Посвящается моим родителям, Хильдегард и Мартину.
Люблю вас всегда!

Оглавление

Содержание

Предисловие

Haskell – язык, который при взгляде извне может показаться окутанным тайной, изобилующим научной терминологией и непонятными концепциями. Кое-что из этого правда, но я абсолютно уверен, что решить задачу на этом языке можно, не обладая степенью PhD по математике. Все дело в правильной подводке. На Haskell можно очень быстро начать писать реальные программы, и книга именно об этом.

Открою вам маленький секрет: моя первая попытка изучить Haskell окончилась весьма плачевно. Никакой другой язык не казался мне таким странным и сбивающим с толку (за исключением разве что Prolog и Uiua). Прошло много лет, прежде чем я снова вернулся к Haskell, и счастье мое не знало пределов!

А изменилась только подводка. В тот момент у меня за плечами уже было несколько лет опыт опыта работы с языками функционального программирования, и я был хорошо знаком с академическим жаргоном, неотъемлемым от рассказа о Haskell. Прежде всего я понимал, как его применять, а ведь язык обретает смысл, только если осмысленно применять его к решаемой задаче, – поэтому я считаю этот аспект критически важным для изучения языка. И в этой книге я хочу акцентировать на нем внимание.

Но зачем вообще изучать Haskell? Что мы от этого получим? Помимо права гордиться проделанной работой, мы получаем совершенно новый способ структурирования программ и обдумывания задач. Haskell – чистый функциональный язык, он учит нас минимизировать состояние, в результате чего программу становится принципиально проще понять. Кроме того, умение применять Haskell к реальным задачам – еще один инструмент в вашем багаже, благодаря которому вы совершенствуетесь как программист. Да и другие функциональные языки после Haskell обычно изучать гораздо проще. Неважно, хотите ли вы изучить Haskell, чтобы более корректно писать программы, из чистого любопытства или чтобы расширить горизонты, эта книга послужит вам путеводителем по глубокому и полному чудес миру Haskell.

Благодарности

Никакая книга не возникает из ничего и не пишется одиноким автором за один вечер (о, как бы я этого желал!) – и эта книга не исключение. Она никогда не появилась бы на свет без неустанной работы многих людей и без неоценимой помощи еще гораздо большего их числа.

Прежде всего я хочу поблагодарить тех, кто всего ближе ко мне в жизни: родителей за их непрестанную поддержку на протяжении всей моей жизни. Без вас я не стал бы тем, кто я есть, и за это я вам вечно благодарен. Кроме того, хочу сказать спасибо Али, всегда готового слушать мои тирады на темы программирования.

Далее я хочу выразить благодарность всему коллективу издательства Manning, а особенно моим редакторам, Патрику Барбу и Кэти Спосато, за слова ободрения и терпение, с которым они переносили меня и эту книгу. Отдельное спасибо Александру Вершилову, который настойчиво побуждал меня уделять внимание техническим аспектам Haskell. Александр 15 лет работал программистом, имеет степень PhD по физике, а сейчас возглавляет команду инженеров, работающих над образовательным проектом на Haskell. Большое спасибо Ивану Мартиновичу за помощь в технических вопросах процесса написания книги. Отдельная благодарность техническому редактору, Тимо фон Хольцу. Без твоих подсказок эта книга никогда не была бы доведена до конца.

Спасибо также бесчисленным рецензентам: Алексу Мэйсону, Алексею Выскубову, Энди Дженнингс, Анкиту Митталу, Антону Ричу, Кэмерону Пресли, Клиффорду Тэрберу, Давиду Паку, Дамиану Эстебану, Дэну Шейху, Эрнесто Босси, Эрнесто Гарсиа, Франсу Ойлинки, Гаутаме Садависаму, Грегорио Пиччоли, Гуннару Альбергу, Джеку Келли, Джоэлю Холмсу, Джуральдиньо Шикину, Каю Гэллиену, Кате Паткин, Кристофу Семьену, Киту Л. Манноку, Кэвину Чунгу, Магнусу Тэрнингу, Мансуру Мухитдинову, Марко Перроне, Марку Эльсону, Майклу Д. Роучу, Полу Силистеану, Тэду Мейеру и Винисиосу Венцу. Ваши предложения помогли сделать книгу лучше.

Наконец, было бы непростительным упущением не поблагодарить чудесное сообщество Haskell – и не только за то, что оно вдохнуло жизнь в этот язык, но и за постоянные нововведения, добавляемые с неустанным рвением. Пишущие на Haskell – одни из самых умных, талантливых и доброжелательных людей, с которыми мне приходилось общаться, и надеюсь, что эта книга привлечет новых адептов в наше прекрасное сообщество.

