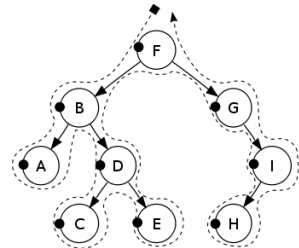


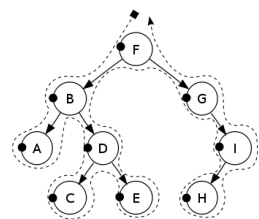
Árvore Binária – Percursos Pré, Em e Pós Ordem

Publicado em junho 7, 2015junho 7, 2015 por surfx2013



Código simples em Java de percurso em árvores binárias Pré, Em e Pós ordem.

Exemplo de leitura Pré, Em e Pós Ordem da árvore:



(<https://emersonbsi.files.wordpress.com/2015/06/2arvore.png>)

PreOrdem (RED): F,B,A,D,C,E,G,I,H
InOrdem (ERD): A,B,C,D,E,G,H,I
PosOrdem (EDR): A,C,E,D,B,H,I,G,F

```

200 public static void main(String[] args) {
201     System.out.println("Exemplo de percurso em árvore binária");
202     Arvore arvore = new Arvore("F");
203     arvore.adicionar("B", "F", "esquerda");
204     arvore.adicionar("G", "F", "direita");
205     arvore.adicionar("A", "B", "esquerda");
206     arvore.adicionar("D", "B", "direita");
207     arvore.adicionar("C", "D", "esquerda");
208     arvore.adicionar("E", "D", "direita");
209     arvore.adicionar("I", "G", "direita");
210     arvore.adicionar("H", "I", "esquerda");
211     System.out.println("PreOrdem (RED): " + arvore.preOrdem());
212     System.out.println("InOrdem (ERD): " + arvore.inOrdem());
213     System.out.println("PosOrdem (EDR): " + arvore.posOrdem());
214 }
215
216 arvore.clear();
217 arvore.adicionar("F", "F", "esquerda");
218 arvore.adicionar("B", "F", "direita");
219 arvore.adicionar("A", "B", "esquerda");
220 arvore.adicionar("D", "B", "direita");
221 arvore.adicionar("C", "D", "esquerda");
222 arvore.adicionar("E", "D", "direita");
223 arvore.adicionar("I", "G", "direita");
224 arvore.adicionar("H", "I", "esquerda");
225
226 }

```

(<https://emersonbsi.files.wordpress.com/2015/06/1percursoarvore.png>)

Formas de Leitura:

- Pré-Ordem: RED = raiz, esquerda, direita
- In-Ordem: ERD = esquerda, raiz, direita
- Pós-Ordem: EDR = esquerda, direita, raiz

Crie um projeto no Eclipse chamado 'Arvores'

Crie a classe 'Node':

```

1 package arvore.binary;
2
3 /**
4  * Classe nó
5  *
6  * @author Emerson Shigueo Sugimoto 07/06/2015 11:42
7  */
8 public class Node {
9
10     private String nome;
11     private Node esquerda;
12     private Node direita;
13
14     public String getNome() { return nome; }
15     public void setNome(String nome) { this.nome = nome; }
16     public Node getEsquerda() { return esquerda; }
17     public void setEsquerda(Node esquerda) { this.esquerda = esquerda; }
18     public Node getDireita() { return direita; }
19     public void setDireita(Node direita) { this.direita = direita; }
20
21     public Node(){
22         this(null, null, null);
23     }
24     public Node(String nome){
25         this(nome, null, null);
26     }
27
28     public Node(String nome, Node esquerda, Node direita){
29         setNome(nome);
30         setEsquerda(esquerda);
31         setDireita(direita);
32     }
33
34     @Override
35     public int hashCode() {
36         return getNome() == null ? 0 : getNome().hashCode();
37     }
38
39     @Override
40     public boolean equals(Object obj) {
41         if (obj == null || !(obj instanceof Node)) { return false; }
42         return ((Node)obj).hashCode() == hashCode();
43     }
44
45     @Override
46     public String toString() {
47         return getNome() == null ? "" : getNome();
48     }
49
50 }

```

O objetivo da classe nó ('Node') é ser um nodo da árvore binária, possuindo um identificador (nome) e os ponteiros (referências) para as subárvores da esquerda (private Node esquerda) e direita (private Node direita).

