

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO
TERCEIRA LISTA DE PROGRAMAÇÃO FUNCIONAL

Profa. Leila Maciel de Almeida e Silva

Data de entrega : 03/09/2019

1. Defina um tipo abstrato de dados que é uma árvore binária de busca em que o tipo exportado é `data Arv a b c = NoNulo | No (a,b,c) (Arv a b c) (Arv a b c)` e exporta as funções: `arvVazia`, `ehVazia`, `ehNoNulo`, `arvEsq`, `arvDir`, `infoNo`, `consultaNo`, `insereNo`, `removeNo`. A chave da árvore pela qual os nós serão organizados, inseridos, consultados e removidos é o primeiro elemento da tupla. Caso seja necessário derivar classes de tipos na definição do TAD faz parte da questão a identificação de quais classes devem derivar.

Usando o tipo abstrato de dados definido no item (1), retome o problema do restaurante, com a categorização dos códigos abaixo, como na lista 2, e faça o que se pede a seguir.

Item	Faixa de códigos
Bebidas	1 - 50
Tira-gostos	51-100
Carnes	101-120
Aves	121-140
Peixes e Mariscos	141-160
Massas	161-180
Acompanhamentos extras	181-200
Sobremesas	201-220

- .2. Considere que o cardápio do restaurante será agora armazenado na forma da ABB do item 1 onde a informação do nó é do tipo `ItemRest`. Escreva as funções a seguir para manipular o cardápio do restaurante armazenado no sistema.

- (a) Coleta um item no menu, informando seu código.

```
coletaItemMenu :: Menu -> Codigo -> ItemRest
```

- (b) Coleta um item no menu, informando seu nome.

```
coletaItemMenu :: Menu -> Nome -> ItemRest
```

- (c) Atualizar os preços do cardápio de um valor percentual informado. Observe que como o preço é sinônimo de `Int`, tudo deve ser expresso usando o tipo `Int`. Assim, se um item custa 1000 e você deseja acrescer de um valor percentual de 10%, você deve informar 10 e sua atualização calculará o novo valor como o valor antigo acrescido de `div (10*1000) 100`.

```
atualizaPrecosMenu :: Menu -> Int -> Menu
```

- (d) Atualizar os preços de uma categoria do cardápio de um valor percentual informado. Para resolver esta questão você pode precisar de uma função auxiliar, que dado a categoria lhe devolva

o maior e menor código desta categoria. Observe que a faixa de códigos e as possíveis categorias são fixas e não serão atualizadas pelo sistema. Para realizar a questão defina o tipo `Categoria` como um tipo algébrico em que os itens da categoria são os construtores deste tipo.

```
type Categoria = ?? defina você??

atualizaPrecosCat :: Menu -> Categoria -> Int -> Menu

catMaiorMenorCod :: Categoria -> (Codigo,Codigo)
```

(e) Inserir um conjunto de itens novo no menu. Sua função só deve inserir se o item ainda não existir no menu.

```
insereItemMenu :: Menu -> [ItemRest] -> Menu
```

(f) Remove um conjunto de itens do menu. Sua função só deve remover se o item existir no menu.

```
removeItemMenu :: Menu -> [ItemRest] -> Menu
```

(g) Coletar todos os itens de uma dada categoria.

```
coletaItensCat :: Menu -> Categoria -> [ItemRest]
```

(h) Contar quantos itens existem no menu.

```
totalMenu :: Menu -> Int
```

(i) Listar quantos itens existem no menu, por categoria.

```
type NomeCat = String

totalCat :: Menu -> [(NomeCat,Int)]
```

(j) Coletar o nome do item mais barato e o mais caro, por categoria.

```
itemCaroBarat :: Menu -> Categoria -> [(NomeCat, (Nome, Nome))]
```

3. Mantenha as funções que já implementou para gerir os pedidos dos clientes no restaurante. Mas reescreva esta função considerando um tipo algébrico para expressar os dias da semana.

(a) Escreva uma função para dar desconto de um dado percentual, de acordo com o dia da semana, nos itens dos pedidos do cliente daquele dia: segunda - bebidas; terça – peixes e mariscos; quarta – tira-gostos; quinta – carnes e aves; sexta-massas; sábado e domingo – acompanhamentos extras e sobremesas. Esta função deve ser aplicada após a função `pedidoCompletoMesa` quando houver eventos promocionais, antes da geração da conta para o cliente.

```
type DiaSemana = ?? defina você ??

desconto :: [(Quant,Nome,Preco)] -> Menu -> DiaSemana -> Int ->
           [(Quant,Nome,Preco)]
```