

Programação Funcional com Haskell

Prof. Dr. Eduardo Takeo Ueda
eduardo.tueda@sp.senac.br

Haskell no mundo real

- Exemplos open-source
 - **GHC** - compilador de Haskell escrito em Haskell
 - **Xmonad** - gerenciador de janelas
 - **Darcs** - sistema distribuído para gerenciamento de código-fonte
 - **Pandoc** - conversor de arquivos de um formato “markup” em outro
- Utilizações industriais
 - Sistemas críticos
 - Criptografia
 - Mercado financeiro
 - Jogos

Standard Prelude

- O módulo Prelude contém um grande conjunto de funções pré-definidas:
 - operadores e funções aritméticas
 - funções genéricas sobre listas
 - ... e muito mais
- O prelude padrão é automaticamente carregado pelo interpretador/compilador e pode ser usado em qualquer programa Haskell

Algumas funções do Prelude

```
> head [1,2,3,4]  
1
```

obter o 1º elemento

```
> tail [1,2,3,4]  
[2,3,4]
```

remover o 1º elemento

```
> length [1,2,3,4,5]  
5
```

comprimento

```
> take 3 [1,2,3,4,5]  
[1,2,3]
```

obter um prefixo

```
> drop 3 [1,2,3,4,5]  
[4,5]
```

remover um prefixo

Mais funções do Prelude

```
> [1,2,3] ++ [4,5]  
[1,2,3,4,5]
```

concatenar

```
> reverse [1,2,3,4,5]  
[5,4,3,2,1]
```

inverter

```
> [1,2,3,4,5] !! 3  
4
```

indexação

```
> sum [1,2,3,4,5]  
15
```

soma dos elementos

```
> product [1,2,3,4,5]  
120
```

produto dos elementos

Identificadores

- Os nomes de funções e argumentos devem **começar por letras minúsculas** e podem incluir letras, dígitos, sublinhados e apóstrofes:
 - fun1
 - x_2
 - y'
- As seguintes **palavras reservadas** não podem ser usadas como identificadores:

```
case class data default deriving do else  
if import in infix infixl infixr instance  
let module newtype of then type where
```

Comentário

- **Simple:** começa por `--` e vai até o final da linha
 - Exemplo: `-- comentário simples`
- **Delimitado:** começa com `{-` e termina com `-}`
 - Exemplo: `{- as duas linhas abaixo estão comentadas
segunda linha
terceira linha -}`

Tipos

- Um **tipo** é um nome para uma coleção de valores relacionados
- Por exemplo, o tipo *Bool* contém dois valores lógicos:
 - **True**
 - **False**

Erros de Tipos

- Algumas operações só fazem sentido com valores de determinados tipos. Por exemplo: não é possível somar números e valores lógicos
 - Hugs> 1 + False
Erro
- Em Haskell, estes erros são detetados classificando cada expressão do programa conforme o **tipo** do resultado

Tipos em Haskell

- Escrevemos **$e :: T$** para indicar que a expressão **e** assume o tipo **T**
- Uma **lista** é uma sequência de tamanho variável de elementos de um mesmo tipo
 - **$[False, True, False] :: [Bool]$**
 - **$['a', 'b', 'c'] :: [Char]$**
 - **$[1, 2, 3, 4] :: [Int]$**
- Uma **tupla** é uma sequência de tamanho fixo de elementos de tipos possivelmente diferentes
 - **$(42, 'a') :: (Int, Char)$**

Tipos Básicos em Haskell

Bool valores lógicos

True, False

Char caracteres simples

e.g. 'A', 'B', '?', '\n'

String sequências de caracteres

e.g. "Abba", "UB40"

Int inteiros de precisão fixa

e.g. entre -2^{31} e $2^{31} - 1$ em 32-bits

Integer inteiros de precisão arbitrária

apenas limitados pela memória do computador

Float vírgula flutuante de precisão simples

e.g. 3.14154, $-1.23e10$

Double vírgula flutuante de precisão dupla