



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Instituto de Ciências Exatas e Informática
Exercícios sobre Árvores Binárias de Busca

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

Considere um sistema de gestão de treinamentos. Nesse tipo de aplicação, a cada treinamento ministrado é criada uma turma e, para cada turma, é necessário armazenar os dados dos alunos nela matriculados e suas respectivas notas no treinamento. Considere ainda que a estrutura de dados escolhida para a implementação de uma turma foi uma Árvore Binária de Busca.

Sobre cada aluno matriculado na turma e que, portanto, receberá algum treinamento, são guardadas as seguintes informações: número de matrícula; nome; sua nota na turma.

Exercícios:

Todos os exercícios abaixo devem ser resolvidos na classe `ABBAlunos`, disponibilizada pela professora.

- 1) Implemente, em Java, o método `public Aluno menorNumeroMatricula()`: método responsável por retornar o aluno, armazenado na árvore, que apresenta o menor número de matrícula. O número de matrícula é a chave dos nós da árvore. Se a árvore estiver vazia, esse método retorna `null`.
- 2) Implemente, em Java, o método `private NodoAluno pesquisar(NodoAluno raizArvore, int matricula)`: método recursivo responsável por localizar um aluno na árvore. Esse método apresenta os parâmetros `raizArvore`, que corresponde à raiz da árvore ou sub-árvore que será pesquisada à procura do aluno; e `matricula`, que representa o número de matrícula do aluno que deverá ser localizado na árvore. Esse método retorna uma referência ao nó da árvore onde encontra-se o aluno pesquisado ou `null`, caso o aluno desejado não tenha sido localizado na árvore. Implemente também o método `public Aluno buscar (int matricula)`, que encapsula a pesquisa recursiva.
- 3) Implemente, em Java, o método `private void imprimirMenorMaior(NodoAluno raizArvore)`: método recursivo

responsável por imprimir os dados dos alunos armazenados na árvore, do menor número de matrícula para o maior, ou seja, em ordem. Esse método apresenta o parâmetro `raizArvore`, que corresponde à raiz da árvore ou sub-árvore da qual os dados dos alunos serão impressos. Implemente também o método `public void imprimirEmOrdem()`, que encapsula a impressão recursiva, em ordem, dos dados dos alunos armazenados na árvore.

- 4) Implemente, em Java, o método `private int contarNumAlunos(NodoAluno raizArvore)`: método recursivo responsável por contar e retornar a quantidade de nós da árvore. Esse método apresenta o parâmetro `raizArvore`, que corresponde à raiz da árvore ou sub-árvore da qual os nós serão contados. Implemente também o método `public int numAlunos()`, que encapsula a contagem recursiva da quantidade de nós da árvore.

Após a implementação dos exercícios 1 a 4, o método `main(String[] args)`, a seguir, deverá funcionar:

```
public static void main(String[] args) {

    Aluno alunoNovo, alunoPesquisado;
    ABBAAlunos turma = new ABBAAlunos();

    alunoNovo = new Aluno (5, "Amanda", 85.00);
    turma.inserir(alunoNovo);

    alunoNovo = new Aluno(8, "Uriel", 87.00);
    turma.inserir(alunoNovo);

    alunoNovo = new Aluno(6, "Ivo", 75.00);
    turma.inserir(alunoNovo);

    alunoNovo = new Aluno(3, "Olivia", 90.00);
    turma.inserir(alunoNovo);

    alunoNovo = new Aluno(2, "Julia", 88.00);
    turma.inserir(alunoNovo);

    alunoNovo = new Aluno(9, "Elaine", 85.00);
    turma.inserir(alunoNovo);

    alunoNovo = new Aluno(4, "Sônia", 82.00);
    turma.inserir(alunoNovo);

    System.out.println("Alunos matriculados na turma, ordenados em ordem crescente de
número de matrícula");
    turma.imprimirEmOrdem(); /// 2 - 3 - 4 - 5 - 6 - 8 - 9

    System.out.println("Resultado da pesquisa pelo aluno de número de matrícula 3:");
    alunoPesquisado = turma.buscar(3);
}
```

```

    if (alunoPesquisado != null)
    {
        alunoPesquisado.imprimir(); /// Olivia - 3 - 90.0
    }

    System.out.println("Número de alunos matriculados na turma: " + turma.numAlunos()); ///
7

    if ((alunoPesquisado = turma.menorNumeroMatricula()) != null)
    {
        System.out.println("Dados do aluno que possui o menor número de matrícula da turma.");
        alunoPesquisado.imprimir(); /// Julia - 2 - 88.0
    }

    turma.remover(5);

    turma.remover(3);

    turma.remover(4);

    System.out.println("Alunos matriculados na turma, ordenados em ordem crescente de
número de matrícula");
    turma.imprimirEmOrdem(); /// 2 - 6 - 8 - 9

    System.out.println("Número de alunos matriculados na turma: " + turma.numAlunos()); ///
4

    System.out.println("Resultado da pesquisa pelo aluno de número de matrícula 3:");
    alunoPesquisado = turma.buscar(3);
    if (alunoPesquisado != null)
    {
        alunoPesquisado.imprimir(); /// O aluno, cuja matrícula é 3, não foi encontrado na árvore.
    }

    turma.remover(9);

    turma.remover(6);

    turma.remover(2);

    turma.remover(8);

    if ((alunoPesquisado = turma.menorNumeroMatricula()) != null)
    {
        System.out.println("Dados do aluno que possui o menor número de matrícula da
turma.");
        alunoPesquisado.imprimir(); /// Nenhum aluno foi encontrado. A árvore está vazia.
    }
}

```