

```
1:
2: program Corbi;
3:
4: uses
5:   Vcl.Forms,
6:   Console1 in 'Console1.pas' {Form2},
7:   Login1 in 'Login1.pas' {Form1},
8:   AnaAtl in 'AnaAtl.pas' {Form3},
9:   Vcl.Themes,
10:  Vcl.Styles,
11:  AnaDspl in 'AnaDspl.pas' {Form4},
12:  AnaGa in 'AnaGa.pas' {Form5},
13:  Gara in 'Gara.pas' {Form6},
14:  aiuto in 'aiuto.pas' {Form7},
15:  Classifiche in 'Classifiche.pas' {Form8},
16:  Tempi in 'Tempi.pas' {Form9},
17:  Dialogs,
18:  Windows,
19:  System.UITypes;
20:
21: //no2
22: {$R *.res}
23: const
24:   TentaLancio='CodeCall.Net Mutex'; //no2
25: resourcestring //no2
26:   DUPLICATE_MSG='Il programma è stato già lanciato. Non consente duplicati'; //no2
27: var //no2
28:   vMutex :THandle; //no2
29: {$D+}
30: begin //codice per evitare doppia istanza
31:   vMutex :=CreateMutex(nil,True,TentaLancio); //no2
32:   if GetLastError = ERROR_ALREADY_EXISTS then //no2
33:     begin //no2
34:       MessageDlg(DUPLICATE_MSG,mtError,[mbOK],0);//no2
35:       Exit; //no2
36:     end; //no2
37:   try //no2
38:     Application.Initialize;
39:     Application.MainFormOnTaskbar := True;
40:     Application.Title := 'Corbi';
41:     Application.HelpFile := 'corbi.chm';
42:     Application.CreateForm(TForm2, Form2);
43:     Application.CreateForm(TForm1, Form1);
44:     Application.CreateForm(TForm7, Form7);
45:     Application.CreateForm(TForm9, Form9);
46:     //posporre a Form2 che contiene il database
47:     Application.ShowMainForm := False;
48:     Application.Run;
49:   finally //no2
50:     CloseHandle(vMutex); //no2
51:   end; //no2
52: end.
```

```
1: unit Console1;
2:
3: interface
4:
5: uses
6:   Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
7:   Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Data.DB, ABSMain, frxClass,
8:   frxDBSet, System.Types, JvBaseDlg, JvSHFileOperation, frxExportPDF,
9:   Vcl.ExtCtrls, System.UITypes, Vcl.Imaging.jpeg, Vcl.HtmlHelpViewer, Winapi.ShellAPI;
10:
11: type
12:   TForm2 = class(TForm)
13:     Esci: TButton;
14:     Button1: TButton;
15:     Button2: TButton;
16:     Button3: TButton;
17:     Button4: TButton;
18:     ABSDatabase1: TABSDatabase;
19:     ABSTable1: TABSTable;
20:     ABSTable2: TABSTable;
21:     frxDBDataset1: TfrxDBDataset;
22:     ABSTable4: TABSTable;
23:     ABSTable5: TABSTable;
24:     ABSTable6: TABSTable;
25:     ABSTable9: TABSTable;
26:     ABSTable7: TABSTable;
27:     ABSTable8: TABSTable;
28:     ABSTable12: TABSTable;
29:     ABSTable10: TABSTable;
30:     ABSTable11: TABSTable;
31:     Button5: TButton;
32:     Button6: TButton;
33:     Button7: TButton;
34:     frxPDFExport1: TfrxPDFExport;
35:     ABSTable3: TABSTable;
36:     ABSTable16T: TABSTable;
37:     Button8: TButton;
38:     Button9: TButton;
39:     ABSQuery2: TABSQuery;
40:     frxReport2: TfrxReport;
41:     frxDBDataset2: TfrxDBDataset;
42:     ABSTable3D: TABSTable;
43:     Button10: TButton;
44:     Button11: TButton;
45:     frxReportLog: TfrxReport;
46:     frxDBDatasetLog: TfrxDBDataset;
47:     Image1: TImage;
48:     Image2: TImage;
49:     Label1: TLabel;
50:     JvSHFile1: TJvSHFileOperation;
51:     frxReport1: TfrxReport;
52:     procedure FormCreate(Sender: TObject);
53:     procedure EsciClick(Sender: TObject);
54:     procedure Button1Click(Sender: TObject);
55:     procedure Button2Click(Sender: TObject);
56:     procedure Button3Click(Sender: TObject);
57:     procedure Button4Click(Sender: TObject);
58:     procedure JVSHFile1Show(Sender: TObject);
59:     procedure Button5Click(Sender: TObject);
60:     procedure Button6Click(Sender: TObject);
61:     procedure Button7Click(Sender: TObject);
62:     procedure Button8Click(Sender: TObject);
63:     procedure Button9Click(Sender: TObject);
64:     procedure Button10Click(Sender: TObject);
65:     procedure Button11Click(Sender: TObject);
66:
```

```

68:  private
69:    { Private declarations }
70:  public
71:    function GetAppVersionStr: string;
72:    { Public declarations }
73:  end;
74:
75:  var
76:    Form2: TForm2;
77:    consflg:Boolean=True;    // quando è True Form2 è closed
78:    Log1,versione:string;
79:    riparato:Boolean;
80:    KBHook: HHook;
81:    function KeyboardHookProc(Code: Integer; WordParam: Word; LongParam: LongInt): LongInt; stdcall
82:  implementation
83:    uses          //per utilizzare il form3 e Form4 ecc
84:      AnaAtl,Login1,AnaDspl,AnaGa,Gara,Classifiche,aiuto;
85:
86:    {$R *.dfm}
87:
88:  procedure TForm2.FormCreate(Sender: TObject);
89:  begin
90:    ABSDatabase1.DatabaseFileName:=ExtractFilePath(Application.ExeName)+'corbi.ABS';
91:    Application.HelpFile := ExtractFilePath(Application.ExeName) + 'Corbi.chm';
92:    ABSDatabase1.Open;
93:    ABSTable1.TableName:='Servizio';
94:    ABSTable1.Open;
95:    ABSTable1.Edit;
96:    Form2.Visible:=False;
97:    consflg:=False;
98:    KBHook:=SetWindowsHookEx(WH_KEYBOARD,@KeyboardHookProc,HInstance,GetCurrentThreadId());
99:    GetAppVersionStr;
100:    caption:=caption + ' - ver: '+versione;
101:  end;
102:
103:  function KeyboardHookProc(Code: Integer; WordParam: Word; LongParam: LongInt) : LongInt;
104:  begin          //per chiamre l'help da F1
105:
106:    if WordParam=VK_F1 then
107:      ShellExecute(0, nil,PChar('Corbi.chm'), nil, nil, SW_SHOWNormal)
108:    else
109:      Result:=0; //necessaria per sbloccare l'intercettazione per tasti diversi da F1
110:    end;
111:
112:  function TForm2.GetAppVersionStr: string;
113:  var
114:    Exe: string;
115:    Size, Handle: DWORD;
116:    Buffer: TBytes;
117:    FixedPtr: PVSFixedFileInfo;
118:  begin
119:    Exe := ParamStr(0);
120:    Size := GetFileVersionInfoSize(PChar(Exe), Handle);
121:    if Size = 0 then
122:      RaiseLastOSError;
123:      SetLength(Buffer, Size);
124:    if not GetFileVersionInfo(PChar(Exe), Handle, Size, Buffer) then
125:      RaiseLastOSError;
126:    if not VerQueryValue(Buffer, '\\', Pointer(FixedPtr), Size) then
127:      RaiseLastOSError;
128:    Result := Format('%d.%d.%d.%d',
129:      [LongRec(FixedPtr.dwFileVersionMS).Hi, //major
130:      LongRec(FixedPtr.dwFileVersionMS).Lo, //minor
131:      LongRec(FixedPtr.dwFileVersionLS).Hi, //release
132:      LongRec(FixedPtr.dwFileVersionLS).Lo]); //build
133:    versione:= Result;
134:

