

Оглавление

Введение	9
I. Синтаксис и идиомы языка	11
1. Функции	12
1.1. Общий вид определения функций	12
1.1.1. Детальный разбор нескольких примеров определения функций	13
1.1.2. Ветвление	16
1.1.3. Замыкания	17
1.1.4. Бинарные операции	20
1.2. Технология сопоставления с образцами	23
1.2.1. Образцы вида $(\mathbf{n} + \mathbf{k})$	25
1.2.2. Именованные образцы	26
1.2.3. Ленивые образцы	27
1.3. Ввод и вывод	28
1.3.1. Действия ввода/вывода	28
1.3.2. Обработка исключений	32
1.4. Приёмы программирования	34
1.4.1. Двумерный синтаксис	34
1.4.2. Рекурсия и корекурсия	35
1.4.3. Накапливающий параметр и хвостовая рекурсия	39
1.4.4. Бесточечная нотация	41
1.4.5. Анонимные функции	42

1.4.6. Охрана	44
1.4.7. Определители списков	46
2. Типы данных	48
2.1. Базовые типы	48
2.1.1. Кортежи	49
2.1.2. Списки	51
2.2. Кратко об алгебраических типах данных	53
2.2.1. Перечисления	54
2.2.2. Простые структуры	56
2.2.3. Именованные поля	59
2.3. Синонимы типов	61
2.4. Параметрический полиморфизм	63
2.5. Типы функций	64
2.5.1. Функции как программные сущности с типом	64
2.5.2. Каррирование и частичное применение	66
2.5.3. Функции высшего порядка	68
3. Классы типов и экземпляры классов	71
3.1. Класс как интерфейс	71
3.2. Контекст и прикладные функции	76
3.3. Экземпляр — связь между типом и классом	78
3.3.1. Экземпляры класса <code>Logic</code>	80
3.4. Изоморфные типы	83
3.4.1. Определение нескольких экземпляров для уникальной пары (класс, тип)	85
3.5. Автоматическое построение экземпляров	86
3.6. Окончательные замечания о системе типов в языке Haskell	88
4. Модули	91
4.1. Система модулей	91
4.1.1. Экспорт программных сущностей	92
4.1.2. Импорт сторонних модулей	93
4.2. Абстракция данных при помощи модулей	97
4.3. Кое-что ещё о модулях	98

5. Сводная информация	100
II. Стандартные библиотеки	105
6. Стандартный модуль Prelude	108
6.1. Prelude: Алгебраические типы данных	108
6.2. Prelude: Классы и их экземпляры	115
6.3. Prelude: Функции	125
6.4. Prelude: Операторы	170
7. Пакет модулей Control	172
7.1. Модуль Applicative	172
7.2. Модуль Arrow	176
7.3. Модуль Concurrent	181
7.3.1. Модуль Chan	186
7.3.2. Модуль MVar	188
7.3.3. Модуль QSem	192
7.3.4. Модуль QSemN	193
7.3.5. Модуль SampleVar	194
7.4. Модуль Exception	197
7.5. Модуль Monad	211
7.5.1. Модуль Fix	220
7.5.2. Модуль Instances	222
7.5.3. Модуль ST	222
7.6. Модуль Parallel	224
8. Пакет модулей Data	226
8.1. Модуль Array	226
8.1.1. Модуль Base	231
8.1.2. Модуль Diff	231
8.1.3. Модуль IArray	233
8.1.4. Модуль IO	234
8.1.5. Модуль MArray	237
8.1.6. Модуль ST	241
8.1.7. Модуль Storable	243

8.1.8. Модуль <code>Unboxed</code>	245
8.2. Модуль <code>Bits</code>	245
8.3. Модуль <code>Bool</code>	247
8.4. Модуль <code>ByteString</code>	248
8.4.1. Модуль <code>Base</code>	277
8.4.2. Модуль <code>Char8</code>	286
8.4.3. Модуль <code>Lazy</code>	287
8.5. Модуль <code>Char</code>	288
8.6. Модуль <code>Complex</code>	298
8.7. Модуль <code>Dynamic</code>	300
8.8. Модуль <code>Either</code>	302
8.9. Модуль <code>Eq</code>	303
8.10. Модуль <code>Fixed</code>	304
8.11. Модуль <code>Foldable</code>	305
8.12. Модуль <code>Graph</code>	313
8.13. Модуль <code>HashTable</code>	320
8.14. Модуль <code>Int</code>	323
8.15. Модуль <code>IntMap</code>	324
8.16. Модуль <code>IntSet</code>	348
8.17. Модуль <code>IORef</code>	360
8.18. Модуль <code>Ix</code>	361
8.19. Модуль <code>List</code>	362
8.20. Модуль <code>Map</code>	374
8.21. Модуль <code>Maybe</code>	383
8.22. Модуль <code>Monoid</code>	385
8.23. Модуль <code>Ord</code>	388
8.24. Модуль <code>Ratio</code>	390
8.25. Модуль <code>Sequence</code>	390
8.26. Модуль <code>Set</code>	396
8.27. Модуль <code>STRef</code>	401
8.27.1. Модуль <code>Lazy</code>	402
8.27.2. Модуль <code>Strict</code>	402
8.28. Модуль <code>Traversable</code>	402
8.29. Модуль <code>Tree</code>	404
8.30. Модуль <code>Tuple</code>	408

