

Lista de Exercícios Cap.s 12 e 18 – Composição de funções e Expressões locais

Fundamentos de Algoritmos – INF05008

Profas. Leila Ribeiro e Ana Bazzan
Professor Responsável (esta lista): Ana L. C. Bazzan

1. Refaça o exercício 3 da lista nr. 7 (ver enunciado abaixo), agora usando, necessariamente, uma expressão `local`. Todos os dados devem ser definidos conforme instruções e template definidos na lista nr. 7, ainda que isto já tenha sido feito anteriormente.

Enunciado:

Escreva a definição da função `maisIdoso-1`, que retorna o ancestral mais idoso de um nó dado (que pode ser o próprio). Se houver mais de um com mesma data de nascimento, retorna um deles. A função `d1<=d2?`, vista na lista nr. 7, deve ser usada para comparar os anos de nascimento.

2. Explique as vantagens do uso da expressão `local` no exercício anterior. (no Dr. Racket, use um box para comentar este texto.)
3. Ainda sobre os dados das questões anteriores, queremos avaliar sua compreensão sobre o tema composição de funções. Imagine que se quer construir um programa para, dado um nó, retornar uma lista contendo este nó bem como os nós de todos os seus ancestrais, ordenados por data de nascimento em ordem crescente de ano. Nesta questão, faça um projeto para este programa, descrevendo **apenas os contratos, objetivos e exemplos de uso** das seguintes funções:
 - (a) `loa` (lista ordenada de ancestrais) que, dado um nó, retorna uma lista contendo este nó e todos seus ancestrais, ordenados pela data de nascimento, em ordem crescente do ano;
 - (b) das adaptações (para os tipos adequados usados neste projeto) das funções `ordena` e `insere` vistas em aula;
 - (c) qualquer / quaisquer outra(s) função / funções auxiliar(es) que precise(m) ser utilizada(s) no seu projeto de solução.

Não é preciso desenvolver as funções. Porém, todo e qualquer dado usado nos contratos tem que ser definido.

Obs.: há vários projetos possíveis para resolver esta questão. O que importa é que a composição das funções seja apropriada. Relembrando, vimos composição de funções no início do semestre.

O exercícios 4 e 5 desta lista continuam trabalhando com parte dos tipos de dados vistos anteriormente na lista sobre estruturas (lista nr. 5).

Uma locadora dispõe de carros locação. Estes carros possuem estes atributos: ano de fabricação, modelo (descrição), valor de mercado, se o carro tem ar condicionado, se tem direção hidráulica e se tem vidros elétricos.

A locadora mantém um cadastro de todos os carros disponíveis para locação, onde os atributos de cada um são conforme definido anteriormente.

Faça o que é pedido:

4. Complete a definição de dados:

```
(define-struct carro (ano modelo valor ar? direção? vidros-el?))  
;;Um elemento carro do conjunto Carro é uma estrutura  
(make-carro ano modelo valor ar? direção? vidros-el?)  
;;onde:  
  
; ano : _____ é o ano de fabricação do veículo  
  
; modelo : String, é o modelo do veículo  
  
; valor : _____ é o valor em reais do veículo  
  
; ar? : _____ indica se veículo possui ar condicionado  
  
; vidros-el? : _____ indica se o veículo possui vidros elétricos  
  
; direção? : _____ indica se veículo possui direção hidráulica.  
  
; uma listaDeCarros é:  
  
_____  
  
_____
```

5. Crie duas versões de um programa que, dada um cadastro (lista de carros) com pelo menos um item, retorna o carro com o maior preço de mercado. Estas duas versões devem:
- (a) no caso da primeira versão, considerar que a lista não está necessariamente ordenada pelo atributo preço de mercado; denomine a função `maiorvalor`.
 - (b) no caso da segunda versão, faça a função `maiorvalor_ord` considerando que a lista recebida está, necessariamente, ordenada em ordem decrescente, por preço de mercado.
6. Defina a função `ordena-l` que encapsula, numa única função, as funções `ordena` e `insere`, cuja finalidade, como visto em aula, é ordenar uma lista de números. O encapsulamento se dá através do uso de uma expressão `local` no escopo da própria `ordena-l`.