

УДК 004.432.42 Haskell
ББК 32.973.28-018.1
К93

К93 Уилл Курт
Программируй на Haskell / пер. с англ. Я. О. Касюлевича,
А. А. Романовского и С. Д. Степаненко; под ред. В. Н. Брагилевского.
– М.: ДМК Пресс, 2019. — 648 с.: ил.
ISBN 978-5-97060-694-0

Языки программирования зачастую отличаются лишь отдельными элементами — несколько ключевых слов, библиотек или платформенных решений. Haskell формирует абсолютно новую точку зрения. По мнению пионера программного обеспечения Алана Кэя, смена перспективы может дать 80 баллов IQ, и разработчики на Haskell соглашаются с исключительными преимуществами мышления в стиле Haskell: функционального мышления с ориентацией на типобезопасность, математическую определённую и многое другое. В этой практической книге вы будете учиться именно этому.

«Программируй на Haskell» проведёт вас через короткие уроки, примеры и упражнения, разработанные так, чтобы вы смогли прочувствовать Haskell. В ней вы найдёте кристально ясные иллюстрации и легко сможете практиковаться под её руководством. Вы будете писать и тестировать дюжины интересных программ, а также погрузитесь в различные модули и библиотеки. Вы получите новую перспективу в программировании и возможность использовать Haskell в реальном мире (80 баллов IQ не гарантируются).

Написано для читателей, который уже знают хотя бы один язык программирования.

УДК 004.432.42 Haskell
ББ 32.973.28-018.1

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 9781617293764 (англ.)
ISBN 978-5-97060-694-0 (рус.)

© 2018 Will Kurt
© Перевод на русский язык, оформление,
ДМК Пресс, 2019

Оглавление

Предисловие	13
Благодарности	15
Об этой книге	17
Об авторе	21
Урок 1. Начало работы с Haskell	22
1.1. Добро пожаловать в мир Haskell	22
1.2. Компилятор GHC языка Haskell	23
1.3. Взаимодействие с Haskell — GHCi	25
1.4. Написание кода на Haskell и работа с ним	28
Модуль 1. Основания функционального программирования	34
Урок 2. Функции и функциональное программирование	36
2.1. Функции	37
2.2. Функциональное программирование	38
2.3. Функциональное программирование на практике	39
Урок 3. Лямбда-функции и лексическая область видимости	46
3.1. Лямбда-функции	47
3.2. Пишем свой аналог блока where	48
3.3. От лямбда-функций к let: изменяемые переменные!	51
3.4. Лямбда-функции и области видимости на практике	53
Урок 4. Функции как значения первого класса	57
4.1. Функции как аргументы	58
4.2. Возвращаем функции	63
Урок 5. Замыкания и частичное применение функций	67

5.1. Замыкания — создание функций функциями	68
5.2. Пример: генерация URL для API	69
5.3. Собираем всё вместе	75
Урок 6. Списки	78
6.1. Анатомия списков	79
6.2. Списки и ленивые вычисления	82
6.3. Основные функции на списках	84
Урок 7. Правила рекурсии и сопоставление с образцом	90
7.1. Рекурсия	91
7.2. Правила рекурсии	92
7.3. Ваша первая рекурсивная функция: наибольший общий делитель	94
Урок 8. Написание рекурсивных функций	99
8.1. Обзор: правила рекурсии	100
8.2. Рекурсия на списках	100
8.3. Патологическая рекурсия: функция Аккермана и гипотеза Коллатца	103
Урок 9. Функции высшего порядка	109
9.1. Использование map	110
9.2. Обобщение вычислений с помощью map	111
9.3. Фильтрация списка	113
9.4. Свёртка списка	114
Урок 10. Итоговый проект: функциональное объектно-ориентированное программирование и роботы!	119
10.1. Объект с одним свойством: кружка кофе	120
10.2. Более сложные объекты: создаём боевых роботов!	124
10.3. Почему важно программировать без состояний	128
10.4. Типы — объекты и многое другое!	130
Модуль 2. Введение в типы	132
Урок 11. Основы системы типов	134
11.1. Типы в Haskell	135
11.2. Типы функций	138
11.3. Типовые переменные	143

