

Programação Funcional

Roteiro de atividades práticas 2: Haskell – Funções, Tipos e Tuplas

Esse roteiro deve ser desenvolvido de forma assíncrona pelo aluno. Para que essas atividades sejam avaliadas e contabilizadas (nota e presença) o arquivo .hs referente às atividades abaixo deve ser enviado para o e-mail claudineyrt@gmail.com.

Data máxima de envio: até 18/03/2021 (Quinta) até 23H59

1) Reescreva as funções do último roteiro (*dobro*, *quadruplicar*, *hipotenusa*, *distância*), definindo a prototipação de tipos de cada função. Obs: se você já fez isso no primeiro roteiro, envie novamente para o exercício ficar completo.

2) Abra o ambiente interativo GHCi e avalie as seguintes expressões.

```
fst (2,5)
snd (5, "Bom dia")
fst(snd ("Ola", (1,2)))
snd(fst ("Ola", (1,2)))
(1,1) == (1,1)
(1,1) /= (1,1)
(1,1) < (1,2)
(2,1) < (1,2)
(1,2,3) < (1,2)
"azul" < "verde"
"azul" < "amarelo"
(1,2,3) == (,,) 1 2 3
```

3) Dado um valor monetário em reais, escreva uma função `conversao` que retorna uma tupla-3 com o valor em Real, Dolar e Euro, sendo que 1 real = 3,96 dólares = 4,45 euros.

4) Implemente a função `bissexto` que, dado um ano (inteiro), indique se ele é bissexto ou não.

5) Defina o tipo `Data` dado a seguir. Escreva a função `bissexto2` que recebe uma data e indique se ela pertence a um ano bissexto ou não.

```
type Data = (Int, Int, Int)
bissexto2 :: Data -> Bool
```

6) Escreva a função `valida` que indica se uma data é válida ou não.

```
valida :: Data -> Bool
```

7) Escreva a função `precede` que recebe 2 datas e indica se a 1a data é anterior à 2a.

```
precede :: Data -> Data -> Bool
```

8) Implemente as estruturas de dados (tuplas) para um sistema de gerenciamento de bibliotecas e depois as defina como tipos. O sistema tem 3 estruturas básicas:

Livro: composto por código do livro, título do livro, autor, editora e ano de publicação.

Aluno: composto por código do aluno, nome, e-mail e telefone.

Empréstimo: composto por código do livro, código do aluno, data de empréstimo, data de devolução e situação. Obs: utilize a estrutura/tipo auxiliar **data** do exercício 4.

9) Seja o tipo `Emprestimo` e o exemplo dado a seguir, composto por código do livro, código do aluno, data de empréstimo, data de devolução e situação. Escreva uma função que verifica se um empréstimo está em dia, dado um empréstimo e a data de hoje.

```
type Emprestimo = (String, String, Data, Data, String)
el::Emprestimo
e1 = ("H123C9", "BSI200945", (12, 9, 2009), (20, 9, 2009), "aberto")
```