Publicidade



Manicure Pro

Manicure Pro - Cintia Stasiak

Curso De Manicure mais completo do mercado, com todas as especializações e módulos extras de marketing e administração

O Curso Mais Completo Do Mercado



Crie a classe 'ArvoreBin':

```
1 package arvore.binary;
2
3 /**
4  * <b>Árvore binária</b>
5  *
6  * <ul>
7  * <li>RED = <u>r</u>aiz, <u>e</u>squerda, <u>d</u>ireita</li>
8  * <li>ERD = <u>e</u>squerda, <u>r</u>aiz, <u>d</u>ireita</li>
9  * <li>EDR = <u>e</u>squerda, <u>d</u>ireita, <u>r</u>aiz</li>
10 * </ul>
11 * @author Emerson Shiguo Sugimoto 07/06/2015 11:39
12 */
13 public class ArvoreBin {
14
15     /**
16      * ao invés deste enum, poderia ser usado um boolean
17      */
18     public enum ModoRL {
19         esquerda, direita
20     }
21
22     private Node arvore;
23
24     public Node getArvore() { return arvore; }
25     public void setArvore(Node arvore) { this.arvore = arvore; }
26
27     /**
28      *
29      * @param raiz raiz da árvore
30      */
31     public ArvoreBin(String raiz){
32         setArvore(new Node(raiz));
33     }
34
35     /**
36      * adiciona um novo nó
37      * @param noPai nome do nó pai
38      * @param nomeNo nome do novo nó
39      * @param modo posição: direita / esquerda
40      */
41     public void AdicionarNo(String noPai, String nomeNo, ModoRL modo){
42         if (modo == ModoRL.direita) {
43             AdicionarNoDireita(noPai, nomeNo);
44         } else if (modo == ModoRL.esquerda) {
45             AdicionarNoEsquerda(noPai, nomeNo);
46         }
47     }
48
49     public void AdicionarNoEsquerda(String noPai, String nomeNo){
50         Node no = findNode(getArvore(), noPai);
51         if (no == null) { return; }
52         no.setEsquerda(new Node(nomeNo));
53     }
54
55     public void AdicionarNoDireita(String noPai, String nomeNo){
56         Node no = findNode(getArvore(), noPai);
57         if (no == null) { return; }
58         no.setDireita(new Node(nomeNo));
59     }
60
61     /**
62      * A ideia não é criar uma árvore binária ordenada,
63      * por isto a busca exaustiva pelos nós.
64
65      * A principal ideia é permitir criar qualquer tipo
66      * de árvore, com qualquer tipo de ordenação.
67      * @param no
68      * @param nome
69      * @return
70      */
71     private Node findNode(Node no, String nome){
72         if (no == null || nome == null || nome.isEmpty()) { return null; }
73         if (no.getName().equals(nome)) { return no; }
74         Node noesquerda = findNode(no.getEsquerda(), nome);
75         if (noesquerda != null) { return noesquerda; }
76         return findNode(no.getDireita(), nome);
77     }
78
79     /**
80      * limpa a árvore
81      * @param raiz raiz da árvore
82      */
83     public void Clear(String raiz){
84         nullnodes(getArvore());
85         setArvore(new Node(raiz));
86     }
87
88     /**
89      * método recursivo auxiliar para limpar a árvore
90      * @param no
91      */
92     private void nullnodes(Node no){
93         if (no == null) { return; }
94         nullnodes(no.getDireita());
95         nullnodes(no.getEsquerda());
96         no.setEsquerda(null); no.setDireita(null);
97         no = null;
98     }
99
100     /**
101      * RED
102      * @return
103      */
104     public String PreOrdem(){
105         if (getArvore() == null) { return ""; }
106         return readPreOrdem(getArvore());
107     }
108
109     /**
110      * RED
111      * @param no
112      * @return
113      */
114     private String readPreOrdem(Node no){
115         if (no == null) { return ""; }
116         String rt = no.toString();
117         if (no.getEsquerda() != null) {
118             rt += (rt.isEmpty() ? "" : ",") + readPreOrdem(no.getEsquerda());
119         }
120         if (no.getDireita() != null) {
121             rt += (rt.isEmpty() ? "" : ",") + readPreOrdem(no.getDireita());
122         }
123         return rt;
124     }
125
126     /**
127      * ERD
128      * @return
129      */
130     public String InOrdem(){
131         if (getArvore() == null) { return ""; }
132         return readInOrdem(getArvore());
133     }
134
135     /**
136      * ERD
137      * @param no
138      * @return
139      */
140     private String readInOrdem(Node no){
```

```

141         if (no == null) {return "";}
142         String rt = "";
143         if (no.getEsquerda() != null) {
144             rt += (rt.isEmpty() ? "" : ",") + readInOrdem(no.getEsquerda());
145         }
146         rt += (rt.isEmpty() ? "" : ",") + no.toString();
147         if (no.getDireita() != null) {
148             rt += (rt.isEmpty() ? "" : ",") + readInOrdem(no.getDireita());
149         }
150         return rt;
151     }
152
153     /**
154     * EDR
155     * @return
156     */
157     public String PosOrdem(){
158         if (getArvore() == null) { return ""; }
159         return readPosOrdem(getArvore());
160     }
161
162     /**
163     * EDR
164     * @param no
165     * @return
166     */
167     private String readPosOrdem(Node no){
168         if (no == null) {return "";}
169         String rt = "";
170         if (no.getEsquerda() != null) {
171             rt += (rt.isEmpty() ? "" : ",") + readPosOrdem(no.getEsquerda());
172         }
173         if (no.getDireita() != null) {
174             rt += (rt.isEmpty() ? "" : ",") + readPosOrdem(no.getDireita());
175         }
176         rt += (rt.isEmpty() ? "" : ",") + no.toString();
177         return rt;
178     }
179 }
180 }

