

**Curso** : Engenharia de *Software*  
**Disciplina** : Algoritmos e Estruturas de Dados II  
**Professora** : Eveline Alonso Veloso

### Exercícios sobre Árvores Binárias de Busca

Os exercícios 1 a 8, abaixo, devem ser resolvidos na classe `ABB<K, V>` implements `IMapeamento<K, V>` implementada durante as aulas teóricas da disciplina.

- 1) Implemente o método `public String caminhamentoPreOrdem()`, capaz de percorrer a árvore binária de busca em pré-ordem, retornando uma *string* com os valores de seus elementos. Se a árvore binária de busca estiver vazia, uma exceção deve ser lançada.  
Sugestão: `V` deve sobrescrever a função `public String toString()`, de forma que a classe `ABB<K, V>` permaneça genérica.
- 2) Implemente o método `public String caminhamentoPosOrdem()`, capaz de percorrer a árvore binária de busca em pós-ordem, retornando uma *string* com os valores de seus elementos. Se a árvore binária de busca estiver vazia, uma exceção deve ser lançada.  
Sugestão: `V` deve sobrescrever a função `public String toString()`, de forma que a classe `ABB<K, V>` permaneça genérica.
- 3) Implemente o método `public String caminhamentoDecrescente()`, capaz de percorrer a árvore binária de busca retornando uma *string* com os valores de seus elementos, em ordem decrescente. Se a árvore binária de busca estiver vazia, uma exceção deve ser lançada.  
Sugestão: `V` deve sobrescrever a função `public String toString()`, de forma que a classe `ABB<K, V>` permaneça genérica.
- 4) Implemente o método `public V obterMenor()`, método responsável por recuperar e retornar o menor elemento armazenado na árvore binária de busca. Se a árvore binária de busca estiver vazia, uma exceção deve ser lançada.
- 5) Implemente a função `public ABB<K, V> clone()`, capaz de criar e retornar uma cópia exata da árvore binária de busca. A árvore binária de busca original e sua cópia, retornada por esse método, podem compartilhar os mesmos itens. No entanto, elas não devem compartilhar os mesmos nós, ou seja, cada uma das duas árvores binárias de busca deve apresentar seus próprios nós.  
Implemente também, na classe `No<K, V>` o método `public No<K, V> clone()`, capaz de criar e retornar uma cópia exata do nó em questão.  
Utilize o método `clone` do nó para implementar a função `clone` da árvore binária de busca.
- 6) Implemente a função `public ABB<K, V> obterSubconjuntoMajores(K chave)`, capaz de criar e retornar um subconjunto da árvore binária de busca formado apenas pelos elementos da `ABB` original cujas chaves são maiores ou iguais àquela passada como parâmetro para esse método.

Se não for encontrado, na árvore binária de busca, nenhum elemento cuja chave corresponde àquela passada como parâmetro para esse método, uma exceção deve ser lançada.

Sugestão: empregue os métodos `clone`, do nó e da árvore binária de busca, implementados anteriormente.

- 7) Implemente a função `public boolean ehRaiz(K chave)`, capaz verificar e retornar se o item, cuja chave foi informada como parâmetro para esse método, corresponde à raiz da árvore binária de busca.  
Se a árvore binária de busca estiver vazia, uma exceção deve ser lançada.
- 8) Implemente o método `public V obterAntecessor(K chave)`, capaz de recuperar e retornar o maior elemento armazenado na árvore binária de busca que seja menor do que a chave informada como parâmetro para esse método. Se não for encontrado, na árvore binária de busca, nenhum elemento cuja chave corresponde àquela passada como parâmetro para esse método; ou o item, cuja chave corresponde à informada como parâmetro, não apresentar antecessor na árvore binária de busca, uma exceção deve ser lançada.
- 9) Implemente o método `public Double calcularValorMedio(Function<V, Double> extrator)`, responsável por calcular e retornar a média de valores numéricos do tipo `Double` extraídos dos elementos armazenados na árvore binária de busca. Para isso, utilize a função `extrator` fornecida como parâmetro, que deverá ser aplicada a cada elemento da árvore binária de busca para obter os valores do tipo `Double` sobre os quais a média será calculada.
- 10) Implemente o método `public int contarSe(Predicate<V> condicional)`, capaz de verificar, contar e retornar quantos elementos armazenados na árvore binária de busca satisfazem uma determinada condição indicada pelo predicado `condicional`, passado como parâmetro para esse método.