



Haskell函数式程序设计

❤ by iCenter



Haskell



- Haskell是一种纯函数式编程语言
- 1980年代以前对函数编程有很多研究,但不同的研究者使用各自不同的语法规则,一起交流时造成一些不便. 后来1987年的时候,在FPCA'87会议上制定了统一的Haskell语言.
- Haskell吸收了各家的长处,是一种纯粹的函数编程语言,并根据科学家Haskell B.Curry的名字命名. Haskell经过多年的发展完善,目前使用的版本是Haskell 98.
- 哈斯凯尔·布鲁克·加里 (Haskell Brooks Curry)
- 三个编程语言Haskell、Brook、Curry,以及柯里化的概念都是以他的名字来命名的。

https://en.wikipedia.org/wiki/Haskell_Curry

Haskell的特点



- 1.Haskell具有引用透明性——以同样参数调用同一函数两次，得到的结果总是相同的
- 2.Haskell是惰性的——允许创建无限长度的数据结构
- 3.Haskell是静态类型的，支持类型推导
- 4.Haskell代码更为简洁

Haskell

<https://www.haskell.org/>

Haskell与C++、Python等区别

- 相对Haskell来说，传统的Basic,Pascal,C++,C#,Java,Python等都是命令(imperative)编程语言，程序语句有一定的执行次序。
 - 命令式编程是面向计算机硬件的抽象，命令式程序就是指导冯*诺依曼架构的计算机的操作序列。
 - 变量（对应着存储单元），赋值语句（获取，存储指令），表达式（内存引用和算术运算）和控制语句（跳转指令）
- 函数式编程是面向数学的抽象，将计算描述为一种表达式求值
 - 函数(functional)编程语言则给出执行的内容，关注于更高层次的"做什么"而不是"怎么做"，这就是二者最明显的一个区别。
- 换言之，Haskell语言是写给人看的，而不是写给机器看的。

Haskell

- 高阶函数 (Higher-order function)
- 偏应用函数 (Partially Applied Functions)
- 闭包 (Closure)
- 柯里化 (Currying)
- 惰性求值 (Lazy evaluation, 也称作call-by-need)

示例

- $a = 2$
- $b = 4$
- $c = a + b$
- 在函数编程中解释为: 定义常数 a 为2, 定义 b 为4, 定义 c 为 a, b 之和.
- 在命令编程中解释为: 给变量 a 赋值2, 给 b 赋值4, 求 a, b 之和赋给 c .
- 定义和赋值的区别在于, 定义不分次序, 赋值有次序, 以上程序在Haskell中完全可以倒过来写

Haskell工具

- 编辑器
- 文件格式要求.hs
- 编译器：Glasgow Haskell Compiler（GHC），Hugs
- <http://hackage.haskell.org/platform>

类型系统

- Eq类型类：可判断相等性的类型——“==”和“/=”
- Ord类型类：可比较大小的类型——“>”“<”“>=”“<=” 返回一个Bool类型值
'compare', 返回一个Ordering类型值，GT（大于），LT（小于），EQ（等于）
- Show类型类：可以表示为字符串的类型——show
- Read类型类：与Show相反——：： 分隔，加类型注释
- Enum类型类：有连续顺序的，值可以枚举的类型——succ（后继）和pred（前趋）
- Bounded类型类：有上下限的类型——maxBound和minBound
- Num类型类：表示数值的类型，包含实数和整数在内的所有数值类型
- Floating类型类：Float和Double两种类型
- Integral类型类：表示数值，仅包含整数

基本运算

- Linux或者Mac OS X系统：打开终端窗口、
- Windows：打开命令提示符，之后输入ghci并按回车键，进入交互模式

- Prelude> 2 + 15

- 17

执行顺序：按照运算符优先级顺序执行或者通过括号指定运算次序

- Prelude> 49 * 100

- 4900

表达式中存在负数常量，最好用括号括起来

- Prelude> 1892 - 1472

- 420

- Prelude> 5 / 2

- 2.5

逻辑运算

- Haskell的布尔值为True和False
- &&与
- ||或
- not非

函数调用

- 用编辑器创建函数
- `doubleMe x = x + x`
- 保存为**baby.hs**或任意名称，执行
- `: l baby`
- 装载函数
- `Prelude> doubleMe 9`
- `18`
- 修改.hs文档改变函数

Hello World

- Prelude> "hello" ++ "world"
- "hello world"
- Prelude> ['h' , 'e' , 'l' , 'l' , 'o'] ++ ['w' , 'o' , 'r' , 'l' , 'd']
- "hello world"
- Prelude> putStrLn "Hello World!"
- Hello World!

Lambda函数

λ



- λ 演算由Alonzo Church提出，记号
- Haskell 中lambda 函数通常也被称为匿名函数(anonymous function.)
- Haskell允许我们编写完全匿名的函数，这样就不必再费力地为函数想名字了。
- lambda函数是从 lambda 演算而来的。
- 在 Haskell 中，lambda函数以反斜杠符号 \ 为开始，后跟函数的参数（可以包含模式），而函数体定义在 -> 符号之后。其中，\ 符号读作 lambda。
- `>\ x -> x * x`

$$\lambda x \rightarrow x^2$$

```
square :: Integer -> Integer
square x = x * x
```

Haskell应用

- Verilog语言：Xilinx的电路编译器
- Haskell compiler for ML accelerator
 - Translating ML programs written in existing ML frameworks into the language of custom ML accelerator chips (an ASIC for running inference on TensorFlow) using a compiler written in Haskell.
- Halide <https://halide-lang.org/>
 - a language for fast, portable computation on images and tensors

谢谢指正！

