

Lista 06

Programação Funcional

Prof. Maycon Amaro

Exercício 1

Os seguintes trechos de código não compilarão devido a erros de tipo. Para cada um, explique a causa do erro e como resolver.

Código 1

```
data Pessoa = Pessoa String Int

amy :: Pessoa
amy = Pessoa "Amy Winehouse" 27

main :: IO ()
main = putStrLn (show amy)
```

Código 2

```
data Humor = Feliz | Triste

alegrar :: Humor -> Humor
alegrar x = if x == Triste then Feliz else Feliz
```

Código 3

```
toLower :: Char -> Char
toLower = (+) 32
```

Exercício 2

Tente derivar a implementação de cada uma das funções a seguir, tendo apenas a informação de seu tipo. Lembre-se de usar as funções do prelúdio se conveniente. Evite criar implementações que não usam toda a informação de tipo.

Exemplo

```
ruim :: Ord a => a -> a -> a
ruim x y = x -- A classe Ord ficou inútil
```

```
bom :: Ord a => a -> a -> a
bom x y = if x <= y then x else y
```

Função 1

```
chk :: Eq b => (a -> b) -> a -> b -> Bool
```

Função 2

```
arith :: Num b => (a -> b) -> Integer -> a -> b
```

Função 3

```
cncat :: (Show a, Show b) => a -> b -> String
```

Exercício 3

Uma classe de tipos nos permite obter polimorfismo com garantias estáticas, ou seja, verificadas pelo compilador antes da execução do programa. Pesquise sobre a classe de tipos `Read` e reflita quais são os benefícios e limitações dela, com respeito às garantias estáticas.