Universidade Presbiteriana Mackenzie

ESTRUTURA DE DADOS II

Árvore Binária
Atividade (máx. três alunos)

Objetivo

Implementar uma árvore binária em Java e testar a sua implementação.

Instruções

- A atividade deve ser resolvida usando a linguagem Java.
- A solução não deve usar as estruturas de dados oferecidas pela linguagem Java (projetos que usarem tais estruturas serão desconsiderados – zero).

Enunciado

1. Crie uma classe Java que define um novo tipo de dado usado para representar os atributos e operações de um nó usado pela árvore binária (ex. Node). Os atributos e operações da classe são:

ATRIBUTO	DESCRIÇÃO
data	Nesta atividade, o nó armazena uma String como dado.
parent	Referência para o nó pai.
left	Referência para o nó filho da esquerda.
right	Referência para o nó filho da direita.

OPERAÇÃO	DESCRIÇÃO
Construtor(es)	Construtor(es) da classe.
get*()	*Getters* dos atributos do nó.
set*()	*Setters* dos atributos do nó.
isRoot()	Verifica se o nó é raiz (true se nó é raiz, false caso contrário).
isLeaf()	Verifica se o nó é folha (true se nó é folha, false caso contrário).
getDegree()	Retorna o grau do nó (int).
getLevel()	Retorna o nível do nó (int).
getHeight()	Retorna a altura do nó (int).

ESTRUTURA DE DADOS II

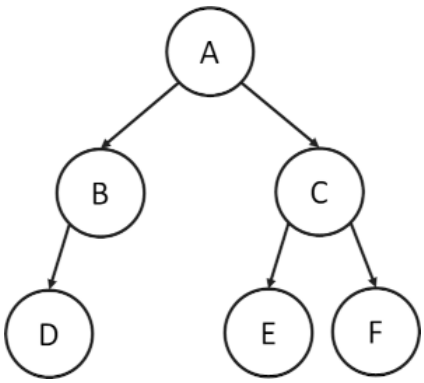
2. Crie uma classe Java que define um novo tipo de dado usado para representar os atributos e operações da árvore binária (ex. BinaryTree). Os atributos e operações da classe são:

ATRIBUTO	DESCRIÇÃO
root	Referência para a raiz da árvore.

OPERAÇÃO	DESCRIÇÃO
Construtor(es)	Construtor(es) da classe.
get*()	*Getters* dos atributos da árvore.
set*()	*Setters* dos atributos da árvore.
isEmpty()	Verifica se a árvore está vazia (true se a árvore está vazia, false caso contrário).
getDegree()	Retorna o grau do da árvore (int).
getHeight()	Retorna a altura da árvore (int).
inOrderTraversal()	Percorre a árvore em ordem.
preOrderTraversal()	Percorre a árvore em pré-ordem.
postOrderTraversal()	Percorre a árvore em pós-ordem.
levelOrderTraversal()	(Bônus opcional) Percorre a árvore por nível.

"Percorrer a árvore", nesta atividade, significa visitar os nós da árvore e exibir o conteúdo de cada nó na saída padrão do sistema (ex. terminal).

3. Para testar o seu código, construa a seguinte árvore na memória e use todos os métodos implementados para validação.



4. A sua main() deve ter código que exiba todas as informações de cada nó (se é raiz, se é folha, grau, nível e altura) e todas as informações da árvore (se está vazia, grau e altura da árvore, percurso em ordem, percurso em pré-ordem e percurso em pós-ordem).