8.31. Модуль <code>Typeable</code>	408
8.32. Модуль <code>Unique</code>	414
8.33. Модуль <code>Version</code>	415
8.34. Модуль <code>Word</code>	416
9. Пакет модулей <code>Debug</code>	419
9.1. Модуль <code>Trace</code>	419
10. Пакет модулей <code>Foreign</code>	421
10.1. Модуль <code>C</code>	422
10.1.1. Модуль <code>Error</code>	423
10.1.2. Модуль <code>String</code>	429
10.1.3. Модуль <code>Types</code>	436
10.2. Модуль <code>ForeignPtr</code>	439
10.3. Модуль <code>Marshal</code>	444
10.3.1. Модуль <code>Alloc</code>	445
10.3.2. Модуль <code>Array</code>	447
10.3.3. Модуль <code>Error</code>	453
10.3.4. Модуль <code>Pool</code>	455
10.3.5. Модуль <code>Utils</code>	459
10.4. Модуль <code>Ptr</code>	462
10.5. Модуль <code>StablePtr</code>	466
10.6. Модуль <code>Storable</code>	468
11. Пакет модулей <code>System</code>	471
11.1. Модуль <code>Cmd</code>	471
11.2. Модуль <code>CPUTime</code>	472
11.3. Модуль <code>Directory</code>	473
11.3.1. Модуль <code>Internals</code>	482
11.4. Модуль <code>Environment</code>	482
11.5. Модуль <code>Exit</code>	484
11.6. Модуль <code>Info</code>	485
11.7. Модуль <code>IO</code>	486
11.7.1. Модуль <code>Error</code>	497
11.7.2. Модуль <code>Unsafe</code>	507
11.8. Модуль <code>Locale</code>	507

11.9. Модуль Mem	510
11.9.1. Модуль StableName	510
11.9.2. Модуль Weak	512
11.10. Модуль Random	515
11.11. Модуль Time	519
12. Пакет модулей Text	528
12.1. Модуль Printf	528
12.2. Модуль Read	531
12.2.1. Модуль Lex	533
12.3. Модуль Show	535
12.3.1. Модуль Functions	536
Заключение	537
Литература	538

Введение

Язык Haskell является динамично развивающимся функциональным языком программирования, который получает всё больше и больше сторонников во всём мире, в том числе и в России. Этот язык вырвался из рамок научных лабораторий и стал языком программирования общего назначения. Вместе с тем хорошей литературы об этом прекрасном языке программирования категорически мало, тем более на русском языке.

В конце 2006 года из печати вышла первая и на текущий момент (2007 год) единственная книга на русском языке, рассматривающая функциональное программирование на языке Haskell [1]. Несмотря на то что в этой книге тема языка Haskell раскрыта практически полностью, его описание в ней страдает неполнотой и некоторой «поверхностью». С другой стороны, достаточно серьёзная математика в книге немного отпугивает неподготовленного читателя. Поэтому книга явилась своеобразным «первым блином», который необходим для первоначального ввода в проблематику. Однако в связи с ростом популярности как языка Haskell, так и парадигмы функционального программирования, необходимо больше всевозможных материалов, охватывающих различные аспекты и предназначенных для разной целевой аудитории.

Данная книга является кратким справочником по функциональному языку программирования Haskell стандарта Haskell-98 (без описания многочисленных расширений языка). В книге собрано описание знаний по успешному применению языка Haskell на практике. Она предназначена для тех, кто уже знает принципы функциональной парадигмы и сам язык Haskell. Это связано с тем, что, несмотря на то что практически всю информацию можно почерпнуть из интернета, очень часто необходимо иметь под рукой полноценный справочник, в котором мож-

но быстро найти ответы на специализированные вопросы. И эта книга как раз и предназначена для подобных целей.

Поскольку книга названа «кратким справочником», одним из принципов, которым руководствовался автор при её написании, является минимизация информации и предоставление компактно выраженных знаний, с достаточной степенью полноты раскрывающих смысл конструкций языка Haskell, идиом, существующих функций и других программных объектов, определённых в стандартных библиотеках. Поэтому стиль этого справочника является более или менее сухим и выдержанным, а описание программных сущностей наиболее формализованным.

Справочник разбит на две части. В первой части представлено краткое описание синтаксиса языка Haskell, а также наиболее часто и успешно используемые техники программирования на нём (ведь не секрет, что в каждом языке имеются свои особые методы «правильного» программирования). Во второй части описываются наиболее часто использующиеся стандартные модули, входящие в поставку двух наиболее известных трансляторов языка — HUGS 98 и GHC. Первая часть разбита на главы, каждая из которых описывает одну из пяти существующих в языке программных сущностей (и дополнительная шестая глава со сводной информацией). Главы второй части соответствуют стандартным библиотекам языка Haskell.

В целях единообразия представления информации в книге используется специальное форматирование текста, выделяющее определённые структурные элементы. Так, наименования программных сущностей выделяются моноширинным шрифтом обычного начертания: `head`, `True`, `Enum` и т. д. В отличие от идентификаторов ключевые слова записываются моноширинным шрифтом с подчёркиванием: `if`, `do`, `instance` и т. д. Знаки операций и специальные символы при записи ограничиваются круглыми скобками, чтобы выделить и отделить знаки от основного текста: `(+)`, `(>=)`, `(‘)` и т. д., в то время как сами скобки в случае необходимости записываются в кавычках: `«(», «|»`. Кроме того, наименования модулей, библиотек и специальных утилит также записываются моноширинным шрифтом: `Prelude`, `Data.List` и пр.

Краткость — сестра таланта, как говаривал русский классик А. П. Чехов. Поэтому осталось только упомянуть, что автор чрезвычайно благодарен Роганову В. А. за помощь в создании книги, и то, что автор будет рад получить комментарии и замечания по адресу электронной почты `darkus.14@gmail.com`.