

Lista de Exercícios 10 (Caps. 19 e 20) – INF05008

Siga as instruções sobre elaboração de exercícios de INF05008. Você **deve** usar funções de alta-ordem em **todas as questões** (exceto na 2a, que é para definir tipos de dados e não funções).

1. Considere as definições de dados e os programas do template (questão 1). Reescreva o código das funções dos itens (A) a (E) usando as funções `filter`, `map` e/ou `foldl`. O código das novas funções não pode ter nenhuma expressão condicional (`cond` ou qualquer outra expressão condicional), mas deve preservar o comportamento da função, ou seja, quem chama a função não deve notar nenhuma diferença entre as duas versões, são apenas duas formas diferentes de implementar a mesma função. As novas funções devem ter os mesmos nomes das originais, então apague ou comente as funções originais. Inclua pelo menos 4 testes para cada uma das funções (A) e (E), sendo que para a funções que geram imagens, os testes devem ser como os da função `desenha`, ou seja, indicar em código (e não através de imagens) o desenho resultante da execução da função, sendo que este código só pode conter as seguintes funções pré-definidas de desenho do Racket : `triangle`, `rectangle`, `ellipse`, `beside`, `empty-image`.
2. Um professor tem várias turmas de alunos e quer automatizar a geração dos conceitos de suas turmas. Cada registro de aluno contém o nome do aluno, sua turma e as notas da primeira e segunda provas e a nota do trabalho.
 - (a) Faça uma definição de dados para representar os alunos de um professor, gerando dois tipos: o tipo `Aluno`, que deverá ser uma estrutura chamada `aluno` com 5 campos, representando seu nome (string), turma (string) e notas das provas 1 e 2 e do trabalho (números), nesta ordem; e o tipo `ListaDeAlunos`, que é a lista de todos os alunos deste professor.
 - (b) Construa o programa `lista-alunos-turma` que, dada a lista de alunos de um professor e uma turma, nesta ordem, devolve todos alunos daquela turma, na ordem na qual eles aparecem na lista de alunos. *(Desafio – não vale nota: Você conseguiria modificar esta função para devolver os alunos na ordem invertida em relação à ordem original?)*
 - (c) Imagine que a média de um aluno é calculada como a média aritmética das 3 notas (das 2 provas e do trabalho). Construa a função `gera-conceitos` que, dada a lista de alunos do professor, gera uma lista com os nomes dos alunos e seus conceitos correspondentes: médias entre 9 e 10 (inclusive) geram conceito A, entre 7.5 (inclusive) e 9 (exclusive) geram conceito B, entre 6 (inclusive) e 7.5 (exclusive) geram conceito C, nos outros casos o conceito é D. Note que aqui o resultado é uma lista de pares contendo o nome do aluno e seu conceito. Cada par deve ser um elemento de uma estrutura definida com o nome `par`.
 - (d) O professor gostaria de avaliar o impacto de usar diferentes formas de calcular a média dos alunos nos seus conceitos. Para isso, generalize a função do exercício anterior incluindo como segundo argumento uma função que calcula a média. A nova função deve se chamar `gera-conceitos-gen` e deve ser uma função de alta-ordem. Dê exemplos de sua função usando, pelo menos, 3 funções diferentes para calcular a média de um aluno.
 - (e) Desenvolva a função de alta-ordem `ordena-turma` que, dados um critério de ordenação, a lista de alunos de um professor e uma turma, ordena os alunos da turma segundo este critério, devolvendo a lista de alunos desta turma ordenada. Sua função deve ser de alta-ordem (o critério deve ser uma função), e você deve fazer pelo menos 3 testes usando critérios de ordenação diferentes. Se você quiser comparar strings, existem as funções `string<?` e `string>?` (e também as versões com `<=` e `>=`) que comparam os strings de acordo com a ordem lexicográfica. Para a ordenação, adapte as funções `ordena` e `insere` vistas em aula.

Desafio (não vale nota): Você notou que as funções usadas como critérios de ordenação são parecidas em sua estrutura? Você conseguiria generalizar, construindo uma função para gerar critérios de ordenação para listas de alunos? A ideia seria, por exemplo, gerar critérios de comparação baseados no conceito de um aluno e em alguma função de comparação de strings; ou na média dos alunos e em alguma função de comparação de números, etc. Dica: Esta função recebe dois argumentos, que são funções. O resultado também é uma função, e por isso a implementação da função deve usar expressões lambda (para definir a função que será retornada como resultado).