

CURSO: Ciência da Computação

PERÍODO: 4o.

DISCIPLINA: Técnicas Alternativas de Programação

CÓDIGO: CI062

PROFESSOR: Andrey

AULA: 10

1 Apresentação

Na aula de hoje vamos apresentar e discutir e exercitar a Implementação de uma Árvore Binária de Busca em Java. A aula de hoje é baseada em DEITEL and DEITEL (2010)

2 Desenvolvimento

2.1 Classe Árvore

```
1 public class Arvore{
2     private Nodo raiz;
3
4     public Nodo getRaiz() {
5         return raiz;
6     }
7
8     public void setRaiz(Nodo raiz) {
9         this.raiz = raiz;
10    }
11
12    public Arvore() {
13        this.raiz = null;
14    }
15
16    public void insere(int elem){
17        if (this.raiz == null){
18            this.raiz = new Nodo (null,null,elem);
19            System.out.println("inseriu");
20        }else{
21            raiz = insereR(elem, this.raiz);
22        }
23    }
24    public Nodo insereR(int elem, Nodo p){
25        if (p == null){
26            p = new Nodo (null,null,elem);
27            System.out.println("inseriu "+ elem);
28            return p;
29        }
30        else{
31            if (p.getElem()>elem) {
32                p.setDir(insereR(elem, p.getDir()));
33                System.out.println("direita");
34            }
35            else if (p.getElem()<elem){
36                p.setEsq(insereR(elem, p.getEsq()));
37                System.out.println("esquerda");
38            }
39        }
```

```
40         return p;
41     }
42
43     public void imprimeR(Nodo p){
44         if (p == null) {
45             System.out.print("(");
46             return;
47         }
48         System.out.print("(" + p.getElem() + " ");
49         imprimeR(p.getDir());
50         imprimeR(p.getEsq());
51         System.out.print(")");
52     }
53
54
55 }
```

2.2 Classe Nodo

```
1     public class Nodo{
2         private Nodo esq, dir;
3         private int elem;
4         public Nodo getEsq() {
5             return esq;
6         }
7         public void setEsq(Nodo esq) {
8             this.esq = esq;
9         }
10        public Nodo getDir() {
11            return dir;
12        }
13        public void setDir(Nodo dir) {
14            this.dir = dir;
15        }
16        public int getElem() {
17            return elem;
18        }
19        public void setElem(int elem) {
20            this.elem = elem;
21        }
22        public Nodo(Nodo esq, Nodo dir, int elem) {
23            super();
24            this.esq = esq;
25            this.dir = dir;
26            this.elem = elem;
27        }
28    }
```

2.3 Classe TestaArvore

```
1     public class TestaArvore {
2         public static void main(String[] args) {
3             // TODO Auto-generated method stub
```

```
4      Arvore a = new Arvore();
5      a.insere(10);
6      a.insere(5);
7      a.insere(15);
8      a.insere(3);
9      a.insere(8);
10     a.insere(12);
11     a.insere(17);
12     a.imprimeR(a.getRaiz());
13     }
14 }
```

3 Atividades

1: Construa uma função parametrizada que receba um vetor de elementos de um tipo e os ordene.

2: Construa uma classe parametrizada Vetor que armazene elementos de um tipo em um array e tenha métodos para ordenar o array e realizar buscas no array.

Referências

DEITEL, P. and DEITEL, H. (2010). *Java: Como programar. 8ª Edição*. Prentice Hall, São Paulo.