

Trabalho Prático II – Programação Funcional

REGRAS

Este trabalho deve ser postado no Moodle como um arquivo compactado (.zip) contendo o arquivo fonte do programa. O nome do arquivo enviado deve ser formado pelo nome do aluno sem espaço e acentuação. Por exemplo, João Souza fez o trabalho, então o nome do arquivo é JoaoSouza.zip.

É proibido usar as funções prontas fornecidas pela linguagem, exceto: as funções `length`, `sum`, `product` e as funções apresentadas nos Slides 2, 7, 8, 10, 11 e 13 da disciplina.

Somente os operadores fornecidos pela biblioteca padrão (Prelude) podem ser usados.

Faz parte do trabalho implementar funções auxiliares próprias. Além disso, a implementação de uma função pode requerer do uso de uma função implementada em uma questão anterior. Identificar essas relações entre as questões e aplicá-las fazem parte da avaliação.

O uso de sites e ferramentas (como Google, Stack Overflow, ChatGPT, etc.) para auxiliar no desenvolvimento do programa proposto no trabalho não é proibido. Contudo, faz parte da avaliação entender e explicar o código fornecido por essas ferramentas.

Questão 01 [1,0]. Crie os módulos (arquivos) *Util*, *Dados*, *Alunos*, *Livros* e *Emprestimo* dentro do módulo (diretório) *Biblioteca*. No módulo *Util*, crie o tipo algébrico *Data*, com dia, mês e ano (Int) e a função *dataStr* que recebe uma data e retorna uma String no formato dd/mm/aaaa. Também é no módulo *Util* que devem ser implementadas as funções auxiliares que venham a ser utilizadas por outras funções. No módulo *Dados*, crie a classe de tipos *Dado*. Nos demais módulos, crie, respectivamente, os tipos algébricos: *Aluno*, com código (Int) e nome (String) e email (String); *Livro*, com registro (Int), título (String) e edição (Int); e *Empréstimo*, com número (Int), aluno, data de empréstimo, data de devolução e livros (lista normal do tipo Livro).

Questão 02 [1,0]. Crie no módulo *Dados* um Tipo Abstrato de Dados (TAD) Set para armazenar valores de tipos pertencentes a classe *Dado*. Set (conjunto) é um TAD semelhante a uma lista, porém não permite valores repetidos. Ele deve ter funções que permitem inserir, remover e buscar um dado e verificar se está vazio.

Questão 03 [1,0]. Crie no módulo *Util* uma função chamada *formata* que recebe duas Strings (chave e valor) e retorna uma String formatada com 30 caracteres para a chave (completados com '.' e ':' alinhado à esquerda) e 50 para o valor (completados com espaço alinhado à direita). Ex.:

```
> formata "Nome" "João"
"Nome.....:                               Joao"
> formata "Data de Devolução" "05/06/2017"
"Data de Devolução.....:                   05/06/2017"
```

Questão 04 [1,0]. Insira na classe *Dado* a definição da função *imprimir* e implemente uma versão para *Aluno*, *Livro* e *Empréstimo* nos respectivos módulos. Essa função recebe um valor de um tipo pertencente à classe *Dado* e imprime na tela todos os dados desse valor. Repare que a função já faz a impressão na tela ao invés de retornar um String. A impressão dos atributos e valores dos dados devem estar formatados com base na função *formata*.

Questão 05 [1,0]. Insira na classe *Dado* a definição da função *cadastrar* e implemente uma versão para *Aluno*, *Livro* e *Empréstimo* nos respectivos módulos. Essa função solicita do usuário os dados do respectivo tipo e grava em um arquivo (cada tipo tem seu próprio arquivo).

Questão 06 [1,0]. Insira na classe *Dado* a definição da função *obter* e implemente uma versão para *Aluno*, *Livro* e *Empréstimo* nos respectivos módulos. Essa função retorna um TAD Set (Conjunto) contendo, respectivamente, todos os alunos, livros ou empréstimos do arquivo.

Questão 07 [1,0]. Insira na classe *Dado* a definição da função *buscar* e implemente uma versão para *Aluno*, *Livro* e *Empréstimo* nos respectivos módulos. Essa função recebe um inteiro, que corresponde ao código, registro ou número, e retorna o dado correspondente a partir da TAD Set carregado do arquivo. Note que a função pode não encontrar o dado com o inteiro informado e retornar *Nothing*.

Questão 08 [1,0]. Insira na classe *Dado* a definição da função *apagar* e implemente uma versão para *Aluno*, *Livro* e *Empréstimo* nos respectivos módulos. Essa função recebe um inteiro, que corresponde ao código, registro ou número, remove dado correspondente do TAD Set carregado do arquivo e atualiza esse arquivo. Um aluno ou livro não pode ser apagado se houver um empréstimo cadastrado com ele.

Questão 09 [1,0]. Insira na classe *Dado* a definição da função *showmenu* e implemente uma versão para *Aluno*, *Livro* e *Empréstimo* nos respectivos módulos. Essa função imprime na tela um menu com as opções voltar, visualizar, cadastrar e apagar e solicita que o usuário digite uma opção, retornando o valor digitado.

Questão 10 [1,0]. Crie o módulo *Main* e implemente nele a função *main*, em que o usuário pode escolher entre sair ou entrar nos menus de aluno, livro e empréstimo. Em cada um desses menus ele pode voltar ao menu principal, visualizar, cadastrar ou apagar os dados de um aluno, livro ou empréstimo.