```

```
135:
136:
137: procedure TForm2.JVSHFile1Show(Sender: TObject);
138: begin
139:     ShowMessage('Effettuato il backup del Database');
140: end;
141:
142:
143: procedure TForm2.Button1Click(Sender: TObject); //Anagrafica atleti
144: begin                                //Form2 =Console
145:     if Form2.visible=True then
146:     Form2.Visible:=False;
147:     if atlflg=False then
148:     begin
149:         Form3:= TForm3.Create(Self);
150:         Form3.show ;
151:     end
152:     else          //essendo atlflg True occorre ri-creare il Form3 prima di poterlo
153:     begin        // rendere visibile se si ripete il click
154:         Form3:= TForm3.Create(Self);
155:         Form3.Visible:=True;
156:         atlflg:=False;
157:     end;
158: end;
159:
160: procedure TForm2.Button2Click(Sender: TObject); //Anagrafiva discipline
161: begin
162:     if Form2.visible=True then
163:     Form2.Visible:=False;
164:     if Dsplflg=False then
165:     begin
166:         Form4:= TForm4.Create(Self);
167:         Form4.show ;
168:     end
169:     else          //essendo atlflg True occorre ri-creare il Form3 prima di poterlo
170:     begin        // rendere visibile se si ripete il click
171:         Form4:= TForm4.Create(Self);
172:         Form4.Visible:=True;
173:     end;
174: end;
175:
176: procedure TForm2.Button3Click(Sender: TObject); //Anagrafica Gare
177: begin
178:     if Form2.visible=True then
179:     Form2.Visible:=False;
180:     if AnaGAFlg=False then
181:     begin
182:         Form5:= TForm5.Create(Self);
183:         Form5.show;
184:     end
185:     else          //essendo AnaGAFlg True occorre ri-creare il Form5 prima di poterlo
186:     begin        // rendere visibile se si ripete il click
187:         Form5:= TForm5.Create(Self);
188:         Form5.Visible:=True;
189:     end;
190: end;
191:
192: procedure TForm2.Button4Click(Sender: TObject); // Gara Corrente Iscrizioni
193: begin
194:     if Form2.visible=True then
195:     Form2.Visible:=False;
196:     if GAFlg=False then
197:     begin
198:         Form6:= TForm6.Create(Self);
199:         Form6.show ;
200:     end
201:     else          //essendo GAFlg True (cioè Form chiuso) occorre ri-creare il Form6
```

```

203:         Form6:= TForm6.Create(Self);
204:         Form6.Visible:=True;
205:     end;
206: end;
207:
208: procedure TForm2.Button5Click(Sender: TObject);    // Backup
209: begin
210:     if MessageDlg('Confermi di voler effettuare il backup del DataBase ?',
211:         mtconfirmation, [mbYes,mbNo], 0) = mrYes then
212:     begin
213:         ABSDatabase1.Close;
214:         JVSHFile1.Execute;
215:         // JVSHFile1.OnClose(Self);
216:         Form2.ABSTable1.Open;
217:         Form2.ABSTable1.Last;
218:         if Form2.ABSTable1.FieldName('Password').AsString<>' ' then
219:             Form2.ABSTable1.Insert;
220:             Form2.ABSTable1.Edit;
221:             Form2.ABSTable1.FieldName('IN').AsDateTime:=NOW;
222:             Form2.ABSTable1.FieldName('what').AsString:='Effettuato Backup';
223:             Form2.ABSTable1.Post;
224:             Form2.ABSTable1.Close;
225:         end else Exit;
226:
227:
228: end;
229:
230: procedure TForm2.Button6Click(Sender: TObject);    // Ripara
231:     Var
232:     result:string;
233: begin
234:     Form2.ABSDatabase1.Close;
235:     result:= ABSDatabase1.RepairDatabase;
236:     if result = '' then
237:         ShowMessage('Il Database è stato verificato ')else
238:         ShowMessage('Il Database è corrotto - Usa il backup');
239:
240: end;
241:
242: procedure TForm2.Button7Click(Sender: TObject);    //Classifiche
243: begin
244:     if Form2.visible=True then
245:         Form2.Visible:=False;
246:         if ClassFlg=False then
247:             begin
248:                 Form8:= TForm8.Create(Self);
249:                 Form8.show ;
250:             end
251:         else //essendo ClassFlg True occorre ri-creare il Form8 prima di poterlo
252:             begin // rendere visibile se si ripete il click
253:                 Form8:= TForm8.Create(Self);
254:                 Form8.Visible:=True;
255:             end;
256: end;
257:
258: procedure TForm2.Button8Click(Sender: TObject);    //Azzer Storico
259: begin
260:     if MessageDlg('Stai per azzerare l''intero Storico delle Classifiche. Vuoi continuare ?'
261: ,
262:         mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrYes then
263:     begin
264:         Form2.ABSTable16T.Close;
265:         Form2.ABSTable16T.EmptyTable;
266:         ShowMessage('Tutta lo Storico è stato cancellato !');
267:         Form2.ABSTable1.Open;
268:         Form2.ABSTable1.Last;
269:         Form2.ABSTable1.Insert;

