Душкин Р. В.

14 занимательных эссе о языке Haskell и функциональном программировании

ДМК Пресс
издательство
Москва, 2011

УДК 004.4
ББК 32.973.26-018.2
Д86

Душкин Р. В.
Д86 14 занимательных эссе о языке Haskell и функциональном программировании. – М.: ДМК Пресс, 2011. – 288 с., ил.

ISBN 978-5-94074-691-1

В книге представлено 14 статей автора, которые в разное время были опубликованы или подготовлены к публикации в научно-популярном журнале для школьников и учителей «Потенциал». Статьи расположены и связаны таким образом, чтобы они представляли собой логически последовательное повествование от начал к более сложным темам. Также в книге сделан упор на практические знания, предлагается решение многих прикладных задач при помощи языка функционального программирования Haskell.

Книга будет интересна всем, кто живо интересуется функциональным программированием, студентам технических ВУЗов, преподавателям информатики.

УДК 004.4
ББК 32.973.26-018.2

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 978-5-94074-691-1

© Душкин Р. В., 2011
© Оформление ДМК Пресс, 201

Принимаются благодарности

Вниманию всех читателей! Данная книга издана в электронном виде и распространяется абсолютно бесплатно. Вы можете свободно использовать её для чтения, копировать её для друзей, размещать в библиотеках на сайтах в сети Интернет, рассылать по электронной почте и при помощи иных средств передачи информации. Вы можете использовать текст книги частично или полностью в своих работах при условии размещения ссылок на оригинал и должном цитировании.

При этом автор будет несказанно рад получить читательскую благодарность, которая позволит как улучшить текст данной книги, так и более качественно подойти к подготовке следующих книг. Благодарности принимаются на счёт Яндекс.Деньги, на который можно перечислить малую лепту, и при помощи терминалов:

4100137733052

Убедительная просьба; при перечислении благодарности указывать в пояснении к переводу наименование книги или какое-либо иное указание на то, за что именно выражается благодарность.

Оглавление

От автора **7**

Типовой процесс разработки программ на языке Haskell **8**
Инструментальные средства . . . 8
Описание процесса разработки . . . 11

Функциональный подход в программировании **15**
Введение . . . 15
Общие свойства функций в функциональных языках программирования . . . 16
Примеры определения функций . . . 17
Заключение . . . 21

Алгебраические типы данных в языке Haskell **22**
Введение . . . 22
Простые перечисления . . . 23
Параметризация . . . 25
Параметрический полиморфизм . . . 27
Заключение . . . 28

Объектно-ориентированное и функциональное программирование **30**
Введение . . . 30
Именованные поля и структуры . . . 31
Классы типов . . . 34
Экземпляры классов . . . 35
Окончательные замечания . . . 37
Заключение . . . 39

Введение в λ-исчисление для начинающих **40**
Введение . . . 40
Неформальное описание теории . . . 41
Некоторые дополнения . . . 43
Редукция как стратегия вычислений . . . 43
Примеры кодирования данных и функций . . . 46
Заключение . . . 51

Комбинаторы? — Это просто! **52**
Введение . . . 52
Формальная теория . . . 52
Примеры сложных комбинаторов . . . 55
Модуль на языке Haskell для преобразования комбинаторов . . . 57

Представление данных и функций . . . 59
Булевские значения . . . 59
Нумералы Чёрча . . . 60
Упорядоченные пары . . . 60
Общие замечания . . . 61
Заключение . . . 61

Ввод и вывод на языке Haskell **63**
Введение . . . 63
Основы функционального ввода/вывода . . . 64
Стандартные функции ввода/вывода . . . 66
Примеры программ . . . 69
Вывод результатов исполнения функции на экран . . . 69
Альтернатива: экран или файл . . . 70
Копирование файлов . . . 71
Заключение . . . 72

Простой интерпретатор команд **73**
Введение . . . 73
Постановка задачи . . . 73
Основной набор функций . . . 74
Вспомогательные типы данных . . . 75
Цикл интерпретации . . . 76
Функции для исполнения команд . . . 78
Заключение . . . 80

Теория чисел и язык Haskell **81**
Введение . . . 81
Простейшие задачи . . . 82
Такие непростые простые числа . . . 83
Числа Мерсенна . . . 85
Числа Ферма . . . 86
Числа Софи Жермен . . . 86
Другие последовательности простых чисел . . . 86
Совершенству нет предела . . . 87
Заключение . . . 89

Магические квадраты и решение переборных задач **90**
Введение . . . 90
Простейший вариант перебора . . . 91
Перебор с использованием перестановок . . . 94
Перебор с использованием размещений . . . 96
Дальнейшая универсализация алгоритма . . . 99
Заключение . . . 102

Задача о ранце **103**
Введение . . . 103
Классическая задача . . . 104
Реализация решения на языке Haskell . . . 105
Заключение . . . 108

Кривая Дракона **109**
Введение . . . 109
Что такое Кривая Дракона? . . . 111
Алгоритм построения . . . 113
Реализация на языке Haskell . . . 113
Подготовительные описания геометрических образов . . . 114
Построение Кривой Дракона . . . 116
Заключение . . . 118