Об этой книге

Эта книга ставит задачей научить языку Haskell с нуля до создания практически полезных приложений, а не нудных упражнений. Слишком часто к Haskell относятся как к чисто академическому языку. Долой это предвзятое мнение! Haskell практичен настолько, насколько это нужно.

Сместив акцент на применение языка, мы поможем читателю лучше понять фундаментально присущие ему сильные стороны, а также то, как его красивые абстракции становятся не просто мысленным экспериментом, а реальным инструментом для реальной работы.

На кого рассчитана эта книга

Это книга для программистов, которые уже знакомы хотя бы с одним языком программирования и хотят распробовать на вкус функциональное программирование на Haskell. Предварительное знакомство с функциональным программированием не предполагается. Книга не является полным руководством по языку, но может послужить точкой входа, которая позволит читателю приступить к написанию собственного кода.

Структура книги

Эта книга состоит из 15 глав, в которых описываются девять программных проектов. Понятия и конструкции языка объясняются по мере необходимости.

- Глава 1 содержит введение в Haskell и книгу в целом. В ней приводятся исторические сведения и рассматриваются основы функционального программирования, не вдаваясь в специфику кода на Haskell.
- В главе 2 начинается вступительный проект: шифр Цезаря, древний алгоритм шифрования.

- В главах 3 и 4 мы познакомимся с клоном программы `nl` для подсчета строк текстового файла; это первая в книге исполняемая программа.
- Предметом глав 5 и 6 служит обработка данных и проектирование алгоритмов на примере игры в цепочку слов.
- В главах 7 и 8 мы познакомимся с практическим инструментом для разбора, сериализации, поиска и модификации CSV-файлов.
- Глава 9 посвящена проверке свойств и содержит практические советы по написанию комплектов тестов для вашей программы.
- В главах 10 и 11 рассматривается программный синтезатор со встроенным предметно-зависимым языком для сочинения музыки.
- Главы 12 и 13 содержат введение в библиотеку для разбора данных изображений и предоставляют распараллеленный конвейер преобразования изображений.
- Главы 14 и 15 посвящены работе с исключениями в приложении для синхронизации файлов.
- В главах 16 и 17 читатель научится писать сервер REST API с использованием JSON, базы данных и автоматически генерируемого клиентского приложения.

Вообще говоря, каждая глава опирается на знания, полученные в предыдущих главах, поэтому читать их лучше подряд, ничего не пропуская. Даже если сам проект вас не интересует, необходимо освоить излагаемые в главе ключевые концепции, прежде чем переходить к следующей.

О примерах кода

В этой книге много примеров кода как в пронумерованных листингах, так и в виде небольших фрагментов прямо в тексте. В обоих случаях исходный код набран `моноширинным шрифтом`, чтобы отделить его от обычного текста. Иногда код набран **`полужирным шрифтом`**, чтобы показать, что он отличается от предыдущих шагов в этой же главе, например когда в старую строку кода добавляется новая возможность.

Во многих случаях оригинальный исходный код переформатирован; мы разбивали строки и изменяли отступы, сообразуясь с размером печатной страницы. Иногда этого оказывалось недостаточно, и тогда мы добавляли маркеры продолжения строк (➥). Кроме того, комментарии, присутствующие в исходном коде, часто удалялись из листингов, если код описан в тексте. Многие листинги сопровождаются аннотациями, подчеркивающими важные концепции.

Исполняемые фрагменты кода можно найти в онлайновой версии книги на сайте https://livebook.manning.com/book/learn-haskell-by-example. Полный код примеров можно скачать бесплатно с сайта издательства Manning по адресу https://www.manning.com/books/learn-haskell-by-example, также он доступен на Github по адресу https://github.com/phagenlocher/learn-haskell-by-example.

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу dmkpress@gmail.com. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Manning Publications очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Об авторе

Филипп Хагенлохер (Philipp Hagenlocher) получил степень магистра информатики в Мюнхенском техническом университете, где приобрел солидные знания в области формальных методов и функционального программирования. Он работает программистом на Haskell на полной ставке и занимается распределенными системами, для которых абсолютная корректность программы – насущная необходимость. Филипп уже давно питает страсть к обучению других концепциям функционального программирования и на своем опыте знает, какие трудности могут возникнуть у читателя. Его серия роликов «Haskell for Imperative Programmers» на YouTube собрала миллион просмотров.