Exercício sobre Exceções

Relembre que exceções existem para representar situações anormais de funcionamento do sistema. Tais situações podem estar relacionadas a diversas naturezas: negócio, persistencia, comunicação, etc.

O mecanismo de exceções permite lidar com isso através da representação de erros como “objetos” que podem ser retornados quando ocorre uma situação anormal de funcionamento.

As exceções devem guardar alguma informação que permita a recuperação do sistema, pois um software deve ir de um estado consistente para outro consistente, independente da ocorrencia de erros/exceções.

Relembre o exercício de implementação de Bag (multiconjunto) usando Data.List e Data.Map como estruturas sobrejacentes.

As seguintes situações anormais de funcionamento foram definidas para essas implementações:

- remove - não é possivel remover um element que nao existe no bag. Neste caso, deve-se retornar uma exceção chamada ElementoInexistente
- minus – na subtração de bags, caso um elemento x esteja no primeiro bag com uma quantidade menor que no bag a ser subtraído, então deve retornar uma excecao OperacaoNaoPermitida. Exemplo: {(a,3)} \ {(a,4)} retornaria essa exceção porque o primeiro Bag nao possui quantidade suficiente de ‘a’ para subtrair.
- Embora as excecoes acima sejam tipificadas, é sempre bom ter uma mensagem de texto associada a elas. Considere isso na hora que for implementa-las.

Exercício sobre Testes

Elabore casos de teste para as implementacoes de Multiset (Bag). Note que os testes podem ser unicos, pois as assinaturas das funcoes são iguais.

Tente pensar nas diversas situações de uso da estrutura: quanto ela tem que funcionar com sucesso e quando tem que haver um erro. Para saber se o funcionamento está correto é bom também ter um “snapshot” da estrutura ou ver propriedades internas.