Немного о шахматных задачах **119**
Введение . . . 119
Вспомогательные программные сущности . . . 119
Задача о расстановке фигур . . . 122
Задача о ходе коня . . . 123

Генерация рекурсивных сказок **126**
Введение . . . 126
Колобок . . . 127
Теремок . . . 131
Обобщение функций и построение генератора . . . 133
Репка . . . 136
Заключение . . . 138

Литература **139**

От автора

В период с 2006 по 2009 год в образовательном журнале для старшеклассников и учителей «Потенциал» (ISSN 1814-6422) были опубликованы или подготовлены к публикации четырнадцать научно-популярных статей о функциональном программировании и языке Haskell. Многие из данных статей нашли определённый отклик в сердцах и умах читателей, поэтому автором было решено свести их в одну книгу, выстроив в более или менее стройную систему повествования.

Вместе с тем со времени начала публикации научно-популярных статей прошло достаточное количество времени, чтобы и язык Haskell, и функциональное программирование в целом эволюционировали, поэтому необходима и актуализация информации, приведение её в более современный вид. Всё это значит, что в настоящей книге исходные статьи не только не приведены в исходном порядке публикации, но и в некоторых случаях достаточно серьёзно переработаны.

Книга будет полезна школьникам старших классов, живо интересующимся информатикой и смежными областями науки и техники, а также учителям, которые проводят факультативные занятия по информатике и компьютерной грамотности. Кроме того, я надеюсь, что книга также будет полезна студентам технических высших учебных заведений, которые хотели бы овладеть пониманием функционального программирования в целом и языка Haskell в частности. Ну и в любом случае книга станет неплохим источником идей, задач и их решений для всех, кто интересуется функциональным программированием.

Я выражаю самую серьёзную благодарность всем моим коллегам, которые в той или иной мере способствовали появлению многих из приведённых в книге статей на свет. Особо стоит отметить Антонюка Д. А. и Астапова Д. Е. (в том числе и в качестве рецензента), чьи бессменные советы и предложения по улучшению всегда приходились кстати. Также хочу выразить благодарность всему коллективу редакции журнала «Потенциал» за его самоотверженный труд на поприще просвещения и популяризации науки среди подрастающего поколения. Отдельно хотелось бы отметить Ворожцова А. В., редактора отдела «Информатика».

Буду крайне признателен любым конструктивным комментариям, замечаниям и предложениям, которые можно присылать по адресу электронной почты `roman.dushkin@gmail.com`. Также по этому адресу можно присылать запросы на файлы с исходными кодами примеров, приведённых в книге (указывайте, пожалуйста, наименования эссе, для которых необходимо выслать исходные коды примеров).

В добрый путь.

Душкин Р. В.
Москва, 2011.

Типовой процесс разработки программ на языке Haskell

Статья была подготовлена к публикации в один из номеров журнала «Потенциал» в конце 2009 года. Опубликована не была.

В эссе рассказывается о типовом процессе разработки программных средств на функциональном языке программирования Haskell. Описываются некоторые из имеющихся на текущий момент программных средств и инструментов, делающих процесс разработки простым и быстрым. Даётся краткая суммарная информация о том, где и на каких условиях можно получить весь необходимый для работы инструментарий.

В дальнейшем изложении в настоящей книге приводятся разнообразнейшие задачи из области математики и информатики, а также описываются решения этих задач при помощи языка функционального программирования Haskell. Опыт общения автора с читателями показал, что имеется проблема непосредственно практического характера — многие читатели просто не знают, что делать с приводимыми примерами и исходными кодами: какой инструментарий использовать, как компилировать программы, как загружать дополнительные пакеты и т. д.

Осмысление проблемы привело к пониманию того, что язык Haskell при всех его достоинствах очень сложно входит в русло практического использования как любителями, так и профессиональными программистами. В связи с этим возникает необходимость в использовании новой парадигмы популяризации языка Haskell и функционального подхода в программировании (впервые введена в книге [8]).

Данная парадигма предполагает необходимость всемерного распространения знаний и информации о практических средствах и методах разработки, а теоретическая часть может быть получена заинтересованными людьми самостоятельно, благо в литературе именно по теоретической части недостатка нет (см. в том числе книги [6, 7, 14, 15, 16]). Этот шаг позволит на первоначальном этапе заинтересовать и через интерес привлечь к функциональному программированию множество неофитов.

Данное эссе описывает типовой процесс разработки программного средства на языке Haskell на примере простейшего приложения «Hello, world!». Сложность разрабатываемого программного средства здесь совершенно не имеет значения, поскольку обрисовываются именно процесс и инструментарий. Читатель волен самостоятельно применять описываемые в разделе средства для решения произвольных задач, хотя бы и для рассмотрения примеров в следующих эссе, входящих в состав книги.

Инструментальные средства

Весь набор инструментальных средств, используемых в работе над любым проектом, можно разделить на классы. Это относится не только к языку программирования Haskell, но и к любому другому языку

программирования, впрочем, как и к любым иным проектам — от проектирования детской игрушки до постройки города. В принципе, разделить на классы инструментальные средства для работы с языком Haskell можно так, как показано на рис. 1.