Урок 12. Создание пользовательских типов	148
12.1. Использование синонимов типов	149
12.2. Создание новых типов	151
12.3. Использование синтаксиса записей	156
Урок 13. Классы типов	161
13.1. Дальнейшее исследование типов	162
13.2. Классы типов	163
13.3. Преимущества классов типов	164
13.4. Определение класса типов	164
13.5. Основные классы типов	166
13.6. Порождение экземпляров классов типов	169
Урок 14. Использование классов типов	172
14.1. Тип, требующий классы	173
14.2. Реализация Show	173
14.3. Классы типов и полиморфизм	174
14.4. Реализации методов по умолчанию и минимально полные определения	176
14.5. Реализация Ord	178
14.6. Порождать или нет?	180
14.7. Классы типов для более сложных типов	182
14.8. Схема классов типов	184
Урок 15. Итоговый проект: секретные сообщения!	186
15.1. Шифры для начинающих: ROT13	186
15.2. XOR: магия криптографии!	194
15.3. Представление значений как битов	196
15.4. Одноразовый блокнот	199
15.5. Класс Cipher	201
Модуль 3. Программирование в типах	205
Урок 16. Создание типов с помощью «И» и «ИЛИ»	207
16.1. Типы-произведения — объявление типов с помощью «И»	208
16.2. Типы-суммы — объявление типов с помощью «ИЛИ»	213
16.3. Собираем книжный магазин	216
Урок 17. Проектирование композицией: полугруппы и моноиды	220
17.1. Введение в композицию: комбинирование функций	221
17.2. Комбинирование схожих типов: полугруппы	222

17.3. Композиция с нейтральным элементом: моноиды	226
17.4. Комбинирование элементов моноида функцией mconcat . .	228
Урок 18. Параметризованные типы	235
18.1. Типы, которые принимают аргументы	236
18.2. Типы с более чем одним параметром	241
Урок 19. Тип Maybe: работа с отсутствующими значениями	248
19.1. Maybe: возможность отсутствия значения как тип	249
19.2. Проблема с null	251
19.3. Вычисления с Maybe	253
19.4. Назад в лабораторию! Снова вычисления с Maybe	255
Урок 20. Итоговый проект: временные ряды	260
20.1. Данные и тип для их представления	261
20.2. Сшивание временных рядов	265
20.3. Вычисления на временных рядах	270
20.4. Преобразование временных рядов	273
20.5. Скользящее среднее	275
Модуль 4. Ввод и вывод в Haskell	279
Урок 21. «Привет, мир!» — введение в ввод-вывод	282
21.1. Типы IO: работаем с «грязным» миром	283
21.2. До-нотация	288
21.3. Пример: вычисление стоимости пиццы	290
Урок 22. Командная строка и ленивый ввод-вывод	295
22.1. Энергичное взаимодействие с командной строкой	296
22.2. Взаимодействие с ленивым вводом-выводом	301
Урок 23. Работа с типом Text и Юникодом	307
23.1. Тип Text	308
23.2. Использование Data.Text	309
23.3. Тип Text и Юникод	315
23.4. Ввод-вывод для Text	317
Урок 24. Работа с файлами	319
24.1. Открытие и закрытие файлов	320
24.2. Простые средства ввода-вывода	323
24.3. Проблемы ленивого ввода-вывода	325