```

```
270:         Form2.ABTable1.FieldByName('IN').AsDateTime:=NOW;
271:         Form2.ABTable1.FieldByName('what').AsString:='Cancellato lo Storico delle Classif
iche';
272:         Form2.ABTable1.Post;
273:         Form2.ABTable1.Close;
274:     end;
275:
276: end;
277:
278: procedure TForm2.Button9Click(Sender: TObject);    //Complessivo
279: var
280:     ca1,ca2 : Integer;
281:     d1,d2:TDateTime;
282:     trovato:Boolean;
283: begin
284:     ABTable16T.Open;
285:     if MessageDlg('Desideri ottenere il punteggio storico per Atleta ?'
286:     +#13+'Dovrai fornire i codici della prima ed ultima Gara da conteggiare',
287:     mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrNo then Exit;
288:     ca1:=StrToInt(InputBox('Codice Gara iniziale','Inserisci la prima',''));
289:     if ca1<>0 then
290:     ca2:=StrToInt(InputBox('Codice Gara finale','Inserisci l''ultima',''));
291:     trovato:=ABTable16T.Locate('CODGA',ca1,[]);
292:     if trovato=True then
293:     d1:=ABTable16T.FieldByName('DATAGA').AsDateTime else
294:     begin
295:         ShowMessage('Gara codice '+inttostr(ca1)+' non esiste.');
```

