

## 2. Implementação de balanceamento de Árvore Binária de Pesquisa

Escolher uma das árvores possíveis para serem estudadas são, obrigatoriamente: **Árvores AVL** ou **Árvores rubro-negras (Red-Black Tree)**.

### a) Estudar e entender o funcionamento da árvore de pesquisa escolhida

A primeira etapa consiste no estudo e entendimento dos algoritmos da árvore de pesquisa escolhida. O conteúdo pode ser encontrado nos livros indicados na bibliografia da disciplina, lembrando que o livro "GOODRICH, M. T.; TAMASSIA, R. Estruturas de dados e algoritmos em Java" está disponível *online* e aborda este conteúdo. Pesquisas na Web em sites confiáveis, tais como sites de universidades, podem ajudar no entendimento do conteúdo. Exemplos de sites:

### b) Implementar a árvore de pesquisa escolhida de acordo com a interface fornecida

Após o entendimento do funcionamento da árvore, é preciso implementá-la. A **implementação pode ser uma adaptação de uma implementação pronta, mas deve seguir o padrão trabalhado em aula**, isto é, deve ser implementada uma árvore binária de pesquisa balanceada para armazenar números inteiros. O TAD também deve seguir o padrão de TAD visto em aula para árvores binárias, sendo **obrigatório que a classe da árvore tenha os seguintes métodos**:

- Adicionar elementos na árvore: *void add(obj);*
- Retornar o pai de um elemento: *obj getParent(obj);*
- Limpar o conteúdo da árvore: *void clear();*
- Verificar se um elemento está armazenado na árvore ou não: *boolean contains(obj);*
- Verificar qual é a altura da árvore: *int height();*
- Verificar quantos elementos tem na árvore: *int size();*
- Verificar se a árvore está vazia ou não: *boolean isEmpty();*
- Retornar os elementos da árvore em uma lista usando diferentes caminhamentos (métodos *preOrder()*, *inOrder()*, *posOrder()*);
- Apresentar a visualização da árvore usando a classe *TreeFormatter*.

Para a implementação dos métodos acima será preciso incluir outros métodos necessários para o correto funcionamento da árvore e para que ela fique balanceada. Após a implementação de todos os métodos, é obrigatório analisar e incluir como comentário a notação O para os métodos *add*, *contains*, *height* e *clone*.

### c) Testar a árvore de pesquisa implementada de acordo com as instruções fornecidas

Além da implementação da classe da árvore, é preciso implementar um método *main* que instancie a árvore implementada e chame os seus métodos para verificar o seu correto funcionamento. Para isto, é obrigatório ter os seguintes comandos no método *main*:

- Instanciar a árvore implementada;
- Incluir os números 1, 2, 3, 4, 5, 6, 7, 8, 9, nesta ordem, na árvore;
- Apresentar a visualização da árvore;
- Apresentar a altura da árvore;
- Limpar o conteúdo da árvore;
- Incluir os números 9, 8, 7, 6, 5, 4, 3, 2, 1, nesta ordem, na árvore;

- Apresentar o conteúdo da árvore usando um caminhamento central;

**d) Relatório para explicar o funcionamento da árvore e os seus algoritmos**

Após terminar a implementação, cada dupla deverá elaborar um relatório para explicar o funcionamento da árvore estudada e os seus algoritmos de inserção e balanceamento. Este relatório deve conter no início, como título, o nome da árvore implementada, seguido dos nomes dos alunos. Deve ser usado espaçamento simples e fonte tamanho 12, sendo que o número de páginas deve ser entre 2 a 4, incluindo figuras para ilustrar o funcionamento dos algoritmos e referências para todos os sites, livros e artigos consultados.