

Exercícios sobre Type Classes

prof. André Rauber Du Bois

Universidade Federal de Pelotas
<http://sites.google.com/site/haskellufpel/>
dubois@inf.ufpel.edu.br

1 Questionário

1. Dada a seguinte definição de listas

```
data Lista a = Cons a (Lista a) | Vazio
```

criar uma instância da classe `show` na qual a lista é convertida para o formato tradicional do Haskell:

```
> Cons 1 (Cons 2 (Cons 3 Vazio))  
[1,2,3]
```

2. Dada a seguinte definição de árvore

```
data Tree a = Nodo a (Tree a) (Tree a) | Folha a
```

criar uma instância da classe `show` que imprime essa árvore como se fosse uma lista

3. Dada a seguinte definição de árvore

```
data Arvore a b = NodoA a (Arvore a b) (Arvore a b)  
                  | NodoB b (Arvore a b) (Arvore a b)  
                  | FolhaA a | FolhaB b
```

criar uma instância da classe `show` que imprime essa árvore como se fosse uma tupla de listas, onde a primeira lista contém os elementos do tipo `a` e a segunda lista contém os elementos de tipo `b`

4. Instanciar o tipo `SocioClube` na classe `Ord` e testar o algoritmo insertion sort com uma lista de sócios
5. Criar uma classe `ParaLista` que agrega todos os tipos que podem ser convertidos para listas. A classe deve possuir a seguinte função que deve ser sobrecarregada para as instâncias:

```
toList :: a -> [a]
```

Instanciar os tipos `Char`, `Int` nesta classe.