296: Exit;

297: end;

298: trovato:=ABTable16T.Locate('CODGA',ca2,[]);

299: if trovato=True then

300: d2:=ABTable16T.FieldByName('DATAGA').AsDateTime else

301: begin

302: ShowMessage('La Gara con codice '+inttostr(ca2)+' non contiene dati di classifiche.');

303: Exit;

304: end;

305: if (ca2=0) or (ca1=0) then

306: begin

307: ShowMessage('Nessun dato inserito');

308: Exit;

309: end;

310: Form8.ABTable19.Close;

311: Form8.ABTable19.EmptyTable;

312: Form8.ABTable19.Open;

313: with ABSQuery2 do

314: begin

315: Close;

316: SQL.Text:='INSERT INTO totale(CODATL,COGNOME,NOME,NAZIONE,RNK) SELECT CODATL,COGNOME,

317: NOME,NAZIONE,SUM(RNK) AS RNK ' +

318: ' FROM T16T where CODGA BETWEEN '+QuotedStr(IntToStr(ca1))+' AND '+QuotedStr(IntToStr

319: (ca2))+' GROUP BY'+

320: ' CODATL,COGNOME,NOME,NAZIONE ORDER BY RNK desc';

321: ExecSQL;

322: //Open;

323: end;

324: frxReport2.Variables['ini']:= quotedstr(DateToStr(d1));

325: frxReport2.Variables['fin']:= quotedstr(DateToStr(d2));

326: frxReport2.ShowReport;

327: ABTable16T.Close;

328: Form8.ABTable19.Close;

329: Form8.ABTable19.EmptyTable;

330:

331: end;

332:

