

# Princípios de Programação

## Exercícios

Universidade de Lisboa  
Faculdade de Ciências  
Departamento de Informática  
Licenciatura em Engenharia Informática

2019/2020

### Tipos e classes de tipos

**Tópicos endereçados neste capítulo:** Declaração de tipos, os tipos Haskell mais comuns, variáveis de tipos, classes de tipos, as classes de tipos mais representativas.

1. Que tipos usaria para representar:
  - (a) Um ponto tridimensional
  - (b) Um número de 1 a 10
  - (c) Um polígono
  - (d) Um aluno: nome, número e nomes/notas das disciplinas feitas
  - (e) Os alunos de uma turma
  - (f) As palavras de um parágrafo
  - (g) Os parágrafos de um texto
2. Quais das seguintes frases são expressões Haskell?
  - (a) `['a', 'b', 'c']`
  - (b) `('a', 'b', 'c')`
  - (c) `['a', True]`
  - (d) `[True, False]`
  - (e) `["a_disciplina_de_PP", "é_fixe"]`
  - (f) `[('a', False), ('b', True)]`
  - (g) `[isDigit 'a', isLower 'f', isUpper 'h']`

- (h) `(['a', 'b'], [False, True])`
- (i) `(['a', 'b'], [False, True])`
- (j) `[isDigit, isLower, isUpper]`

3. Qual o valor das seguintes expressões?

- (a) `"Taprobana" < "As_armas_e_os_baroes"`
- (b) `"Taprobana" `compare` "As_armas_e_os_baroes"`
- (c) `show True ++ "_or_" ++ show False`
- (d) `show "As_armas_e_os_baroes"`
- (e) `read "True"`
- (f) `read (fst ("True", 27)) || False`
- (g) `[False .. True]`
- (h) `[True .. False]`
- (i) `2 + 3.5`
- (j) `(2 :: Double) + 3.5`
- (k) `fromIntegral 2 + 3.5`

4. Seja uma função com tipo `(Ord a, Num b) => (a -> a) -> a -> b`.  
O que ficamos a saber sobre os seus parâmetros e resultado?

5. Encontre o valor e um tipo para cada expressão abaixo.

- (a) `splitAt (length ['a', 'b', 'c', 'd']) [1..5]`
- (b) `tail (init (tail (init ['a'..'z'])))`
- (c) `take 2 (drop 4 (replicate 6 'p'))`

6. Qual o tipo (definido sobre classes de tipo) das seguintes funções?

- (a) `f1 x y = x < y`
- (b) `f2 x y z = x == y || z`
- (c) `f3 x y z = x == (y || z)`
- (d) `f4 x y = show x ++ y`
- (e) `f5 x y = show (x ++ y)`
- (f) `f6 x y z = x + y > z`

Nota: para obter o tipo de uma qualquer expressão no `ghci` utilize o comando `:t <expressao>`. Por exemplo `:t f1` ou `:t show` ou `:t show 1`.

7. Verdadeiro ou falso?

- (a) `f1` tem tipo `Int -> Int -> Bool`
- (b) `f1` tem tipo `Integer -> Integer -> Bool`

- (c) f1 tem tipo `Int -> Integer -> Bool`
- (d) f2 tem tipo `[Char] -> [Char] -> Bool -> Bool`
- (e) f2 tem tipo `[a] -> [a] -> Bool -> Bool`
- (f) f4 tem tipo `Bool -> [Char] -> [Char]`
- (g) f4 tem tipo `(Int -> Int) -> [Char] -> [Char]`

8. Qual o tipo mais geral das seguintes funções?

- (a) `second xs = head (tail xs)`
- (b) `swap (x,y) = (y,x)`
- (c) `pair x y = (x,y)`
- (d) `double x = 2*x`
- (e) `palin xs = reverse xs == xs`
- (f) `twice f x = f (f x)`