Functional Programming for BDA - List 3

User-defined types and classes, trees.

Marcin Michalski, DCS FFPT WUST 2019/2020

Treat Exercises as a warm up. Submit Tasks 1 and 2.

Exercise 1. Let us define

```
data Point = Point Float Float
data Shape = Circle Point Float | Rectangle Point Point
```

where `Circle x y` models a planar circle with the center x and the radius y and `Rectangle x y` models a rectangle with top-left corner x and bottom-right corner y. Implement a function that calculates the surface of a given shape.

Exercise 2. Let us define

```
data Vector3D a = Vector a a a
```

that models 3D vectors. Define addition, multiplication by a scalar and scalar multiplication for your vectors.

Exercise 3. Consider the following IntOrString type

```
data IntOrString = Word String | Number Int
```

Make it an instance of classes `Eq` and `Show`.

Task 1. Let us recall the definition of a binary tree structure

```
data Tree a = Leaf a | Node (Tree a) a (Tree a)
```

a) Implement `foldr` that starts with the right-most branch.

b) Implement functions that count numbers of roots and leafs.

c) Implement a function that determines whether a given x is an element of a given tree.

d) Implement a function that determines the height of a given tree, i.e. the length of the longest branch.
Hint: implement a special kind of fold that ignores values attached to leafs and roots or use fmap to alter the tree so it suits your needs.

Task 2. a) Define trees with roots that may have any number of children (*Hint:* `[Tree a]`) and make it a nice looking instance of the class `Show`.

b) Make it an instance of the class `Functor`.

c) Make it a partial instance of the class `Foldable` (i.e. define some kind of fold).

d) Repeat b) - d) of the previous Task.