333: procedure TForm2.Button10Click(Sender: TObject);

```
335: s:string;
336: begin                                     //per mascherare i caratteri premetter #31
337:
338:     s:=InputBox('Richiesta PW',#31'Inserisci PW dell''Amministratore','');
339:     if s<>'Corbi77' then Exit;
340:     frxReportLog.ShowReport;
341: end;
342:
343: procedure TForm2.Button11Click(Sender: TObject); //Cancella il log
344: var
345: s:string;
346: n:Integer;
347: begin
348:     s:=InputBox('Richiesta PW',#31'Inserisci PW dell''Amministratore','');
349:     Form2.ABSTable1.Open;
350:     Form2.ABSTable1.Edit;
351:     if s<>'Corbi77' then Exit;
352:     n:=Form2.ABSTable1.RecordCount;
353:     if MessageDlg('Il LOG ha '+IntToStr(n)+' records, vuoi cancellarlo ? '
354: ,mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrNo then
355:         Exit;
356:     Form2.ABSTable1.First;
357:     while not Form2.ABSTable1.Eof do
358:     begin
359:         if Form2.ABSTable1.FieldName('Password').AsString<>' ' then
360:             Form2.ABSTable1.Next else
361:             Form2.ABSTable1.Delete; //NB il delete causa già un next
362:         end;
363:     Form2.ABSTable1.Close;
364: end;
365:
366: procedure TForm2.EsciClick(Sender: TObject); //Esci
367: begin
368:     Form2.ABSTable1.Open;
369:     Form2.ABSTable1.Last;
370:     Form2.ABSTable1.Insert;
371:     Form2.ABSTable1.Edit;
372:     Form2.ABSTable1.FieldName('OUT').AsDateTime:=NOW;
373:     Form2.ABSTable1.FieldName('what').AsString:='Chiusura della Sessione';
374:     Form2.ABSTable1.Post;
375:     Form2.ABSTable1.Close;
376:     ABSDataBasel.Close;
377:     Form2.release;
378:     consflg:=True;
379:     Application.Terminate;
380: end;
381:
382: end.
```

```

1: unit AnaAtl;    // Unità per il carico dell'anagrafe atleti
2:
3: interface
4:
5: uses
6:   Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
7:   Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, ABSMain, Data.DB,
8:   Vcl.Grids, Vcl.DBGrids, Vcl.DBCGrids, Vcl.ExtCtrls, Vcl.DBCtrls, System.UITypes,
9:   frxClass, frxDBSet, System.Types;
10:
11: type
12:   TForm3 = class(TForm)
13:     Button1: TButton;
14:     ABSQuery1: TABSQuery;
15:     dbgrd1: TDBGrid;
16:     dbnvgr1: TDBNavigator;
17:     Button2: TButton;
18:     Button3: TButton;
19:     Button4: TButton;
20:     Edit1: TEdit;
21:     Edit2: TEdit;
22:     Label1: TLabel;
23:     Label2: TLabel;
24:     Button5: TButton;
25:     ds1: TDataSource;
26:     Label3: TLabel;
27:     procedure Button1Click(Sender: TObject);
28:     procedure FormCreate(Sender: TObject);
29:     procedure Button5Click(Sender: TObject);
30:     procedure Button2Click(Sender: TObject);
31:     procedure dbnvgr1Click(Sender: TObject; Button: TNavigateBtn);
32:     procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
33:     procedure Button3Click(Sender: TObject);
34:     procedure Button4Click(Sender: TObject);
35:     procedure ds1DataChange(Sender: TObject; Field: TField);
36:
37:   private
38:     { Private declarations }
39:   public
40:     { Public declarations }
41:   end;
42:
43: var
44:   Form3: TForm3;
45:   atlflg: Boolean = True; // Quando è True il form3 è closed
46:   filtro: string; // per SQL
47:   inserisco: Boolean = False; // flag per inserimento dati;
48:   canclose: Boolean = True;
49: implementation
50:   uses
51:     Console1;
52:   {$R *.dfm}
53:   {$D+}
54:   procedure TForm3.FormCreate(Sender: TObject);
55:   begin    // ricordarsi di assegnare una datasource al Navigator
56:     Form2.ABSTable2.TableName := 'AnaAtl';
57:     Form2.ABSTable2.Open;
58:     atlflg := False;
59:     ds1.DataSet := Form2.ABSTable2; // aggiunto per evitare circular DataLink error
60:     Form2.ABSTable2.Edit;
61:     Form2.ABSTable1.Open;
62:     Form2.ABSTable1.INSERT;
63:     Form2.ABSTable1.FieldName('IN').AsDateTime := NOW;
64:     Form2.ABSTable1.FieldName('what').AsString := 'Accesso alla Anagrafica Atleti';
65:     Form2.ABSTable1.FieldName('OUT').AsString := '';
66:     Form2.ABSTable1.Post;

