестройка	116
8.1.	Порционная перестройка	116
8.2.	Глобальная перестройка	118
8.2.1.	Пример: очереди реального времени по Худу–Мелвиллу	119
8.3.	Ленивая перестройка	122
8.4.	Двусторонние очереди	124
8.4.1.	Деки с ограниченным выходом	125
8.4.2.	Деки по методу банкира	126
8.4.3.	Деки реального времени	129
8.5.	Примечания	130
9.	Числовые представления	133
9.1.	Позиционные системы счисления	134
9.2.	Двоичные числа	134
9.2.1.	Двоичные списки с произвольным доступом	138
9.2.2.	Безнулевые представления	142

9.2.3. Ленивые представления . . . . .	144
9.2.4. Сегментированные представления . . . . .	147
9.3. Скошенные двоичные числа . . . . .	150
9.3.1. Скошенные двоичные списки с произвольным доступом	152
9.3.2. Скошенные биномиальные кучи . . . . .	155
9.4. Троичные и четверичные числа . . . . .	159
9.5. Примечания . . . . .	161
10. Развёртка структур данных . . . . .	162
10.1. Структурная декомпозиция . . . . .	163
10.1.1. Гетерогенная рекурсия и Стандартный ML . . . . .	164
10.1.2. Снова двоичные списки с произвольным доступом . .	165
10.1.3. Развёрнутые очереди . . . . .	169
10.2. Структурная абстракция . . . . .	173
10.2.1. Списки с эффективной конкатенацией . . . . .	175
10.2.2. Кучи с эффективным слиянием . . . . .	181
10.3. Развёртка до составных типов . . . . .	186
10.3.1. Префиксные деревья . . . . .	186
10.3.2. Обобщённые префиксные деревья . . . . .	190
10.4. Примечания . . . . .	193
11. Неявное рекурсивное замедление . . . . .	195
11.1. Очереди и деки . . . . .	195
11.2. Двусторонние очереди с конкатенацией . . . . .	200
11.3. Примечания . . . . .	209
A. Код на языке Haskell . . . . .	210
A.1. Очереди . . . . .	210
A.2. Двусторонние очереди . . . . .	215
A.3. Списки с конкатенацией . . . . .	216
A.4. Двусторонние очереди с конкатенацией . . . . .	217
A.5. Списки с произвольным доступом . . . . .	221
A.6. Кучи . . . . .	225
A.7. Сортируемые коллекции . . . . .	232
A.8. Множества . . . . .	232
A.9. Конечные отображения . . . . .	234
Литература . . . . .	236
Предметный указатель . . . . .	247