

INF 05008 - Fundamentos de Algoritmos 2020/2
Lista de exercícios 4
Estruturas (Capítulo 6)

Instruções:

- Escreva todas as expressões e funções pedidas nos exercícios no ambiente de sua preferência (*DrRacket* ou *WeScheme*).
- Salve seu programa em um arquivo e envie pelo *Moodle*. O nome do arquivo deve ser do tipo:

lista4-NomeDoAluno-TurmaDoAluno.rkt

onde em NomeDoAluno você deve colocar o seu nome e em TurmaDoAluno você deve colocar a sua turma (A, B, C ou D). **Não coloque espaços nem acentos no nome do arquivo.**

- Preste atenção no **nome das funções** e na **ordem dos argumentos** pedidos nos exercícios.
- Para facilitar, utilize o **modelo** de arquivo fornecido no *Moodle*.
- Em todas as questões, **deve ser colocada a documentação completa**, ou seja, contrato, objetivo, exemplos e testes.
- Nas questões que novos tipos são criados, **deve ser colocada a definição do novo tipo criado**.
- É sempre bom você quebrar os problemas maiores em problemas menores. Isso deixa seu código mais elegante e legível, além de facilitar a resolução do problema.
- Fique atento ao **prazo** para a entrega da lista.

Exercícios

Para realizar os exercícios desta lista, utilize o seguinte tipo Equipe como base:

```
;; DEFINICAO DO TIPO EQUIPE:

(define-struct equipe (nome tamanho emblema pontuacao))
;; Um elemento do conjunto Equipe eh
;; (make-equipe um-nome um-integrantes um-emblema uma-pontuacao) onde
;;   um-nome: String, eh o nome da equipe
;;   um-integrantes: Numero, eh o numero de integrantes da equipe
;;   um-emblema: Imagem, eh o emblema da equipe
;;   uma-pontuacao: Numero, eh o numero de pontos da equipe
```

Exercício 1. Defina 4 instâncias do tipo Equipe. Pode escolher os nomes que quiser, e invente emblemas como os exemplos abaixo:

```
(define LOSANGO (rhombus 20 60 "solid" "yellow")) ;; losango amarelo

(define EMBLEMA1 (overlay
  (star 30 "solid" "white")
  (circle 50 "solid" "red")))
(define EMBLEMA2 (overlay
  (above LOSANGO (beside LOSANGO LOSANGO))
  (rectangle 100 100 "solid" "blue")))
```

Dica 1: Use a documentação do pacote de imagens do *Racket* para criar emblemas mais sofisticados: <https://docs.racket-lang.org/teachpack/2htdpimage.html>.

Exercício 2. Faça uma função chamada `maior-equipe` que, dadas 2 equipes, devolve a maior equipe. Indique o critério usado no caso de equipes com números de integrantes iguais.

Exercício 3. Faça uma função chamada `nome-maior-equipe` que, dadas 2 equipes, devolve o nome da maior equipe. Indique o critério usado no caso de empate.

Exercício 4. Qual a diferença entre as funções das questões 2 e 3? Dadas 2 equipes, elas produzem o mesmo resultado? Explique.

Exercício 5. Quando uma equipe ganha uma tarefa, sua pontuação deve ser atualizada. Construa uma função chamada `aumenta-pontuacao` que, dados uma equipe e um número de pontos, atualiza o registro da equipe aumentando sua pontuação e acordo com o número de pontos dado.

Obs.: Quando é pedido para “atualizar” o registro, espera-se que a função retorne uma instância de equipe completa.

Exercício 6. Defina o tipo `Confronto`, que deve conter uma identificação do confronto (número), 2 equipes, e um número de pontos que o vencedor ganhará. Dê 4 exemplos de elementos do conjunto `Confronto`.

Obs.: Quando falamos do conjunto gerado por uma definição de dados, usamos letra maiúscula. Porém, quando definimos a estrutura de fato, utilizamos letra minúscula. Logo, o nome do tipo a ser criado deve ser `confronto`.

Exercício 7. Faça uma função chamada `emblema-maior-equipe` que, dados 2 confrontos, devolve o emblema do time com maior pontuação envolvido nestes confrontos.

Exercício 8. Invente uma forma de visualizar os dados de uma equipe. Construa uma função chamada `mostra-equipe` que, dada uma equipe, gera uma imagem mostrando os dados desta equipe.