Синтаксис языка Haskell

Haskell является классическим примером функционального языка программирования, и, следовательно, имеет довольно специфичный синтаксис, который для человека, не встречавшегося с функциональным программированием, будет довольно необычен. Для меня было очень неожиданно узнать, что в Haskell, в отличии от императивных языков программирования, например, Java, отсутствуют банальные циклы или даже оператор присваивания. С другой стороны, написав несколько программ, стало понятно, что это не является минусом языка, на Haskell программы получаются намного лаконичнее и понятнее, нежели на той же самой Java. Циклы очень хорошо заменяются рекурсией. Код программы легко читается и воспринимается. Например, алгоритм быстрой сортировки занимает на Haskell около 5 строчек, в то время как на Java это было бы несколько десятков строк кода. Плюсом является определение функции, которое не требуют немедленной ее реализации. Немного может ввести в замешательство объявлении функции нескольких переменных, например

Test :: Int -> String -> Int

С первого взгляда очень много стрелок и непонятно, что функция принимает, а что возвращает. Но если записать это в виде Test :: Int -> (String -> Int), то становится понятно, что функция принимает параметр типа Int и возвращает функцию, которая в свою очередь принимает параметр типа String и возвращает тип Int.

Одним из главных недостатков синтаксиса, на мой взгляд, является то, что неправильно поставленный пробел или перенос строки может поломать всю программу. Этим Haskell напоминает Python, где роль фигурных скобок играет количество пробелов. Код становится более прозрачным, но становится сложнее дебажить и искать ошибку.

В заключение хочу сказать, что несмотря на мой достаточно небольшой опыт программирования на Haskell это довольно интересно и необычно. Синтаксис языка имеет как положительные, так и отрицательные аспекты, но в целом это лишь вопрос привычки. Не бывает плохих или хороших языков, каждый создан для выполнения своих задач и целей.