

Disciplina: Paradigmas de Programação
Professor: Maicon Rafael Zatteli
Entrega: Moodle

Atividade VIII - Haskell

Atenção: Faça um ZIP com todos os arquivos de solução. Use o nome do arquivo de maneira a entender qual problema você está resolvendo. Por exemplo, problema1.zip, problema2.zip e assim por diante, visto que cada problema pode ter mais de um arquivo.

Resolva os seguintes problemas na linguagem Haskell:

1. Modifique o módulo de Formas visto em sala de aula e disponível nos slides para que suporte o cálculo de área de um trapézio. Portanto, crie uma nova forma para o Trapézio, exporte ela e faça um teste no módulo Main.
2. Crie um módulo chamado Ponto, o qual deve definir o tipo Ponto, que é um Ponto2D formado por uma tupla (Float, Float), ou seja, `data Ponto = Ponto2D (Float, Float)`. Crie um módulo Main para testar o módulo Ponto. Neste módulo crie as seguintes funções:
 - `distancia :: Ponto -> Ponto -> Float`, a qual deve retornar a distância entre dois pontos.
 - `colineares :: Ponto -> Ponto -> Ponto -> Bool`, a qual deve verificar se os três pontos são colineares. Os três pontos são colineares se o determinante referente a esses pontos é zero.
 - `formaTriangulo :: Ponto -> Ponto -> Ponto -> Bool`, a qual deve verificar se os três pontos podem formar um triângulo.
3. Crie um módulo para as funcionalidades de árvores vistas em aula e resolvidas nos exercícios de listas anteriores. Faça também um módulo Main para exemplificar o uso do módulo da árvore.
4. Explore os arquivos `principal.hs` e `pilha.hs` disponíveis no Moodle. Entenda seu funcionamento, tente compilar e executar em seu computador. Teste também outros exemplos. Depois faça:
 - Crie um módulo para a estrutura de dados Fila.
 - Crie a estrutura para a Fila (Queue).
 - Crie a operação para enfileirar algum elemento (`enqueue`).
 - Crie a operação para desenfileirar algum elemento (`dequeue`).
 - Crie a operação para retornar o primeiro elemento da fila (`first`).
 - Crie a operação para criar uma fila vazia (`emptyQueue`).
 - Crie um módulo Main que ilustre o uso de todas as funcionalidades implementadas.
 - Inclua pequenos comentários em seu código.
5. Pesquise sobre definição de precedências no Haskell (`fixity`). Faça um breve resumo e um pequeno exemplo ilustrativo. Utilize módulos em seu exemplo. SUGESTÃO: utilize os operadores `***` e `+++` e classes já criados em aulas anteriores. Entregue um `.zip` com seus arquivos `.hs` de exemplo.