**Documentação: Manipulação de Árvore Binária em Java  
  
Integrantes: Diego Magno Lopes Martins; Jefferson Henrique Machado; João Vitor Oliveira da Silva  
Link repositório git: https://github.com/jeffersonmachado95/arvoreBinaria**

# 1. Introdução

Este trabalho tem como objetivo a implementação de uma estrutura de dados baseada em árvore binária, utilizando a linguagem de programação Java. A proposta consiste em permitir operações como inserção, remoção e busca de elementos em uma árvore, além da verificação de características estruturais, como se a árvore é estritamente binária, cheia ou completa. Também foi incluída a possibilidade de impressão da árvore em diferentes ordens (in-ordem, pré-ordem e pós-ordem), cálculo do nível da árvore e do grau de um nó específico.

# 2. Implementação

## Estrutura de Dados

Foi utilizada uma estrutura de árvore binária, composta por nós. Cada nó possui um valor inteiro e duas referências: uma para o nó à esquerda e outra para o nó à direita. A estrutura é composta por duas classes principais:  
- Node: representa cada nó da árvore.  
- BinaryTree: contém os métodos responsáveis por manipular a árvore.

## Principais Funções Implementadas

- insert(int key): insere um novo valor na árvore, respeitando as regras da árvore binária de busca.  
- remove(int key): remove um valor da árvore e reorganiza os nós conforme necessário.  
- search(int key): verifica se um determinado valor existe na árvore.  
- printInOrder(): imprime os elementos da árvore em ordem (esquerda, raiz, direita).  
- printPreOrder(): imprime os elementos na ordem pré-fixada (raiz, esquerda, direita).  
- printPostOrder(): imprime os elementos na ordem pós-fixada (esquerda, direita, raiz).  
- isStrictlyBinary(): retorna verdadeiro se todos os nós possuem 0 ou 2 filhos.  
- isFull(): verifica se a árvore é cheia (todos os nós possuem 0 ou 2 filhos e todos os níveis estão completos).  
- isComplete(): verifica se a árvore é completa (todos os níveis, exceto o último, estão completamente preenchidos, e os nós do último nível estão o mais à esquerda possível).  
- getHeight(): retorna o nível (altura) da árvore.  
- getDegree(int value): retorna o grau (quantidade de filhos) de um nó especificado.

## Formato de Entrada e Saída

As entradas são números inteiros inseridos diretamente no método main. As saídas são exibidas no terminal por meio do comando System.out.println, com os resultados das operações e verificações realizadas.

## Decisões de Implementação

Para garantir a clareza e modularidade do código, foi seguido o paradigma da programação orientada a objetos. A remoção de elementos foi implementada considerando os três casos possíveis:  
1. Nó sem filhos.  
2. Nó com um único filho.  
3. Nó com dois filhos (utiliza-se o sucessor em ordem).

# 3. Listagem de Testes Executados

Foram realizados os seguintes testes práticos, com os valores inseridos: 50, 30, 70, 20, 40, 60, 80.  
  
Testes:  
- Impressão em ordem, pré-ordem e pós-ordem.  
- Remoção do valor 20.  
- Busca pelos valores 60 (presente) e 90 (ausente).  
- Verificação do grau do nó 30 (esperado: 2).  
- Verificação se a árvore é:  
- Estritamente binária (esperado: verdadeiro).  
- Cheia (esperado: verdadeiro).  
- Completa (esperado: verdadeiro).  
- Altura da árvore (esperado: 2).  
  
Todos os testes retornaram os valores esperados, confirmando o correto funcionamento da árvore.

# 4. Conclusão

O trabalho permitiu compreender melhor o funcionamento das árvores binárias, especialmente no que se refere à manipulação dinâmica de dados. As principais dificuldades encontradas foram relacionadas à lógica de remoção de nós e à verificação das condições estruturais da árvore. No geral, a implementação foi bem-sucedida, e todas as funcionalidades propostas foram contempladas.

# 5. Bibliografia

- Jovana, S. (2019). Referências bibliográficas da ABNT: qual é o padrão e como fazer a referência bibliográfica em um artigo? Acessado em 04/04/2025.  
- Team, O. (2025). Overleaf. Acessado em 04/04/2025.  
- Oracle Java Documentation: https://docs.oracle.com/javase/8/docs/  
- GeeksforGeeks. Binary Tree Data Structure: https://www.geeksforgeeks.org/binary-tree-data-structure/