```

A classe 'ArvoreBin' contém a referência para a árvore binária (private Node arvore), os métodos para inserir nós à direita e esquerda de um nó (nó pai) e forma de percurso recursivo na árvore binária.

O método

```

1 | private Node findNode(Node no, String nome){

```

têm por objetivo encontrar um nó na árvore por busca exaustiva, pois, o objetivo da árvore não é ordenar a entrada de novos nós, mais sim permitir ao usuário inserir nós na ordem e posição que desejar, pois a finalidade da estrutura é realizar o estudo das formas de leitura de árvore binária: Pré, Em e Pós-Ordem.

E por fim a classe 'Main':

```
1 package arvore.main;
2
3 import arvore.binary.ArvoreBin;
4 import arvore.binary.ArvoreBin.ModorL;
5
6 /**
7  * <b>Classe Main - Árvore binária</b>
8
9  * Objetivo: estudar percurso em árvore binária.
10  * <ul>
11  * <li>Pré-Ordem: RED = <u>r</u>aiz, <u>e</u>squerda, <u>d</u>ireita</li>
12  * <li>In-Ordem: ERD = <u>e</u>squerda, <u>r</u>aiz, <u>d</u>ireita</li>
13  * <li>Pós-Ordem: EDR = <u>e</u>squerda, <u>d</u>ireita, <u>r</u>aiz</li>
14  * </ul>
15  *
16  * @author Emerson Shigueo Sugimoto 07/06/2015 11:41
17  */
18
19 public class Main {
20
21     public static void main(String[] args) {
22         System.out.println("-----");
23         ArvoreBin arvore = new ArvoreBin("A");
24
25         arvore.AdicionarNo("A", "B", ModorL.esquerda);
26         arvore.AdicionarNo("B", "D", ModorL.esquerda);
27         arvore.AdicionarNo("A", "C", ModorL.direita);
28         arvore.AdicionarNo("C", "E", ModorL.esquerda);
29         arvore.AdicionarNo("E", "G", ModorL.direita);
30         arvore.AdicionarNo("C", "F", ModorL.direita);
31         arvore.AdicionarNo("F", "H", ModorL.esquerda);
32         arvore.AdicionarNo("F", "I", ModorL.direita);
33
34         System.out.println("PreOrdem (RED): " + arvore.PreOrdem());
35         System.out.println("InOrdem (ERD): " + arvore.InOrdem());
36         System.out.println("PosOrdem (EDR): " + arvore.PosOrdem());
37         System.out.println("-----");
38
39         arvore.Clear("F");
40
41         arvore.AdicionarNo("F", "B", ModorL.esquerda);
42         arvore.AdicionarNo("B", "A", ModorL.esquerda);
43         arvore.AdicionarNo("B", "D", ModorL.direita);
44         arvore.AdicionarNo("D", "C", ModorL.esquerda);
45         arvore.AdicionarNo("D", "E", ModorL.direita);
46         arvore.AdicionarNo("F", "G", ModorL.direita);
47         arvore.AdicionarNo("G", "I", ModorL.direita);
48         arvore.AdicionarNo("I", "H", ModorL.esquerda);
49
50         System.out.println("PreOrdem (RED): " + arvore.PreOrdem());
