



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 usem tais estruturas serão desconsiderados – zero).
- Inclua a identificação do grupo (nome completo e RA de cada integrante) no início de cada arquivo de código, como comentário.
- Inclua todas as referências (livros, artigos, sites, vídeos, entre outros) consultadas para solucionar a atividade como comentário no arquivo `.java` que contém a `main()`.

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
<code>data</code>	Nesta atividade, o nó armazena uma <code>String</code> como dado.
<code>parent</code>	Referência para o nó pai.
<code>left</code>	Referência para o nó filho da esquerda.
<code>right</code>	Referência para o nó filho da direita.

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

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
<code>root</code>	Referência para a raiz da árvore.

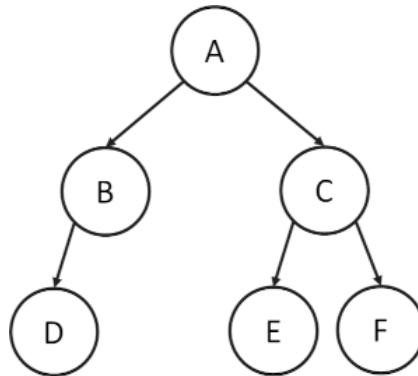


ESTRUTURA DE DADOS II

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

Entrega

Compacte o código-fonte (somente arquivos `*.java`) no formato `zip`.

Atenção: O arquivo `zip` não deve conter arquivos intermediários e/ou pastas geradas pelo compilador/IDE (ex. arquivos `*.class`, etc.).

Prazo de entrega: via link do Moodle até 19/03/2024 23:59.



ESTRUTURA DE DADOS II

Cr terios de avalia  o

A nota da atividade   calculada de acordo com os crit rios da tabela a seguir.

ITEM AVALIADO	PONTUA��O M�XIMA
1. Implementa��o b�sica da classe que representa um n� usado pela �rvore bin�ria (atributos, construtor(es), <i>getters/setters</i>).	0,5
1. Implementa��o da opera��o <i>isRoot()</i> .	0,25
1. Implementa��o da opera��o <i>isLeaf()</i> .	0,25
1. Implementa��o da opera��o <i>getDegree()</i> .	0,25
1. Implementa��o da opera��o <i>getDepth()</i> .	1,0
1. Implementa��o da opera��o <i>getHeight()</i> .	1,0
2. Implementa��o b�sica da classe que representa uma �rvore bin�ria (atributos, construtor(es), <i>getters/setters</i>).	0,5
2. Implementa��o da opera��o <i>isEmpty()</i> .	0,25
2. Implementa��o da opera��o <i>getDegree()</i> .	0,5
2. Implementa��o da opera��o <i>getHeight()</i> .	0,5
2. Implementa��o da opera��o <i>inOrderTraversal()</i> .	1,0
2. Implementa��o da opera��o <i>preOrderTraversal()</i> .	1,0
2. Implementa��o da opera��o <i>postOrderTraversal()</i> .	1,0
2. Implementa��o da opera��o <i>levelOrderTraversal()</i> (b�nus opcional).	1,0
3. �rvore bin�ria de teste.	1,0
4. Funcionamento geral do programa, de acordo com o enunciado.	1,0

Tabela 1 - Crit rios de avalia  o.

A tabela a seguir cont m crit rios de avalia  o que podem **reduzir** a nota final da atividade.

ITEM INDESEJ�VEL	REDU��O DE NOTA
O projeto � c�pia de outro projeto.	Projeto � zerado
O projeto usa estruturas de dados oferecida pela linguagem Java.	Projeto � zerado
H� erros de compila��o e/ou o programa trava durante a execu��o ¹ .	-1,0
N�o h� identifica��o do grupo. N�o h� indica��o de refer�ncias. Arquivos enviados em formatos incorretos. Arquivos e/ou pastas intermedi�rias que s�o criadas no processo de compila��o ou pela IDE foram enviadas junto com o c�digo-fonte.	-1,0

Tabela 2 - Crit rios de avalia  o (redu  o de nota).

O c digo-fonte ser  compilado com o compilador *javac* (21.0.2) na plataforma Windows da seguinte forma:

```
> javac *.java -encoding utf8
```

O c digo compilado ser  executado com *java* (21.0.2) na plataforma Windows da seguinte forma:

```
> java <Classe>
```

Sendo que *<Classe>* deve ser substituído pelo nome da classe que cont m o m todo *public static void main(String[] args)*.

¹ Sobre erros de compila  o: considere apenas erros. N o h  problema se o projeto tiver *warnings* (embora *warnings* podem avisar sobre poss veis travamentos em tempo de execu  o, como loop infinito, divis o por zero, etc.).