```java
public class HumanList {
   //Campos base
   private HumanNode first;
   private HumanNode last;
   //Tamanho da lista
   int nodeCount = 0;
   //Construtor
   public HumanList () {
   }
   /**     Insere um nodo por atributos    */
   public void insere (    String  n,
                           String  a,
                           String  c,
                           char    s,
                           int     i
                   ) {

           HumanNode nod = new HumanNode ( n, a, c, s, i );

           if ( this.empty() ) {
                   first = last = nod;
           }
           else {
                   last.next = nod;
                   nod.prev = last;
                   last = nod;
           }
           nodeCount++;

   }
   /**     Insere um nodo instanciado      */
   public void insere (HumanNode nod) {
           if ( this.empty() ) {
                   first = last = nod;
           }
           else {
                   last.next = nod;
                   nod.prev = last;
                   last = nod;
           }
           nodeCount++;
   }
   /**     Retorn um nodo na posicao x da lista */
   public HumanNode getNodo ( int x ) {

           HumanNode n;

           if ( x < this.size() ) {
                   n = first;
                   while ( x-- > 0) n = n.next;
           }
           else {
                   n = last;
                   while ( x++ < this.size() ) n = n.prev;
           }

           return n;
   }
   /**     Pesquisa os nodos da lista em busca de um com o Campo CPF determinado */
   public HumanNode getByCPF ( String cpf ) {

           HumanNode atual = this.first;

           int x = 0;

           while ( x < this.size() ) {
                   if ( cpf.equalsIgnoreCase (atual.getCPF()) ) return atual;
                   else {
                           atual = atual.next;
                           x++;
                   }

           }

           return new HumanNode();
   }
   /**     Pesquisa os nodos da lista por todos com o campo name determinado */
```

```
 78:     public HumanList getByName ( String name ) {
 79:             HumanList results = new HumanList();
 80:             int x = 0;
 81:             HumanNode atual = this.first;
 82:
 83:             while ( x < this.size() ) {
 84:                     if ( name.equalsIgnoreCase (atual.getName()) ) results.insere (atual);
 85:                     atual = atual.next; x++;
 86:             }
 87:             return results;
 88:     }
 89:     /**     Remove da lista um nodo instanciado     */
 90:     public void removeNodo ( HumanNode nod ) {
 91:             if ( this.empty() ) {
 92:                     first = last = null;
 93:             }
 94:             else {
 95:                     if ( nod == first ) {
 96:                             first = nod.next;
 97:                             nod.next.prev = null;
 98:                     }
 99:                     else if ( nod == last ) {
100:                             last = nod.prev;
101:                             nod.prev.next = null;
102:                     }
103:                     else {
104:                             nod.next.prev = nod.prev;
105:                             nod.prev.next = nod.next;
106:                     }
107:
108:             }
109:             nodeCount--;
110:     }
111:     /**     Remove o nodo na posicao x da lista     */
112:     public void remove (int x) {
113:             HumanNode c = getNodo (x);
114:             this.removeNodo (c);
115:     }
116:     /**     Retorna  o tamanho, ou quantidade de nodos, da lista    */
117:     public int size () {
118:             return nodeCount;
119:     }
120:     /**     Retorna true caso a lista esteja vazia, caso contrario, retorna false   */
121:     public boolean empty () {
122:             if ( this.size() == 0 ) return true;
123:             else return false;
124:     }
125:     /**     Retorna o primeiro elemento     */
126:     public HumanNode getFirst() {
127:             return this.first;
128:     }
129:     /**     Retorna o Segundo elemento      */
130:     public HumanNode getLast() {
131:             return this.last;
132:     }
133:     /**     Imprime todos os nodos da lista na tela */
134:     public void printList(){
135:             HumanNode obj = first;
136:             for (int i=0; i< this.size(); i++) {
137:                     obj.printMe();
138:                     obj = obj.next;
139:             }
140:     }
141:     /**     Retorna todos os nodos da lista numa String     */
142:     public String printListG(){
143:             HumanNode obj = first;
144:             String retorno = "";
145:             for (int i=1; i< this.size()+1; i++) {
146:                     retorno +="Ocorrencia "+i+"\n"+
147:                     obj.printMeG();
148:                     obj = obj.next;
149:             }
150:             return retorno;
151:     }
152: }
```