51         System.out.println("InOrdem (ERD): " + arvore.InOrdem());
52         System.out.println("PosOrdem (EDR): " + arvore.PosOrdem());
53         System.out.println("-----");
54
55         arvore.Clear("A");
56
57         arvore.AdicionarNo("A", "B", ModorL.esquerda);
58         arvore.AdicionarNo("B", "C", ModorL.esquerda);
59         arvore.AdicionarNo("C", "E", ModorL.esquerda);
60         arvore.AdicionarNo("E", "I", ModorL.direita);
61         arvore.AdicionarNo("C", "F", ModorL.direita);
62         arvore.AdicionarNo("F", "J", ModorL.direita);
63         arvore.AdicionarNo("B", "D", ModorL.direita);
64         arvore.AdicionarNo("D", "G", ModorL.esquerda);
65         arvore.AdicionarNo("D", "H", ModorL.direita);
66         arvore.AdicionarNo("H", "K", ModorL.esquerda);
67         arvore.AdicionarNo("H", "L", ModorL.direita);
68
69         System.out.println("PreOrdem (RED): " + arvore.PreOrdem());
70         System.out.println("InOrdem (ERD): " + arvore.InOrdem());
71         System.out.println("PosOrdem (EDR): " + arvore.PosOrdem());
72         System.out.println("-----");
73     }
74 }
75
76 /**
77
78 Saída
79
80 PreOrdem (RED): A,B,D,C,E,G,F,H,I
81 InOrdem (ERD): D,B,A,E,G,C,H,F,I
82 PosOrdem (EDR): D,B,G,E,H,I,F,C,A
83
84 PreOrdem (RED): F,B,A,D,C,E,G,I,H
85 InOrdem (ERD): A,B,C,D,E,F,G,H,I
86 PosOrdem (EDR): A,C,E,D,B,H,I,G,F
87
88 PreOrdem (RED): A,B,C,E,I,F,J,D,G,H,K,L
89 InOrdem (ERD): E,I,C,F,J,B,G,D,K,H,L,A
90 PosOrdem (EDR): I,E,J,F,C,G,K,L,H,D,B,A
91
92
93 */
```

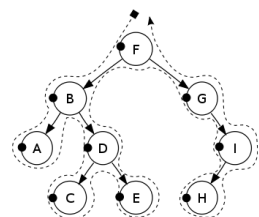
O resultado será:

PreOrdem (RED): A,B,D,C,E,G,F,H,I
InOrdem (ERD): D,B,A,E,G,C,H,F,I
PosOrdem (EDR): D,B,G,E,H,I,F,C,A

PreOrdem (RED): F,B,A,D,C,E,G,I,H
InOrdem (ERD): A,B,C,D,E,F,G,H,I
PosOrdem (EDR): A,C,E,D,B,H,I,G,F

PreOrdem (RED): A,B,C,E,I,F,J,D,G,H,K,L
InOrdem (ERD): E,I,C,F,J,B,G,D,K,H,L,A
PosOrdem (EDR): I,E,J,F,C,G,K,L,H,D,B,A

Um exemplo de inserção da árvore binária via código é:



(<https://emersonbsi.files.wordpress.com/2015/06/2arvore.png>)

```
1 arvore.Clear("F");
2
3 arvore.AdicionarNo("F", "B", ModoRL.esquerda);
4 arvore.AdicionarNo("B", "A", ModoRL.esquerda);
5 arvore.AdicionarNo("B", "D", ModoRL.direita);
6 arvore.AdicionarNo("D", "C", ModoRL.esquerda);
7 arvore.AdicionarNo("D", "E", ModoRL.direita);
8 arvore.AdicionarNo("F", "G", ModoRL.direita);
9 arvore.AdicionarNo("G", "I", ModoRL.direita);
10 arvore.AdicionarNo("I", "H", ModoRL.esquerda);
```

A leitura será:

```
1 System.out.println("PreOrdem (RED): " + arvore.PreOrdem());
2 System.out.println("InOrdem (ERD): " + arvore.InOrdem());
3 System.out.println("PosOrdem (EDR): " + arvore.PosOrdem());
```

