

Objectivos

- Consolidação de árvores de pesquisa binárias

Enunciado:

Na classe `BinarySearchTree` desenvolvida na ficha 4. Crie os seguintes métodos:

1. `boolean isLeaf(Object o)` que devolve `true` se o objecto passado por parâmetro se encontrar numa folha, ou `false` caso contrário. Se o objecto não existir na árvore o método retorna `false`.
2. `boolean isUnique(Object o)` que devolve `true` se o objecto passado por parâmetro for único na árvore, ou `false` caso contrário. Se o objecto não existir na árvore o método retorna `false`.
3. `Object brother(Object o)` que devolve o objecto que está no nodo irmão do nodo que possui o objecto `o`. Se o objecto `o` não existir ou se não tiver um irmão o método devolve `null`.
4. `int level(Object o)` que devolve o nível em que se encontra a 1ª ocorrência do objecto `o` na árvore. Assuma que a raiz da árvore se encontra no nível 1.
5. `int biggerLevel()` que devolve o nível da árvore com mais elementos. Note que a raiz da árvore é o nível 1.
6. `int numberOfElements()` que devolve o número de elementos da árvore.
7. `void showDecreasingOrder()` que mostra na consola os elementos da árvore ordenados por ordem decrescente.
8. `int numberOfLeaves()` que devolve o número de folhas que uma árvore tem.
9. `Object max()` que devolve o maior elemento de uma árvore.

10. `int numberOfOccurrences(Object value)` que devolve o numero de ocorrências do valor passado por parâmetro.
11. `int numberElementsInLevel(int level)` que devolve o número de elementos que existem num determinado nível. Note que a raiz da árvore é o nível 1.
12. `int numberElementsGratherThan(Object value)` que devolve o numero de elementos da árvore maiores do que o valor passado por parâmetro.
13. `int heigth()` que devolve a altura da árvore. Note que a altura da árvore é o nodo com maior nível.
14. `Tree union(Tree t1, Tree T2)` que devolve uma árvore de pesquisa binária com a união das duas árvores passadas por parâmetro.