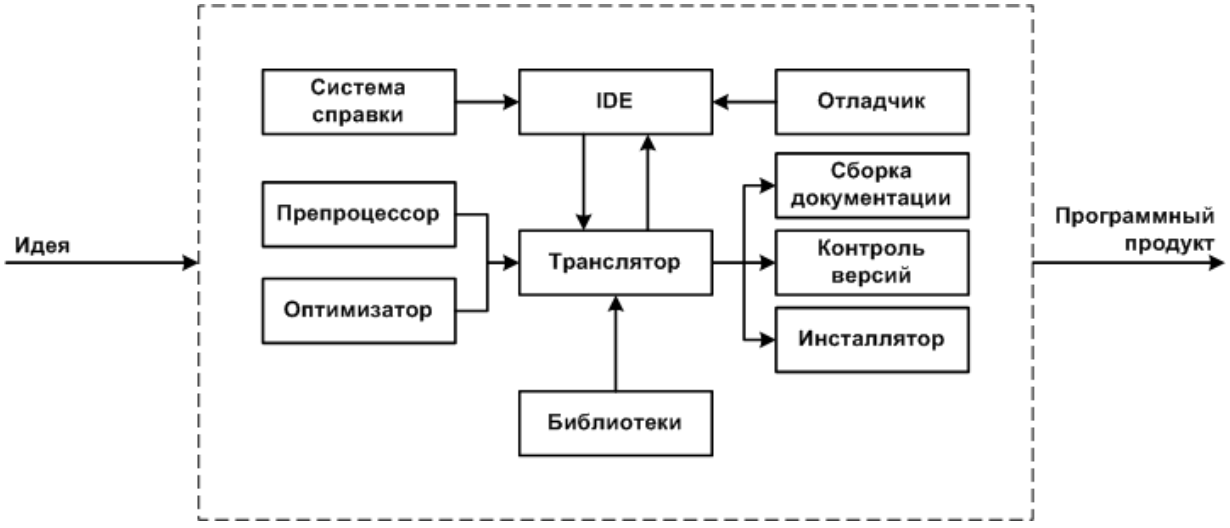


Рис. 1. Кибернетическая схема преобразования идеи в законченное программное средство

Также инструментарий можно классифицировать по степени полезности и необходимости. Некоторые классы инструментов обычно не используются (препроцессоры для языка Haskell используются редко), а без, к примеру, трансляторов языка вообще невозможно обойтись. Так что ниже в таблице перечислены наиболее часто используемые инструментальные средства, которые подходят для каждого разработчика, для любого проекта.

Инструмент	**Описание**
1	2
Компилятор GHC	Самый мощный и совершенный компилятор, на сегодняшний день не имеющий аналогов, — GHC. Реализует не только стандарт Haskell-98, но и множество расширений языка, многие из которых уже стали стандартом де-факто. Имеет возможность работы в режиме интерпретации (в состав поставки входит интерпретатор GHCi). `http://www.haskell.org/ghc/`
Тестирование QuickCheck	Для проведения тестирования обычно используются специальные библиотеки вроде QuickCheck. Они позволяют создавать семантические правила, описывающие поведения функций в проекте, после чего запускать эти правила на проверку с различными аргументами. Проверка осуществляется компилятором в процессе сборки программы. `http://www.md.chalmers.se/~rjmh/QuickCheck/`

Продолжение на следующей странице

1	2
Документирование Haddock	При написании исходных кодов можно сразу документировать программные сущности таким образом, что использование утилиты Haddock позволит собрать справочную систему для разработанного проекта в формате HTML. Это — правильный подход к разработке, поскольку документация позволяет без проблем передавать разработанный проект. `http://www.haskell.org/haddock/`
Контроль версий Darcs	Для хранения файлов с исходными кодами и ведения их версий можно воспользоваться утилитой распределённого хранения Darcs. Она обладает рядом дополнительных возможностей, превышающих стандартную функциональность систем контроля версий. `http://www.darcs.net/`
Система сборки Cabal	Утилита Cabal позволяет собрать пакет для распространения среди разработчиков и пользователей. Пакет собирается в стандартном формате, который может быть использован на произвольной платформе, поддерживающей язык Haskell. Кроме того, эта же утилита может установить в систему требуемый пакет, самостоятельно скачав его из сети Интернет. `http://www.haskell.org/cabal/`
Распространение Hackage	Для того чтобы каждый желающий смог воспользоваться разработанным пакетом, его можно положить в общий архив исходных кодов на языке Haskell Hackage. Этот архив является централизованным, и сегодня в нём хранятся сотни различных пакетов для решения практически любых задач. `http://hackage.haskell.org/`

Перечисленные в приведённой таблице инструментальные средства являются свободно распространяемым программным обеспечением, доступным для получения с указанных адресов в сети Интернет. Читателям настоятельно рекомендуется скачать перечисленные программные средства для возможности реальной работы с языком Haskell, что позволит без проблем работать над примерами, описанными далее в этой книге.