RELATÓRIO: HIERARQUIA DE CLASSES

Curso: Tópicos em Ciência de Dados A (MAE013)

Professor: Milton R Ramirez

Integrantes

Leonardo Domingos (DRE: 120168324);

Calculadora de Matrizes

1. Introdução

Nesse projeto, iremos implementar uma classe Pilha, com o intuito de usá-la para resolver o problema da Torre de Hanoi da forma mais otimizada possível (2*número_discos - 1). Esse problema tem as seguintes regras:

- **Apenas um disco pode ser movido por vez.**
- **Um disco maior nunca pode ser colocado sobre um menor.**
- **Apenas o disco do topo de uma torre pode ser movido.**

O usuário poderá dar como input a quantidade de discos que a torre terá, e o programa irá comunicar a quantidade de movimentos necessários para resolver o problema, seguindo as regras acima.

2. Implementação

A solução foi construída com base em dois blocos principais: a definição de uma classe **Pilha** para representar cada torre, e a função recursiva **torre_de_hanoi** que executa a lógica de movimentação dos discos.

Vale notar que o uso da Pilha enquanto estrutura de dados para resolver o problema é perfeito, uma vez que as pilhas da torre de hanoi literalmente seguem todas as regras da Pilha da classe de programação.

3. Comentários sobre o Desenvolvimento

Durante o desenvolvimento, primeiramente tentou-se resolver o problema de forma simplificada, sem usar a Stack como estrutura de dados (só usando listas). Isso foi feito com o intuito de desenvolver primeiro a torre_de_hanoi. Contudo, essa metodologia se provou mais difícil do que a forma “correta”, a de implementar a função primeiro. Tratar cada disco e cada pilha como números valores de uma lista se provou desafiador, e a cada tentativa de melhorar o código, percebeu-se que estávamos somente lentamente implementando a funcionalidade de uma Pilha no nosso programa. Por fim, decidiu-se começar do zero, dessa vez fazendo o caminho reverso.

Outra dificuldade foi de fato implementar a função torre_de_hanoi. Principalmente pois a função recursiva era chamada infinitamente, ou não era chamada suficientemente. Foi necessário pesquisar sobre boas práticas de funções recursivas, e viu-se que a forma “correta” é:

- Primeiro implementar quando você deseja que a função pare de rodar (no caso, n=1)
- Depois implementar como você deseja que a função “caminhe” (implementado no else)

Dividir a função assim não só facilitou o desenvolvimento, como também melhorou a legibilidade do código.