24.4. Строгий ввод-вывод	328
Урок 25. Работа с двоичными данными	331
25.1. Обработка двоичных данных с помощью ByteString	332
25.2. Добавление помех на изображение JPEG	334
25.3. ByteString, Char8 и Юникод	343
Урок 26. Итоговый проект: обработка двоичных файлов и книжных данных	346
26.1. Работа с книжными данными	348
26.2. Работа с MARC-записями	351
26.3. Собираем всё вместе	362
Модуль 5. Работа с типами в контексте	365
Урок 27. Класс типов Functor	369
27.1. Пример: вычисление с Maybe	370
27.2. Класс типов Functor и вызов функций в контексте	372
27.3. Функторы повсюду!	374
Урок 28. Приступаем к аппликативным функторам: функции в контексте	382
28.1. Расчёт расстояния между городами	383
28.2. Операция <*> и частичное применение в контексте	387
28.3. Использование <*> для данных в контексте	393
Урок 29. Списки как контекст: углубляемся в аппликативные вычисления	397
29.1. Представляем класс типов Applicative	398
29.2. Контейнеры и контексты	401
29.3. Список как контекст	403
Урок 30. Введение в класс типов Monad	412
30.1. Ограничения Applicative и Functor	413
30.2. Операция (>=>)	419
30.3. Класс типов Monad	421
Урок 31. Облегчение работы с монадами с помощью do-нотации	427
31.1. Возвращаемся к do-нотации	428
31.2. Использование кода в разных контекстах и do-нотация	431
31.3. Контекст списка — обработка списка кандидатов	436

Урок 32. Монада списка и генераторы списков	442
32.1. Построение списков при помощи монады	443
32.2. Генераторы списков	447
32.3. Монады: больше, чем просто списки	449
Урок 33. Итоговый проект: SQL-подобные запросы в Haskell	451
33.1. Начало работы	452
33.2. Простые запросы на списках: select и where	455
33.3. Соединение типов данных Course и Teacher	457
33.4. Построение интерфейса HINQ и тестовые запросы	459
33.5. Определение типа HINQ для запросов	461
33.6. Выполнение HINQ-запросов	462
Модуль 6. Организация кода и сборка проектов	468
Урок 34. Организация кода на Haskell с помощью модулей	469
34.1. Что случится, если использовать имя из Prelude?	470
34.2. Сборка многофайловой программы с помощью модулей . .	473
Урок 35. Сборка проектов при помощи stack	480
35.1. Создание нового проекта stack	481
35.2. Разбор структуры проекта	482
35.3. Написание кода	485
35.4. Сборка и запуск вашего проекта	487
Урок 36. Тестирование свойств с помощью QuickCheck	490
36.1. Создание нового проекта	491
36.2. Разные виды тестирования	492
36.3. Тестирование свойств с помощью QuickCheck	497
Урок 37. Итоговый проект: библиотека для простых чисел	504
37.1. Создание нового проекта	505
37.2. Изменение файлов, созданных по умолчанию	506
37.3. Реализация основных библиотечных функций	507
37.4. Написание тестов для кода	511
37.5. Написание кода факторизации чисел	515
Модуль 7. Применение Haskell на практике	519
Урок 38. Ошибки в Haskell и тип Either	521

38.1. Функция head, частичные функции и ошибки	522
38.2. Обработка частичных функций с помощью Maybe	526
38.3. Первая встреча с Either	528
Урок 39. Создание HTTP-запросов в Haskell	535
39.1. Первоначальная настройка проекта	536
39.2. Использование модуля HTTP.Simple	539
39.3. Создание HTTP-запроса	542
39.4. Собираем всё вместе	544
Урок 40. Работа с данными JSON с использованием Aeson	546
40.1. Первоначальная настройка	548
40.2. Использование библиотеки Aeson	549
40.3. Экземпляры FromJSON и ToJSON для своих типов	551
40.4. Чтение данных, полученных от NOAA	559
Урок 41. Использование баз данных в Haskell	563
41.1. Первоначальная настройка проекта	564
41.2. Использование SQLite и настройка базы данных	565
41.3. Вставка данных: пользователи и данные об аренде	569
41.4. Чтение данных из БД и класс типов FromRow	571
41.5. Модификация существующих данных	575
41.6. Удаление данных из БД	578
41.7. Собираем всё вместе	578
Урок 42. Эффективные массивы с изменением состояния в Haskell	583
42.1. Тип UArray и эффективные массивы	585
42.2. Изменение состояния с помощью STUArray	592
42.3. Извлечение значений из контекста ST	595
42.4. Реализация сортировки методом пузырька	597
Послесловие	601
Примерные решения задач	607
Предметный указатель	631