

Minicurso de Haskell



Exercícios



Dado o tipo algébrico Temperature

Crie instâncias de Show e Eq.

Eq deve converter a temperatura e compará-las na mesma unidade.

Tal que $(1.8 * \text{celsius} + 32) = \text{fahrenheit}$

```
data Temperature =
```

```
    F Float
```

```
  | C Float
```

Dado o tipo algébrico Shape

Implemente uma função

`area :: Shape -> Double`

Use a função `area` para definir a
`typeclass` `Eq` e `Ord`.

A `typeclass` `Ord` implementa a função
(`<=`) e permite o uso de outros
comparadores como (`>`).

```
data Shape =  
    Circle Double  
| Rectangle Double Double  
| Triangle Double Double  
deriving (Show)
```

Dado o tipo algébrico Solid

Implemente a *typeclass*

class HasArea a where

area :: a -> Double

para Solid e Shape

```
data Solid =
```

```
    Sphere Double
```

```
  | Cuboid Double Double Double
```

```
deriving (Show)
```

area (Sphere r) = 4 * pi * r²

area (Cuboid w h d) = 2 * (w*h + w*d + h*d)

Texto com Metadados

Implemente a typeclass Semigroup para este tipo.

A typeclass Semigroup define uma operação associativa (<>). Esta operação deve combinar os valores de dois MetaText.

```
data MetaText = MetaText {  
    text :: String,  
    wordC :: Int,  
    lineC :: Int  
}  
deriving (Show, Eq)
```

```
fromString str = MetaText str (length (words str)) (length (lines str))
```