**Сравнение синтаксисов Haskell и Java.**

В данном эссе мы сравним синтаксисы Haskell и Java. Сначала вкратце рассмотрим каждый из языков. Начнем с Java.

Java — ООП язык с функциональными особенностями, выпущен компанией Sun Microsystems. В последующем приобретён компанией Oracle. Приложения Java могут работать на любой виртуальной Java-машине вне зависимости от компьютерной архитектуры.

Haskell — чисто функциональный язык (нет ООП). Логично, что основная управляющая структура — функция. Тщательно проработана типизация. В связи с этим язык назван в честь исследователя теории типов и изобретателя комбинаторной логики Хаскелла Карри.

Будучи языками с кардинально различными ООП и функциональными возможностями, Haskell и Java весьма различаются по синтаксису. Допустим, в Haskell, в отличие от Java, нет переменных и циклов. Вместо этого — функции. Их объявление, конечно, отличается от Java-функций:

Haskell:

function::Obj1->Obj2->Obj3

function a = ...

function b = ...

...

Java:

private Obj function(Obj1 param1, Obj2 param2) {

...

return result;

}

В Java есть void-функции, которые не возвращают никакого значения. В Haskell же подобных функция не существует, так как любая его функция должна возвращать значение. Зато можно передать функцию в качестве параметра, чего нет в Java (было частично реализовано в Java 8). Присвоение типов также различаются:

Haskell:

5::Integer

Java:

(int) 5

В Haskell встроен механизм автоматического определения типа, поэтому приведенное выше выражение используется только в случае неоднозначности для интерпретатора. Напоследок скажем о выравнивании кода. В Java отступы влияют только на вид кода, его читаемость. А в Haskell они синтаксически разделяют блоки кода (вместо {}).

В эссе мы перечислили лишь часть отличий синтаксисов Haskell и Java. На деле различий несравненно больше. Большинство из них порождено расхождением языков по функциональной и ООП парадигмам.