



A Linguagem Haskell

Origens e Influências

- Em 1987, ocorreu a conferência “Functional Programming Languages and Computer Architecture”, consolidação de linguagens existentes e criação de um padrão aberto.
- Em 1990, lançou-se a primeira versão, criada por Simon Peyton-Jones, Paul Hudak e Philip Walder e nomeada em homenagem ao lógico Haskell Curry.
- Surgiram novas versões até que em 1999 foi publicada o Haskell 98, que especifica uma versão mínima, estável e portátil da linguagem

Características

- Paradigma Funcional
 - Funções puras
- Tipagem estática
 - A tipagem de Haskell é forte. Cada expressão possui um tipo, e é possível obtê-lo em tempo de compilação
- Ausência de variáveis
 - Em Haskell há apenas constantes, não podemos fazer por exemplo, $a = 1$ e em seguida $a = 2$
- Tuplas heterogêneas
 - Ex: Podemos ter tuplas deste tipo: ["Wallace", 26, 1,65]
- Polimorfismo paramétrico
 - Podemos declarar funções que fazem cálculos diferentes dependendo do parâmetro

Expressividade

- Em Haskell há a seguinte tipagem “integer” que nos permite calcular números muito grandes. Esta tipagem possui a mesma característica que o “int”, só que com tamanho ilimitado
 - Ex: Fatorial de 99

Expressividade

- Variáveis anônimas

- Ex:

```
caso_logico :: Bool -> Bool -> Bool
```

```
caso_logico False _ = False
```

```
caso_logico True True = True
```

```
caso_logico _ False = False
```

Bibliografia

- <http://www.inf.ufes.br/~vitorsouza/wp-content/uploads/teaching-lp-20132-seminario-haskell.pdf>
- [https://pt.wikipedia.org/wiki/Haskell_\(linguagem_de_programa%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/Haskell_(linguagem_de_programa%C3%A7%C3%A3o))
- <https://pt.stackoverflow.com/questions/13359/pr%C3%B3s-e-contras-do-paradigma-funcional-e-do-haskell/14338>
- https://www.youtube.com/watch?v=QvZG_brQSaA