```



```

68: end;
69:
70: //Intercetto DBNavigator
71: procedure TForm3.dbnvgr1Click(Sender: TObject; Button: TNavigateBtn);
72:
73: begin      //per selezionare il primo campo
74:
75:     case button of
76:         nbInsert:      //se inserisce un nuovo record
77:             begin
78:                 Form2.ABSTable2.Last;
79:                 Form2.ABSTable2.Append;
80:                 inserisco:=True; //inserisce il flag nel nuovo record
81:                 dbgrd1.SetFocus;
82:                 DBGrd1.SelectedField := Form2.ABSTable2.FieldName('NOME');
83:             end ;
84:
85:         nbPost:        //se lo carica nella tabella
86:             inserisco:=False ;
87:         end;
88: end;
89:
90:     procedure TForm3.ds1DataChange(Sender: TObject; Field: TField);
91: begin
92:
93: end;
94:
95: // per monitorare il form close quando si è aggiunto un record
96: procedure TForm3.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
97: begin
98:     if inserisco=True then
99:         if MessageDlg('Hai un dato da salvare ?' , mtconfirmation, [mbYes,mbNo], 0) = mrYes
100:        then
101:            begin
102:                CanClose:=False;
103:                Exit;
104:            end
105:        else
106:            begin
107:                CanClose:=True;
108:                if inserisco=True then
109:                    inserisco:=False;
110:                    Button1Click(Self);
111:                end;
112:                if CanClose=False then
113:                    CanClose:=True;
114:            end;
115:
116:
117: procedure TForm3.Button1Click(Sender: TObject); // Esci
118:
119: begin
120:     if inserisco=True then //se ha inserito un record
121:         begin
122:             FormCloseQuery(Self,canclose);
123:         end
124:         else
125:             begin
126:                 Form2.ABSTable1.Open;
127:                 Form2.ABSTable1.Last;
128:                 if Form2.ABSTable1.FieldName('Password').AsString<>' ' then
129:                     Form2.ABSTable1.Insert;
130:                     Form2.ABSTable1.Edit;
131:                     Form2.ABSTable1.FieldName('OUT').AsDateTime:=NOW;
132:                     Form2.ABSTable1.Post;
133:                     Form2.ABSTable1.Close;
134:                     Form2.ABSTable2.Close;

```

```
136:         atflg:=True; //per consentire di monitorizzare che Form3 è closed
137:         Form2.Visible:=True;
138:         Form3.Release;
139:     end;
140: end;
141:
142: procedure TForm3.Button2Click(Sender: TObject); //cerca
143: label
144: vai;
145: var
146: chi:array[1..2]of shortint;
147: i:ShortInt;
148: begin
149:     ds1.DataSet:=ABSQuery1;
150:     i:=0; //determina se nome o/e cognome è in ricerca
151:     if edit1.text<>' ' then
152:     begin
153:         i:=i+1;
154:         chi[i]:=1;
155:     end;
156:     if edit2.Text<>' ' then
157:     begin
158:         i:=i+1;
159:         chi[i]:=2;
160:     end;
161:     if chi[2]>0 then //elabora i query
162:     begin
163:         filtro:='NOME LIKE ' + QuotedStr('%'+Edit1.Text+'%')+ ' AND COGNOME LIKE '
164:         +quotedStr('%'+edit2.text+'%');
165:         goto vai;
166:     end;
167:     if chi[1]>0 then
168:     begin
169:         if (edit1.Text<>' ') then
170:             filtro:= 'NOME LIKE ' +quotedstr('%'+Edit1.Text+'%');
171:         if (edit2.Text<>' ') then
172:             filtro:='COGNOME LIKE ' +QuotedStr('%'+Edit2.Text+'%');
173:         goto vai;
174:     end;
175:     vai :
176:         //effettua il query
177:         with ABSQuery1 do
178:         begin
179:             close;
180:             if i>0 then
181:             begin
