

Введение в Haskell

1. Парадигма функционального программирования
2. Функции в Haskell
3. Операторы, их уровни приоритета и варианты ассоциативности
4. Условное выражение `if`
5. Списки, функции для работы с ними. Генераторы списков
6. Кортежи
7. Типы. Классы типов. Типы списков и кортежей
8. Типы функций. Каррирование функций. Композиция функций
9. Импортирование модулей
10. Сопоставление с образом и его применение в рекурсии
11. Охранные выражения и обработка ошибок
12. Конструкции `where` и `let`
13. Конструкция `case of`
14. Переменные типов и параметрически полиморфные функции
15. Функции высших порядков. Свертка. Лямбда функции
16. Создание типов. Ключевые слова `data` и `type`
17. Создание класса типов и его представителей
18. Полиморфный представитель класса типов
19. Расширение класса типов
20. Ленивые вычисления и строгие функции. Строгая свертка
21. Конструктор типов `Maybe`. Развертка списка
22. Ленивые образцы
23. Параметризованные типы. Конструктор типов `Either`
24. Строгие конструкторы
25. Обертка типа с помощью `newtype`
26. Компилятор GHC языка Haskell. Ввод и вывод с помощью IO
27. Работа с файлами. Тип `IOMode`. Обработка ошибок с `bracket`
28. Реализация списка дел. Проблема ленивости чтения
29. Функторы. Законы функторов. Примеры: список, `Maybe`, IO
30. Аппликативные функторы и их реализация
31. Моноиды. Примеры моноидов. Два моноида на одном множестве
32. Монады. Аксиомы монад. Реализация монады `Identity`
33. Монада `Maybe`. Монада списка. Задача о канатоходце
34. Нотация с помощью `do`
35. Монада `Writer`
36. Монада `Reader`
37. Монада `State`. Методы `get` и `put`
38. Монадические функции