PreOrdem (RED): F,B,A,D,C,E,G,I,H
InOrdem (ERD): A,B,C,D,E,F,G,H,I
PosOrdem (EDR): A,C,E,D,B,H,I,G,F

o código pode ser encontrado em: <https://mega.co.nz/#ZAaHVK7I!HvO0CXrjpiFRvqO21CAKO1WrmL7EN9UAC13C4rawHiA> (<https://mega.co.nz/#ZAaHVK7I!HvO0CXrjpiFRvqO21CAKO1WrmL7EN9UAC13C4rawHiA>)

Sponsored Content

Calcule quanto você poderia ganhar investindo \$ 249 na Itaú e em outras ações Junte-se ao sucesso | Sponsored

(https://tracking.capital-gennr-online.com/34f6441-a6f3-473c-9691-02db89d0022ad?id=0008e8be6b265a34b820da3e77564193d&adtitle=Calcule+quanto+você

%3C%3A+poderia+ganhar+investindo+\$249+na+Ita%3C%3A+e+em+outras+ações+Junte-se+ao+sucesso=&source_id&campaignid=00476ceec9ed9ee2c0e980a7e1b7b1ab4§ionid=section_id§ionname=Section_name&publisherid=publisher_id&publishername=publisher_name&ob_click_id=sob_click_id&obOrigUrl=true)

Azeites de Oliva da Espanha Os Mais Premiados do Mundo Azeites da Espanha | Sponsored
(https://www.osazeitedesdaespanha.com/?utm_source=outbrain&utm_medium=cpm&utm_campaign=institucional&utm_term=awareness&utm_content=display&obOrigUrl=true)

Mulher sofre com ressecamento vaginal por não conhecer isto Lubrifique Mito
 ival | Sponsored
 (https://plus.maispopulares.com.br/aff/02/ob/desejo?src=outbrain&OutbrainClickId=sob_click_id&affiliate_code=USATUTJ5OMM7VMXF&utm_source=outbrain&utm_medium=section_name&utm_campaign=YS-
 XTC-OB-KV-019-030&utm_content=Mulher+sofre+com+ressecamento+vaginal+por+n
 %C3%A3o+conhecer+isto&utm_term=YS-XTC-OB-KV-019-0090&obOrigUrl=true)

Próstata inchada: "Homens precisam anular essa substância agora, ou será tarde demais" Eu imploro para que você veja isso | Sponsored
(<https://receitasnaturais.maispopulares.com.br/aff/01/ob/prostata-inchada?obOrigUrl=true>)

Gel estimulante deixa maridos com sorriso de orelha a orelha. Chega de Falhar na hora H! Sponsored
(https://podnatural3.maispopulares.com.br/aff/01/ob/virilidade?src=outbrain&affiliate_code=HWBJLL7GWOK8VBMB&OutbrainClickId=&ob_click_id&utm_source=outbrain&utm_medium=Section_name&utm_campaign=C3&utm_content=Gel-estimulante+deixa+maridos+com+sorriso+de+orelha+a+orelha.&utm_term=YS-RAZ-OB-HG-001-0001-&obOrigUrl=true)

Óculos para prevenir acidentes em estradas Últimas Unidades! | Sponsored
(https://trk.alhawattatili.com/a/9a331c2-
eae6-4625-9979-8eb432066117?campaign_id=00a87740370b9fd17c2367ec45a403a468&
publisher_id=\$publisher_id&publisher_name=\$publisher_name&id=
ad_id=00a76f86ae17900d8240d9dc6d1baadf6&ad_title=
%3C%39culos+para+prevenir+acidentes+em+estradas&
section_name=\$section_name&req_id=\$req_id&
promoted_link_id=00a76f86ae17900d8240d9dc6d1baadf6&
time_stamp=\$time_stamp&ob_click_id=00b_cdb_click_id&
campaign_id=00a87740370b9fd17c2367ec45a403a468&
publisher_id=\$publisher_id&publisher_name=\$publisher_name&
ad_id=00a76f86ae17900d8240d9dc6d1baadf6&ad_title=
%3C%39culos+para+prevenir+acidentes+em+estradas&
section_name=\$section_name&req_id=\$req_id&
promoted_link_id=00a76f86ae17900d8240d9dc6d1baadf6&
time_stamp=\$time_stamp&ob_click_id=00b_cdb_click_id&ObUrl=true)