182:                 SQL.Text := 'select * from AnaAtl where '+filtro; //Attenzione agli spazi prima e dopo
gli apici
183:                 ExecSql;
184:             end else
185:             begin
186:                 ShowMessage('Nessun elemento di ricerca inserito');
187:                 Exit;
188:             end;
189:             Open;
190:         end;
191:     end;
192:
193: procedure TForm3.Button3Click(Sender: TObject); //Resetta la ricerca
194: begin
195:     Edit1.Text:='';
196:     Edit2.Text:='';
197:     ds1.DataSet:=Form2.ABSTable2;
198: end;
199:
200: //STAMPA
201: procedure TForm3.Button4Click(Sender: TObject);
```

```
203:      Form2.frxReport1.ShowReport;
204:      if MessageDlg('Vuoi stamparlo ?' , mtconfirmation, [mbYes,mbNo], 0,mbNo) = mrNo
205:      then Exit else
206:          Form2.frxReport1.Print;
207:      end;
208:
209:      //AZZERARE il database Atleti
210:      procedure TForm3.Button5Click(Sender: TObject);
211:      begin
212:          if MessageDlg('Stai per azzerare l''intera Anagrafe degli atleti. Vuoi continuare ?',
213:          mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrYes then
214:              begin
215:                  Form2.ABSTable2.Close;
216:                  Form2.ABSTable2.EmptyTable;
217:                  ShowMessage('Tutta l''anagrafe è stata cancellata !');
218:              end;
219:
220:      end;
221:
222:
223:      end.
```

```

1: unit AnaDspl;
2:
3: interface
4:
5: uses
6:   Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
7:   Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, ABSMain, Vcl.ExtCtrls,
8:   Vcl.DBCtrls, Vcl.StdCtrls, Vcl.Grids, Vcl.DBGrids, frxClass, frxDBSet, System.UITypes,
9:   frxExportPDF;
10:
11: type
12:   TForm4 = class(TForm)
13:     dbgrd4: TDBGrid;
14:     Label1: TLabel;
15:     Edit1: TEdit;
16:     Button1: TButton;
17:     Button2: TButton;
18:     Button3: TButton;
19:     Button4: TButton;
20:     Button5: TButton;
21:     dbnvgr4: TDBNavigator;
22:     ds4: TDataSource;
23:     ABSQuery4: TABSQuery;
24:     frxReport4: TfrxReport;
25:     frxDBDataset4: TfrxDBDataset;
26:     frxPDFExport4: TfrxPDFExport;
27:     procedure Button5Click(Sender: TObject);
28:     procedure FormCreate(Sender: TObject);
29:     procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
30:     procedure Button1Click(Sender: TObject);
31:     procedure Button2Click(Sender: TObject);
32:     procedure Button4Click(Sender: TObject);
33:     procedure Button3Click(Sender: TObject);
34:     procedure dbnvgr4Click(Sender: TObject; Button: TNavigateBtn);
35:   private
36:     { Private declarations }
37:   public
38:     { Public declarations }
39:   end;
40:
41: var
42:   Form4: TForm4;
43:   DsplFlg: Boolean = True;    // il crea lo cambia a False
44:   inserisco4: Boolean = False;
45:   CanClose: Boolean = True;
46:   filtro: string;
47: implementation
48:   uses
49:     Console1;
50:   {$R *.dfm}
51:   {$D+}
52:   procedure TForm4.FormCreate(Sender: TObject);
53:   begin
54:     Form2.ABSTable4.TableName := 'AnaDspl';
55:     Form2.ABSTable4.Open;
56:     TDateTimeField(Form2.ABSTable4.FieldByName('TMAX')).DisplayFormat := 'nn:ss,zzz';
57:     DsplFlg := False;
58:     ds4.DataSet := Form2.ABSTable4; //aggiunto per evitare circular DataLink error
59:     Form2.ABSTable4.Edit;
60:     Form2.ABSTable1.Open;
61:     Form2.ABSTable1.INSERT;
62:     Form2.ABSTable1.FieldByName('IN').AsDateTime := NOW;
63:     Form2.ABSTable1.FieldByName('what').AsString := 'Accesso alla Anagrafica Discipline';
64:     Form2.ABSTable1.FieldByName('OUT').AsString := '';
65:     Form2.ABSTable1.Post;
66:     Form2.ABSTable1.Close;

