



## **ATIVIDADE PRÁTICA 6 - DESENVOLVIMENTO DE TAD PARA ÁRVORE BINÁRIA**

**CURSO:** TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

**DISCIPLINA:** ESTRUTURAS DE DADOS

**PERÍODO LETIVO:** 2024-01

**PROFESSOR:** FELIPE MARTIN SAMPAIO

### **INSTRUÇÕES:**

- Atividade avaliativa que será contabilizada como parte do instrumentos “Atividades práticas” da etapa N2;
- Deve ser enviado na tarefa disponível no Moodle:
  - Códigos .java com a implementação do TAD e função principal;
  - Pode ser enviado o projeto inteiro da IDE IntelliJ compactado como .zip.
- Prazo para entrega: **20 de maio de 2024**

### **OBJETIVOS:**

- Compreender o desenvolvimento de um Tipo Abstrato de Dado por meio do paradigma de orientação a objetos;
- Aplicar a compreensão sobre o funcionamento das operações planejadas para a estrutura de uma árvore binária na implementação dos métodos na linguagem de programação Java.

### **ESPECIFICAÇÃO:**

1. Desenvolver um Tipo Abstrato de Dado (TAD), utilizando a linguagem Java, para a estrutura de dados do tipo Árvore Binária.
2. Este TAD deve utilizar como armazenamento interno dos elementos da estrutura:

```
public class Nodo {  
    public int elem;  
    public Nodo esq, dir;  
    (...)
```

```
public class ArvoreTAD {  
    private Nodo raiz;  
    (...)
```

3. Além disso, o TAD desenvolvido deve oferecer as seguintes operações:

- a. Inicialização da árvore

```
public ArvoreTAD() { ... }
```



b. Calcula o número de elementos da árvore

```
public int tamanho() { ... }
```

c. Calcula a altura da árvore

```
public int altura() { ... }
```

d. Teste se a estrutura encontra-se vazia

```
public boolean estaVazia() { ... }
```

e. Inserção de um elemento na estrutura

```
public void insere(int elem) { ... }
```

f. Impressão dos elementos da estrutura na tela:

I. Impressão dos elementos através de um percurso em ordem:

```
public void imprimeEmOrdem() { ... }
```

II. Impressão dos elementos através de um percurso em pré ordem

```
public void imprimePreOrdem() { ... }
```

III. Impressão dos elementos através de um percurso em pós ordem

```
public void imprimePosOrdem() { ... }
```

IV. Impressão dos elementos no formato de árvore (para visualização)

```
public void imprimeFormatoArvore() { ... }
```

g. Remoção de um elemento da estrutura

```
public void remove(int elem) { ... }
```



h. Pesquisa por um elemento

```
public boolean pesquisa(int elem) { ... }
```

i. Acessa o menor elemento da árvore

```
public int acessaMenor() { ... }
```

j. Acessa o maior elemento da árvore

```
public int acessaMaior() { ... }
```

k. Limpa a estrutura (remove todos os elementos)

```
public void limpa() { ... }
```

l. Criar e retorna um vetor (objeto do tipo `ArrayList<Integer>`) contendo os elementos da árvore em ordem crescente

```
public ArrayList<Integer> criaVetorEmOrdem() { ... }
```

m. Realiza o balanceamento estático da árvore

```
public void balanceamentoEstatico() { ... }
```

4. Implementar uma função principal (main) para a criação de um objeto do TAD desenvolvido, como forma de testar as implementações das operações.

**IMPORTANTE:** deve ser desenvolvido um menu de operações, permitindo que o usuário escolha as operações que deseja fazer sobre a estrutura de árvore binária criada.

```
public static void main(String[] args) {  
    ArvoreTAD arv = new ArvoreTAD();  
    Scanner scan = new Scanner(System.in);  
  
    int op = scan.nextInt();  
  
    while(op != 9) {  
        System.out.println("MENU DE OPERAÇÕES:");  
        System.out.println("1 - Imprime elementos da árvore (em ordem)");  
        System.out.println("2 - Insere elemento na árvore");  
        System.out.println("3 - Pesquisa por um elemento na árvore");  
        // (...)  
  
        if(op == 1) {  
            //chamar operação de impressão dos elementos da árvore
```



```
    }  
    else if(op == 2) {  
  
        }  
        // (...)  
  
        op = scan.nextInt();  
    }  
  
}
```