

**1A** Classes primitivas em Haskell são as que não herdam de nenhuma e servem de base para construção de outras. As secundárias são compostas pelo restante que herda de alguma outra classe. São 3 exemplos de classes primitivas: Bounded, Enum, Eq.

Bounded é uma classe que estabelece limites mínimos e máximos e que instancia os tipos Int, Char, Bool, (), Ordering e tuples. Como exemplo, ela possui o método maxBound delimita um máximo.

Enum define operações sobre tipos ordenados sequencialmente. Alguns tipos como Int, Float, char são instâncias de Enum. Ela possui a função prev, por exemplo. Ela retorna o elemento anterior a um valor.

A classe Eq providencia os operadores de igualdade (==) e desigualdade(=/=). Todos os tipos básicos são instâncias desta classe, exceto funções e IO. Um de seus métodos é (==), que verifica igualdade entre dois tipos.

Classes Secundárias:

Num é uma classe básica numérica que define a soma, multiplicação etc. Ela implementa o método fromIntegral que permite conversão de tipos.

Ord define métodos para tipos de dados ordenados, instanciando todos os tipos, exceto funções, IO e IOError. Ela possui entre seus métodos o método compare que compara a igualdade e diferença (maior que, menor que) entre elementos.

**B** Classes numéricas compõem grande parte das classes padrões do haskel. A classe Real é um exemplo por estender Num, que define as operações básicas como soma e subtração.