

Trabalho de Programação Funcional em Haskell

Tema livre, a ser escolhido pelo aluno

O aluno deve escolher um tema que lhe interesse para desenvolver um projeto em Haskell, aplicando os conceitos de programação funcional indicados abaixo.

Objetivo:

Desenvolver um projeto em Haskell que explore os seguintes tópicos:

- **Listas em Haskell:** manipulação e processamento de listas usando funções próprias e funções de ordem superior;
- **Classes de Tipo:** definição de classes de tipo e suas instâncias para modelagem polimórfica;
- **Tipos Algébricos:** uso de tipos algébricos (sum types, product types, tipos compostos) para modelar dados do problema;
- **Abstract Data Types (ADT):** encapsulamento da implementação interna do tipo abstrato e fornecimento de uma interface controlada;
- **IO e Construção de Módulos:** construção modular do projeto, separando funcionalidades em módulos, e uso da monada IO para interação com o usuário e/ou leitura e escrita de arquivos.

Requisitos:

1. Definição do Tema e Justificativa:

- O aluno deve escolher um tema de seu interesse (exemplo: biblioteca de livros, gerenciador de tarefas, sistema bancário simples, jogo de texto, controle de estoque, etc).
- Deve explicar por escrito o problema que pretende resolver e a motivação para a escolha.

2. Modelagem de Dados com Tipos Algébricos:

- Definir tipos algébricos para representar os dados do domínio escolhido.
- Usar somas (data com múltiplos construtores) e produtos (registros) adequadamente.

3. Implementação de Funções com Listas:

- Usar listas para armazenar coleções dos dados.
- Implementar funções para adicionar, remover, buscar e listar elementos.
- Usar compreensão de listas e funções de ordem superior.

4. Classes de Tipo:

- Criar ao menos uma classe de tipo personalizada que capture uma abstração do domínio.
- Implementar instâncias para tipos relevantes do projeto.

5. Abstract Data Type (ADT):

- Encapsular a estrutura principal de dados (ex: uma agenda, uma coleção, uma base de dados simples) como um tipo abstrato.
- Fornecer funções públicas para manipulação, escondendo a estrutura interna.

6. Modularização e IO:

- Organizar o código em módulos coerentes.
- Implementar funcionalidades de entrada e saída para interação com o usuário (via terminal) ou persistência em arquivo.

7. Documentação:

- Comentar o código explicando o uso dos conceitos da programação funcional.
- Incluir um pequeno relatório explicando as escolhas de design, estrutura do projeto e exemplos de uso.

Sugestão de temas (livres):

- Sistema de gerência de biblioteca (livros, usuários, empréstimos).
- Gerenciador de tarefas e compromissos.
- Jogo de texto simples (aventura, quiz).
- Catálogo de filmes ou séries (locadora).
- Controle de estoque para uma pequena loja.
- Simulação de conta bancária.

- Aplicação para organização de receitas culinárias.

Entrega:

1 – Documentação PDF – 40%

2 – Apresentação da implementação + Código. (Meet) 60%

Grupo:

ATÉ 6 PESSOAS (MAXIMO) (NÃO VAI PASSAR DISSO) (NÃO ADIANTA PEDIR)