

Оглавление

Об авторе	9
О научном редакторе	9
Благодарности	10
Введение	11
Зачем изучать функциональное программирование	11
Почему вам нужна эта книга	11
Часть I. Первые шаги	13
Глава 1. Настоящий функциональный язык	14
Почему именно Haskell?	14
История Haskell	19
Ваша рабочая среда	19
Первые шаги в использовании GHCi	24
Магазин, продающий машины времени	26
Выводы	27
Глава 2. Объявление модели данных	28
Работа с символами, числами и списками	28
Создание нового проекта	36
Определение простых функций	40
Работа с типами данных	46
Сопоставление с образцом	49
Записи	58
Выводы	63
Глава 3. Многократное использование кода с помощью списков	64
Параметрический полиморфизм	65
Функции в качестве аргументов	68
Подробнее о модулях	76
Углубленное изучение списков	80
Формирователи списков	91

Haskell-оригами	95
Выводы	98
Глава 4. Контейнеры и классы типов	100
Использование пакетов	100
Контейнеры: проекции, множества, деревья и графы	108
Специальный полиморфизм: классы типов	120
Двоичные деревья для минимальной цены	129
Классы типов, связанные с контейнерами	134
Выводы	138
Глава 5. Ленив и бесконечные структуры	139
Бесконечное количество машин времени	139
Модель ленивых вычислений	144
Профилирование с помощью GHC	152
Строгие аннотации	157
Выводы	160
Часть II. Анализ данных	161
Глава 6. Ознакомление с клиентами с помощью монад	162
Анализ данных	163
Исследование монад	174
Различные виды состояний	186
Выводы	193
Глава 7. Другие рекомендуемые монады	194
Возвращение более одного значения	194
Неудачи и альтернативы	197
Вывод ассоциативных правил	200
Задачи поиска	207
Монада Logic	208
Возрождение монад и списков	210
Выводы	223
Глава 8. Использование нескольких ядер	224
Параллелизм, одновременность, распределенность	224
Монада Par	226
Программная транзакционная память	232
Облачные решения для Haskell	241
Выводы	248

Часть III. Использование ресурсов	249
Глава 9. Работа с файлами: ввод-вывод и библиотека conduit	250
Базовые ввод и вывод	250
Случайность	254
Работа с файлами	257
Обработка ошибок	261
Потоковый ввод-вывод данных с помощью библиотеки conduit	270
За пределами текстовых файлов	277
Выводы	281
Глава 10. Создание и синтаксический разбор текста	282
Пять текстовых типов данных	282
Создание текста со скоростью ветра	287
Синтаксический разбор с использованием пакета attoparsec	290
Представление новых классов типов	296
Не нужно излишеств: используйте формат JSON	304
Выводы	310
Глава 11. Безопасный доступ к базам данных	311
Средства доступа к базам данных	311
Соединение	314
Схемы и переносы	316
Запросы	323
Вставка, обновление и удаление	330
Выводы	333
Глава 12. Веб-приложения	334
Веб-экосистема Haskell	334
Структура RESTful	338
Разработка внутреннего интерфейса в среде Scotty	339
Разработка внешнего интерфейса с помощью компилятора Fay	349
Выводы	354
Часть IV. Предметно-ориентированные языки	355
Глава 13. Строгие типы для описания предложений	356
Предметно-ориентированные языки	356
Безопасность языка выражений	362
Зависимая типизация	365
Программирование на уровне типов в Haskell	370
Функциональные зависимости	373

Семейства типов	379
Продвижение типа и одноэлементные типы	385
Выводы	394
Глава 14. Интерпретация предложений с помощью атрибутов	395
Варианты интерпретации и атрибутивные грамматики	395
Наша первая атрибутивная грамматика	398
Интеграция UUAGC-кода в пакет	401
Интерпретация выражений	404
Варианты интерпретации предложений	409
Оригами-программирование для любого типа данных	415
Выводы	418
Часть V. Инжиниринг магазина	419
Глава 15. Документирование, тестирование и проверка	420
Документирование двоичных деревьев с помощью утилиты Haddock	421
Блочное тестирование с помощью утилиты HUnit	424
Рандомизированное тестирование с помощью библиотеки QuickCheck	430
Формальная верификация с использованием языка Idris	434
Выводы	438
Глава 16. Создание архитектуры приложения	439
Паттерны проектирования и функциональное программирование . .	439
Рекомендации среднего порядка	441
Утилиты	443
Проекты	446
Выводы	458
Приложение А. Дальнейшие перспективы	459
Haskell-ресурсы	460
Приложение Б. Путешествие во времени с Haskell	462