FUNDAÇÃO UNIVERSIDADE REGIONAL DE BLUMENAU CENTRO DE CIÊNCIAS EXATAS E NATURAIS DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO Professor Marcel Hugo

Lista 9 de Exercícios 2023/1

Árvores n-árias

O objetivo desta atividade prática é realizar a implementação de árvores com número variável de filhos, de acordo com o diagrama de classes da figura abaixo.

Visual Paradigm for UML Community Edition [not for commercial Arvore <T> -raiz : NoArvore <T> +setRaiz(raiz : NoArvore<T>) +pertence(info : T) : NoArvore<T> +vazia(): boolean +toString(): String 0..1 0..1 NoArvore <T> info : T -filho : NoArvore <T> -irmao : NoArvore <T> irmao +NoArvore(Info: T) filho +imprimePre() : String +inserirFilho(filho: NoArvore<T>) 0..1 +pertence(info : T) : NoArvore<T> 0..1

Onde, para a classe Arvore:

- O construtor **Arvore** () deve inicializar a árvore sem nenhum nó;
- O método setRaiz (NoArvore) deve tornar o nó recebido como argumento como sendo a nova raiz da árvore;
- O método getRaiz () deve retornar a raiz da árvore;
- O método toString () deve retornar a representação textual da árvore. Para auxiliar a resolução do problema, utilize o método imprimePre() da classe NoArvore.
- O método pertence () deve procurar na árvore se há um nó que armazena um objeto cuja chave de busca seja idêntica à chave de busca do objeto fornecido como argumento. Seu retorno é o objeto cujo conteúdo seja a chave de busca ou null caso não seja localizado. Utilize o método pertence () da classe NoArvore para auxiliar na resolução deste problema.

Na classe NoArvore:

- O construtor deve inicializar um nó de árvore com o dado fornecido como argumento.
- O método inserirFilho () deve receber a raiz de uma subarvore e estabelecer que esta sub-arvore é filha do nó corrente.

Após implementar a árvore, crie uma classe contendo o método main () para testar e demonstrar o funcionamento da árvore implementada:

- crie uma árvore;
- crie nós e suas relações;
- imprima seu conteúdo;
- pesquise elementos que foram inseridos e também valores desconhecidos.