

## Cliente.java

```

1 package Dominio;
2
3 import java.io.Serializable;
4
5
6
7
8
9 public class Cliente implements Serializable{
10
11     private int cartaIdentita;
12     private String nome;
13     private String cognome;
14     private String domicilio;
15     private String email;
16     private int eSocio;
17
18     public Cliente(int ci, String n, String cn, String d, String e, int es) {
19         super();
20         this.cartaIdentita = ci;
21         this.nome = n;
22         this.cognome = cn;
23         this.domicilio = d;
24         this.email = e;
25         this.eSocio = es;
26     }
27     public Cliente() {
28         super();
29     }
30     public int getCartaIdentita() {
31         return cartaIdentita;
32     }
33     public void setCartaIdentita(int ci) {
34         this.cartaIdentita = ci;
35     }
36     public String getNome() {
37         return nome;
38     }
39     public void setNome(String n) {
40         this.nome = n;
41     }
42     public String getCognome() {
43         return cognome;
44     }
45     public void setCognome(String c) {
46         this.cognome = c;
47     }
48     public String getDomicilio() {
49         return domicilio;
50     }
51     public void setDomicilio(String d) {
52         this.domicilio = d;
53     }
54     public String getEmail() {
55         return email;
56     }
57     public void setEmail(String email) {
58         this.email = email;
59     }
60     public int getEsocio() {
61         return eSocio;
62     }
63     public void setEsocio(int e) {
64         this.eSocio = e;
65     }
66     public static int creareCliente(int c,String n,String cn, String d,String e,int

```

## Cliente.java

```

    es){
167         int res=0;
168         Cliente cliente= new Cliente(c,n,cn,d,e,es);
169         try {
170             DBAgente db=new DBAgente();
171             res=db.creareClienteBD(cliente);
172         } catch (RemoteException e1) {
173             e1.printStackTrace();
174         }
175         return res;
176     }
177
178     public static List<Cliente> consultareClienti(){
179         List<Cliente> clienti=null;
180         try {
181             DBAgente db=new DBAgente();
182             clienti = db.consultareClientiDB();
183         } catch (RemoteException e) {
184             e.printStackTrace();
185         }
186         return clienti;
187     }
188
189     public static Cliente consultareCliente(int cartaIdentita){
190         Cliente c=null;
191         try {
192             DBAgente db=new DBAgente();
193             c = db.consultareClienteDB(cartaIdentita);
194         } catch (RemoteException e) {
195             e.printStackTrace();
196         }
197         return c;
198     }
199
200     public static int modificareCliente(int cartaI,String n,String cn, String
    d,String e,int es){
201         int res=0;
202         Cliente c;
203         try {
204             DBAgente db=new DBAgente();
205             c = db.consultareClienteDB(cartaI);
206             if(c.getEsocio()==1){
207                 if(es==1){
208                     c.setNome(n);
209                     c.setCognome(cn);
210                     c.setDomicilio(d);
211                     c.setEmail(e);
212                     res=db.modificareClienteDB(c);
213                 }
214                 else{
215                     c.setNome(n);
216                     c.setCognome(cn);
217                     c.setDomicilio(d);
218                     c.setEmail(e);
219                     Socio.cancellareClienteDB(cartaI);
220                 }
221             }
222             else{
223                 c.setNome(n);
224                 c.setCognome(cn);
225                 c.setDomicilio(d);
226                 c.setEmail(e);

```

## Cliente.java

```

127         c.setEsocio(es);
128         res=db.modificareClienteDB(c);
129     }
130 } catch (RemoteException e1) {
131     e1.printStackTrace();
132 }
133 return res;
134 }
135
136 public static int cancellareCliente(int cartaIdentita){
137     int res=0;
138     Cliente c=consultareCliente(cartaIdentita);
139     if(c.getEsocio()==1){
140         res=-1;
141     }
142     else{
143         try {
144             DBAgente db=new DBAgente();
145             res=db.cancellareClienteDB(cartaIdentita);
146         } catch (RemoteException e) {
147             e.printStackTrace(); res=-1;
148         }
149     }
150     return res;
151 }
152
153 public static List<Cliente> consultareClientiNoSoci() {
154     List<Cliente> clienti=null;
155     try {
156         DBAgente db=new DBAgente();
157         clienti = db.consultareClientiNoSocioDB();
158     } catch (RemoteException e) {
159         e.printStackTrace();
160     }
161     return clienti;
162 }
163
164 public static List<Cliente> consultareClientiSociTitolari() {
165     List<Socio> ls=null;
166     try {
167         DBAgente db=new DBAgente();
168         ls = db.consultareSociTitolariDB();
169     } catch (RemoteException e) {
170         e.printStackTrace();
171     }
172     LinkedList<Cliente> lc=new LinkedList<Cliente> ();
173     for (int i = 0; i < ls.size(); i++) {
174         try {
175             DBAgente db=new DBAgente();
176             lc.add(db.consultareClienteDB(ls.get(i).getCartaIdentita()));
177         } catch (RemoteException e) {
178             e.printStackTrace();
179         }
180     }
181     return lc;
182 }
183
184 public static List<Cliente> consultareClienteGruppo(int idHold) {
185     List<Cliente> l=null;
186     try {
187         DBAgente db=new DBAgente();
188         l = db.consultareClienteGruppoDB(idHold);

```

## Cliente.java

```

189         } catch (RemoteException e) {
190             e.printStackTrace();
191         }
192         return l;
193     }
194
195     @Override
196     public int hashCode() {
197         final int prime = 31;
198         int res = 1;
199         res = prime * res + ((cognome == null) ? 0 : cognome.hashCode());
200         res = prime * res + cartaIdentita;
201         res = prime * res + ((domicilio == null) ? 0 : domicilio.hashCode());
202         res = prime * res + ((email == null) ? 0 : email.hashCode());
203         res = prime * res + eSocio;
204         res = prime * res + ((nome == null) ? 0 : nome.hashCode());
205         return res;
206     }
207
208     @Override
209     public boolean equals(Object obj) {
210         if (this == obj)
211             return true;
212         if (obj == null)
213             return false;
214         if (getClass() != obj.getClass())
215             return false;
216         final Cliente other = (Cliente) obj;
217         if (cognome == null) {
218             if (other.cognome != null)
219                 return false;
220         } else if (!cognome.equals(other.cognome))
221             return false;
222         if (cartaIdentita != other.cartaIdentita)
223             return false;
224         if (domicilio == null) {
225             if (other.domicilio != null)
226                 return false;
227         } else if (!domicilio.equals(other.domicilio))
228             return false;
229         if (email == null) {
230             if (other.email != null)
231                 return false;
232         } else if (!email.equals(other.email))
233             return false;
234         if (eSocio != other.eSocio)
235             return false;
236         if (nome == null) {
237             if (other.nome != null)
238                 return false;
239         } else if (!nome.equals(other.nome))
240             return false;
241         return true;
242     }
243 }

```