Custos de cremação: Opções incrivelmente acessíveis para sua despedida - Pesquise aqui Cremação | Anúncios de pesquisa | Sponsors

(https://trck.tracking505.com/491afaf41-4e6f-4771-9d65-8c8e2672d9c7/campaign_id=0091e96c2bda8f5ed47f315b59026ae9a&publisher_id=Publisher_id&id=Publisher_name=Publisher_name&ad_id=00d69153255ef74aa275c7c1b78578cae6f&title=Custos+de+crema%C3%A7%C3%A3o+C3%A0+Op%C3%A7%C3%B5es+incrivelmente+acess%C3%A0+para+sua+despedida+Pesquise+aqui+Crema%C3%A7%C3%A3o+An%C3%AAn%C3%AAn%C3%AAn%C3%AAn+Section_id=Section_id§ion_name=Section_name&req_id=Req_id&promo_link_id=00d69153255ef74aa275c7c1b78578cae6f&time_stamp=Time_stamp&cost=Cost&click_id=Click_id&origUrl=true)

FioraniSofás e Sofas não vendidos são distribuídos quase de graça (veja os preços)Sofas e sofás | Links Patrocinados | Sponsored
%24/%3Fc3llsdzktxnwnnr.com?subid=§_name\$&adtitle=
%24/%3Bcity/%7D%24+sof%C3%A1s+não+v%C3%A9ndidos+s
o%C3%A3o+distribui%C3%AAdos+quase+de+gra%C3%A7a+%28veja+os+pê
subid3-§_network_id&site=§_publisher_name\$&_section_name\$&
§_website_outbrain|OutbrainClickId=\$ob_click_id&subid4=ar-br-a-sofa#pt-
636255-283108-ob-sm&ckw1=1-Sof%C3%A1s+Modernos+Brasil&
kw2=Sofa+na+Promo%C3%A7%C3%A3o&
kw3=Saldao+De+Sofa+Lista+De+Preço+C3%A7os&
kw4=Jogar+De+Sofa+2+3+4+Lugares&kw5=Saldao+De+Sofa+Agora+Mesmo&
kw6=Saldao+De+Sofa+|City|&kw7=Venda+de+sof
%C3%A1s+de+esquina+Brasil+2023+para+idosos&
kw8=lista+de+Venda+de+sof%C3%A1s+de+esquina+Brasil+2023+par
a+idosos&kw9=Venda+muito+barata+de+conjuntos+de+assentos+de+2023+Para+idosos&kw
10=lista+de+DoF+sof%C3%A1+expans%C3%ADvel+160X200+Para+idosos&
obOrigUrl=true)

Tenis de court legítimo sorprende pelo preço Últimas Unidades! | Sponsored
 (https://trk.althawattatiti.com/b?c3e40b4-70b7-442b-
 98e0-65c04ebae54a?campaign_id=00aa5cd6b83cd7a6bb9c3bbb7bctc7f758&
 publisher_id=&publisher_id&&publisher_name=&publisher_name&&
 ad_id=001132e423dbe72295243dfd8260ab882&ad_title=T
 %C3%AAnis+de+court+legitimo%3CA7Dimo+sorprende+pelo+preço%C3%A7o&
 section_name=Section_name&req_id=\$req_id&\$req_id&
 promoted_link_id=001132e423dbe72295243dfd8260ab882&
 time_stamp=Time_stamp&ob_click_id=Sub_click_id&
 campaign_id=00aa5cd6b83cd7a6bb9c3bbb7bctc7f758&
 publisher_id=&publisher_id&&publisher_name=&publisher_name&&
 ad_id=001132e423dbe72295243dfd8260ab882&ad_title=T
 %C3%AAnis+de+court+legitimo%3CA7Dimo+sorprende+pelo+preço%C3%A7o&
 section_name=Section_name&req_id=\$req_id&\$req_id&
 promoted_link_id=001132e423dbe72295243dfd8260ab882&
 time stamp=Time stamp&ob click id=Sub click id&&OrigUrl=true)

Postado em Árvore Binária, Estruturas de Dados, Java, Programação Com a tag árvore binária, Em Ordem Pós Ordem, Java, percurso, Pré Ordem 1 comentário

Um comentário sobre “Árvore Binária – Percursos Pré, Em e Pós Ordem”

1. **Gabi.P.P** disse:
abril 6, 2021 às 11:38 pm
Muito bom, gostaria o código em C/C++

RESPONDER

Blog no WordPress.com.

