

```

public class BIntNo {
    int valor;
    BIntNo esq, dir;

    BIntNo (int novoValor){
        valor = novoValor;
    }
}

```

```

public class BArvore {

    BIntNo Raiz;
    private BIntNo inserir (BIntNo arvore, int novoNo){
        if (arvore == null)
            return new BIntNo(novoNo);
        else
            {if (novoNo < arvore.valor)
                arvore.esq = inserir (arvore.esq, novoNo);
            else
                arvore.dir = inserir (arvore.dir, novoNo);
            return arvore;}
    }
    public void inserirNo (int novoValor){
        Raiz = inserir (Raiz, novoValor);
    }

    public void in_ordem(){
        exibir_in_ordem(Raiz);
    }
    private void exibir_in_ordem(BIntNo arvore){
        if (arvore!=null) {
            exibir_in_ordem(arvore.esq);
            System.out.println(arvore.valor);
            exibir_in_ordem(arvore.dir);
        }
    }
    public void pre_ordem(){
        exibir_pre_ordem(Raiz);
    }
    private void exibir_pre_ordem(BIntNo arvore){
        if (arvore!=null) {
            System.out.println(arvore.valor);
            exibir_pre_ordem(arvore.esq);
            exibir_pre_ordem(arvore.dir);
        }
    }

    public void pos_ordem(){
        exibir_pos_ordem(Raiz);
    }
    private void exibir_pos_ordem(BIntNo arvore){
        if (arvore!=null) {

```

```

        exibir_pos_ordem(arvore.esq);
        exibir_pos_ordem(arvore.dir);
        System.out.println(arvore.valor);
    }
}

private void exibir_esquerdo (BIntNo arv){
    if (arv != null){
        exibir_esquerdo (arv.esq);
        System.out.println (arv.valor);
    }
}
private void exibir_direito (BIntNo arv){
    if (arv != null){
        exibir_direito (arv.dir);
        System.out.println (arv.valor);
    }
}

public void exibir_raiz()
{System.out.println("Raiz " + Raiz.valor);
}

public void exibirNo_esq (){
    exibir_esquerdo (Raiz);
}

public void exibirNo_dir (){
    exibir_direito (Raiz);
}

public void busca (int v){
    BIntNo r = busca_abb (Raiz, v);
    if (r==null) System.out.println ("Valor não encontrado");
    else System.out.println ("Valor encontrado:" + r.valor);
}

public BIntNo busca_abb (BIntNo r,int v) {
    if( r==null) return null;
    else if(r.valor > v) return busca_abb(r.esq, v);
    else if(r.valor < v) return busca_abb(r.dir,v);
    else return r;
}

public void excluirNo (int item){
    try{
        BIntNo tempNo = Raiz, pai = null, filho = Raiz, temp;
        while (tempNo != null && tempNo.valor != item){
            pai = tempNo;
            if (item < tempNo.valor)
                tempNo = tempNo.esq;
            else
                tempNo = tempNo.dir;
        }
        if (tempNo == null)
            System.out.println ("Item nao localizado.");
    }
}

```

```

        if (pai == null){
            if (tempNo.dir == null)
                Raiz = tempNo.esq;
            else
                if (tempNo.esq == null)
                    Raiz = tempNo.dir;
                else{
                    for (temp = tempNo, filho = tempNo.esq;
                        filho.dir != null; temp = filho, filho =
filho.dir);

                    if (filho != tempNo.esq){
                        temp.dir = filho.esq;
                        filho.esq = Raiz.esq;
                    }
                    filho.dir = Raiz.dir;
                    Raiz = filho;
                }
            } else if (tempNo.dir == null){
                if (pai.esq == tempNo)
                    pai.esq = tempNo.esq;
                else
                    pai.dir = tempNo.esq;
            } else if (tempNo.esq == null){
                if (pai.esq == tempNo)
                    pai.esq = tempNo.dir;
                else
                    pai.dir = tempNo.dir;
                } else {
                    for (temp = tempNo, filho = tempNo.esq;
                        filho.dir != null; temp = filho, filho = filho.dir);
                    if (filho != tempNo.esq){
                        temp.dir = filho.esq;
                        filho.esq = tempNo.esq;
                    }
                    filho.dir = tempNo.dir;
                    if (pai.esq == tempNo)
                        pai.esq = filho;
                    else
                        pai.dir = filho;
                }
        } catch (NullPointerException erro) {
            //Item nao encontrado
        }
    }
}

```

```

public class usaArvore {
    public static void main(String[] args){
        BArvore arvore1 = new BArvore ();
        arvore1.inserirNo (14);
        arvore1.inserirNo (16);
    }
}

```

```

        arvore1.inserirNo (12);
        arvore1.inserirNo (11);
        arvore1.inserirNo (17);
        arvore1.inserirNo (15);
        System.out.println ("In-ordem ");
        arvore1.in_ordem();
        arvore1.inserirNo (10);
        arvore1.inserirNo (13);
        System.out.println ("In-ordem ");
        arvore1.in_ordem();
        arvore1.busca(12);
        arvore1.busca(20);
        System.out.println ("In-ordem ");

        arvore1.pre_ordem();
        System.out.println ("Pos-ordem ");
        arvore1.pos_ordem();

    }
}

```

```

public class usaArvore2 {
    public static void main(String[] args){
        BArvore arvore1 = new BArvore ();
        arvore1.inserirNo (14);
        arvore1.inserirNo (16);
        arvore1.inserirNo (12);
        arvore1.inserirNo (11);
        arvore1.inserirNo (17);
        arvore1.inserirNo (15);
        arvore1.exibirNo_esq();
        arvore1.exibirNo_dir();
        arvore1.inserirNo (10);
        arvore1.inserirNo (13);
        arvore1.exibir_raiz();
        System.out.println ("NÔs a esquerda ");
        arvore1.exibirNo_esq();
        System.out.println ("NÔs a direita ");
        arvore1.exibirNo_dir();
        System.out.println("Exclus,,o de nÔ");
        arvore1.excluirNo(12);
        arvore1.exibir_raiz();
        System.out.println ("Arvore apÔs a exclus,,o do nÔ de valor
12");
        System.out.println ("NÔs a esquerda ");
        arvore1.exibirNo_esq();
        System.out.println ("NÔs a direita ");
        arvore1.exibirNo_dir();
    }
}

```