

Polimorfismo paramétrico e ad hoc

- Atividade 1
- Atividade 2
- Atividade 3
- Atividade 4
- Atividade 5
- Atividade 6

Atividade 1

- Defina uma função que retorna o comprimento de uma lista. Considere uma lista com quaisquer tipos de valores.
- Observação: Os tipos podem ser Integer, Float, Bool, Char, String, etc.
- Para essa atividade, use recursividade.

Atividade 2

- Defina uma função polimórfica que soma todos os elementos de uma lista. Considere uma lista de somente valores numéricos.
- Observação: Os valores numéricos podem ser dos tipos Int, Integer, Float, Double, etc.
- Para essa atividade, use recursividade.

Atividade 3

- Defina uma função polimórfica que determina a lista cujo os elementos são o dobro dos elementos de uma lista de entrada. Considere que as listas sejam numéricas.
- Para essa atividade, use recursividade para definir a função polimórfica

Atividade 4

- Elabore uma função polimórfica que retorna uma lista cujo os elementos seja a soma dos pares de uma lista de tuplas.
- Considere que a tupla seja de quaisquer valores numéricos. Declare o tipo da função.
- Use **compreensão de listas** para definir a função

Atividade 5

- Defina uma função polimórfica com a finalidade de criar uma lista de tuplas a partir dos elementos de duas listas iniciais.
- No Prelude já tem uma função para isso (A função zip). Não é para usar essa função na solução.
- **Use recursividade sem casamento de padrão**

Atividade 6

- Defina uma função polimórfica com a finalidade de criar uma lista de tuplas a partir dos elementos de duas listas iniciais.
- No Prelude já tem uma função para isso (A função zip). Não é para usar essa função na solução.
- **Use recursividade com casamento de padrão**