UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

IMD0040 – LINGUAGENS DE PROGRAMAÇÃO 2

PROJETO 2 – UNIDADE 1 – 2020.2

Instruções:

- O trabalho é individual.
- Deve ser entregue até o final do dia 23/02/2020.
- Deve ser submetido o código fonte de cada questão.
- Cada questão deve vir acompanhada por um vídeo explicando o que foi desenvolvido.
- O código fonte e os vídeos devem ter no máximo 50mb.
- Cada questão vale 2 pontos de um total de 10. A nota desse trabalho corresponde a 50% da nota da primeira unidade.

Questão 1:

Adicione o método equals na classe Tree fornecida. O método equals deve receber uma árvore e retornar verdadeiro caso as duas árvores sejam iguais, isto é, o nó raiz da árvore A armazena o mesmo valor que o nó raiz da árvore B e a subárvore a esquerda/direita da raiz de A é igual a subárvore da raiz a esquerda/direita de B. A classe Tree deve ser movida para o pacote "alunoX.tree" onde X deve ser substituído pela matrícula do aluno.

Restrição: a implementação deve ser recursiva e não deve fazer uso de estruturas de repetição como for, while, do, etc.

Vídeo: deve explicar o código e exibir o resultado da execução comparando duas árvores vazias. Deve exibir a compilação e execução do código via linha de comando.

Questão 2:

Implemente uma nova classe TreeV1 que herda a classe Tree fornecida, isto é, TreeV1 é uma subclasse de Tree. A classe TreeV1 deve ter um método de inserção de valores inteiros satisfazendo a regra de árvores binárias de busca.

Restrição: a implementação deve ser iterativa sem chamadas recursivas.

Vídeo: deve explicar o código e exibir e um exemplo. O exemplo consiste em instanciar uma árvore da classe Tree e outra da classe TreeV1, escolha 20 valores diferentes e os

insira na mesma ordem em cada árvore, por exemplo, para os elementos 1, 2,3, será instanciada uma árvore da classe Tree e inseridos os elementos 1, 2, 3 e depois será instanciada uma árvore da classe TreeV1 e inseridos os elementos 1, 2, 3. Mostre o resultado da comparação do objeto da classe TreeV1 recebendo o objeto da classe Tree.

Questão 3:

Implemente uma nova classe TreeV2 que herda a classe Tree. A classe TreeV2 deve ter um método de inserção de valores inteiros satisfazendo a regra de árvores binárias de busca. A implementação deve ser recursiva onde a função recursiva deve ter a seguinte assinatura:

private Node insert(Node node, int value)

O método de inserção deve inserir um novo nó na árvore contendo o valor especificado pelo argumento "value" e deve retornar a subárvore modificada pela inserção.

Restrição: a implementação deve ser recursiva e não deve fazer uso de estruturas de repetição como for, while, do, etc. Toda a lógica de inserção deve ser efetuada pelo método insert.

Vídeo: deve explicar o código e exibir e um exemplo. O exemplo consiste em instanciar uma árvore da classe Tree e outra da classe TreeV2, escolha 20 valores diferentes e os insira na mesma ordem em cada árvore, por exemplo, para os elementos 1, 2,3, será instanciada uma árvore da classe Tree e inseridos os elementos 1, 2, 3 e depois será instanciada uma árvore da classe TreeV2 e inseridos os elementos 1, 2, 3. Mostre o resultado da comparação do objeto da classe TreeV2 recebendo o objeto da classe Tree.

Questão 4:

Implemente um método protegido na classe Tree que remova o sucessor de um valor indicado. O método deve ser implementado recursivamente com a seguinte assinatura:

protected Node removeSuccessor(Node node, Node parent, int value)

O método recebe como argumentos a raiz (node) da árvore sendo explorada, o nó (parent) parente da raiz e o valor (value) do qual deve ser removido o sucessor. O método deve retornar o nó correspondente ao sucessor do valor armazenado em value.

Restrição: a implementação deve ser recursiva e não deve fazer uso de estruturas de repetição como for, while, do, etc. Toda a lógica de remoção deve ser efetuada pelo método removeSuccessor.

Vídeo: deve explicar o código e explicar um exemplo removendo um elemento de uma árvore com ao menos 8 elementos. Deve ficar clara cada chamada recursiva.

Questão 5:

Crie um novo pacote de nome "alunoX.rem" e adicione cópias das classes Tree e Node fornecidas. Modifique a classe Node para ter uma referência para o nó parente considerando a estrutura da árvore. Crie uma nova classe TreeV3 que herda a classe Tree. A classe TreeV3 deve ter métodos de inserção e remoção de elemento satisfazendo a regra de árvore binária de busca. O método de inserção deve atribuir corretamente o nó parente de cada nó durante a inserção. O método recursivo de remoção deve ter a seguinte assinatura:

private boolean remove(Node node, int value)

O método retorna verdadeiro caso o valor value tenha sido removido considerando a árvore representada pelo nó node.

Restrição: a implementação deve ser recursiva e não deve fazer uso de estruturas de repetição como for, while, do, etc. Toda a lógica de remoção deve ser efetuada pelo método remove.

Vídeo: deve explicar o código e explicar um exemplo removendo um elemento de uma árvore com ao menos 8 elementos. Deve ficar clara cada chamada recursiva.