**Синтаксис языка Haskell**

Haskell - функциональный язык программирования, вследствие чего его синтаксис довольно сильно отличается от синтаксиса классических процедурных или объектно-ориентированных языков программирования. Пожалуй, основным отличием Haskell от таких языков программирования является отсутствие переменных. Эта особенность является основной причиной других его отличий, например, отсутствие циклов. Программисту, не встречавшемуся ранее с функциональными языками программированиями, будет довольно сложно начать писать на Haskell вледствие этих особенностей. Основное, что стоит понять, начиная программировать на Haskell - в программе мы должны сообщить компьютеру **что** мы хотим сделать а не **как** мы хотим это сделать (в отличие от императивных языков программирования, в которых мы пишем последовательные инструкции).

Одной из основных особенностей языка является рекурсия, наличие которой компенсирует отсутствие циклов. Рекурсия присутствует во множестве других языков программирования, следовательно, ее использование в языке Haskell не должно поставить начинающего программиста в тупик.

Код на Haskell выходит довольно лаконичным и понятным, что несомненно является плюсом. Реализация многих основных алгоритмов в Haskell занимает значительно меньше строк кода чем, например, в Java или C#.

Стоит отметить также такую особенность Haskell как табулирование: знаки табуляции по своему интерпретируются компилятором, и от их правильной расстановки зависит правильность работы программы. Haskell не единственный язык, который использует такой подход. У этого есть свои как положительные, так и отрицательные стороны. К положительным стоит отнести возрастающую читабельность кода, к отрицательным же то, что единственный неправильно поставленный пробел может быть причиной неудачной компиляции программы.