```

```
68:
69:
70: procedure TForm4.Button1Click(Sender: TObject); //Cerca
71:
72: begin
73:     ds4.DataSet:=ABSQuery4;
74:     if Edit1.Text<>' ' then
75:     begin
76:         filtro:='DESCR LIKE ' + QuotedStr('%'+Edit1.Text+'%');
77:         //effettua il query
78:         with ABSQuery4 do
79:             begin
80:                 close;
81:                 SQL.text := 'select * from AnaDspl where '+filtro; //Attenzione agli spazi prima e
dopo gli apici
82:                 ExecSql;
83:                 open;
84:             end;
85:         end else
86:         begin
87:             ShowMessage('Nessun elemento di ricerca inserito');
88:             Exit;
89:         end;
90:     end;
91:
92:
93: procedure TForm4.Button2Click(Sender: TObject); //Annulla la ricerca
94: begin
95:     Edit1.Text:='';
96:     ds4.DataSet:=Form2.ABSTable4;
97: end;
98:
99: procedure TForm4.Button3Click(Sender: TObject); //Stampa
100: begin
101:     frxReport4.ShowReport; ///??
102:     if MessageDlg('Vuoi stamparlo ?' , mtconfirmation, [mbYes,mbNo], 0,mbNo) = mrNo
103:     then Exit else
104:         frxReport4.Print;
105: end;
106:
107: procedure TForm4.Button4Click(Sender: TObject); //Azzera la tabella
108: begin
109:     if MessageDlg('Stai per azzerare l''intera Anagrafe delle Discipline. Vuoi continuare ?',
110:         mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrYes then
111:         begin
112:             Form2.ABSTable4.Close;
113:             Form2.ABSTable4.EmptyTable;
114:             ShowMessage('Tutta l''anagrafe è stata cancellata !');
115:         end;
116: end;
117:
118: procedure TForm4.Button5Click(Sender: TObject); //Esci
119: begin
120:     if inserisco4=True then //se ha inserito un record
121:     begin
122:         FormCloseQuery(Self,canclose);
123:     end
124:     else
125:     begin
126:         Form2.ABSTable1.Open;
127:         Form2.ABSTable1.Last;
128:         if Form2.ABSTable1.FieldName('Password').AsString<>' ' then
129:             Form2.ABSTable1.Insert;
130:             Form2.ABSTable1.Edit;
131:             Form2.ABSTable1.FieldName('OUT').AsDateTime:=NOW;
132:             Form2.ABSTable1.Post;
133:             Form2.ABSTable1.Close;
```

```
135:      Form4.Close;    // NB chiama il can close
136:      DsplFlg:=True;  //per consentire di monitorizzare che Form3 è closed
137:      Form2.Visible:=True;
138:      Form4.Release;
139:      end;
140: end;
141:
142: procedure TForm4.dbnvgr4Click(Sender: TObject; Button: TNavigateBtn);
143: begin                // azioni del DENavigator
144:   case button of
145:     nbInsert:        //se inserisce un nuovo record
146:       begin
147:         Form2.ABSTable4.Last;
148:         Form2.ABSTable4.Append;
149:         inserisco4:=True; //inserisce il flag nel nuovo record
150:         dbgrd4.SetFocus;
151:         DBGrd4.SelectedField := Form2.ABSTable4.FieldName('DESCR');
152:       end ;
153:
154:     nbPost:          //se lo carica nella tabella
155:       inserisco4:=False ;
156:     end;
157: end;
158:
159: procedure TForm4.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
160: begin                // Chiusura del Form 4 DSPL
161:   if inserisco4=True then
162:     if MessageDlg('Hai un dato da salvare ?' , mtconfirmation, [mbYes,mbNo], 0) = mrYes
163:     then
164:       begin
165:         CanClose:=False;
166:         Exit;
167:       end
168:     else
169:       begin
170:         CanClose:=True;
171:         if inserisco4=True then
172:           inserisco4:=False;
173:           Button5Click(Self);
174:         end;
175:       end;
176: end.
```

```

1: unit AnaGa;           //Anagrafica Gare    11/04/2015
2:
3: interface
4:
5: uses
6:   Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
7:   Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Vcl.Grids, Vcl.DBGrids, System.UITypes,
8:   Vcl.StdCtrls, Vcl.ExtCtrls, Vcl.DBCtrls, ABSMain, frxClass, frxDBSet,
9:   frxExportPDF;
10:
11: type
12:   TForm5 = class(TForm)
13:     dbgrd51: TDBGrid;
14:     ds51: TDataSource;
15:     dbgrd52: TDBGrid;
16:     ds52: TDataSource;
17:     Button1: TButton;
18:     dbnvgr51: TDBNavigator;
19:     dbgrd55: TDBGrid;
20:     ds55: TDataSource;
21:     dbnvgr55: TDBNavigator;
22:     cbb55: TComboBox;
23:     ABSQuery6: TABSQuery;
24:     Label1: TLabel;
25:     Label2: TLabel;
26:     Label3: TLabel;
27:     Edit1: TEdit;
28:     Edit2: TEdit;
29:     Button2: TButton;
30:     Label4: TLabel;
31:     Button3: TButton;
32:     Button4: TButton;
33:     shp1: TShape;
34:     Label5: TLabel;
35:     Label6: TLabel;
36:     Edit3: TEdit;
37:     Button5: TButton;
38:     Button6: TButton;
39:     frxReport51: TfrxReport;
40:     frxDBDataset51: TfrxDBDataset;
41:     Button7: TButton;
42:     frxPDFExport51: TfrxPDFExport;
43:     Button8: TButton;
44:     Button9: TButton;
45:     Button10: TButton;
46:     shp2: TShape;
47:     Label7: TLabel;
48:     frxReport5: TfrxReport;
49:     frxDBDatasetBA: TfrxDBDataset;
50:     frxDBDatasetGA: TfrxDBDataset;
51:     frxDBDatasetGB: TfrxDBDataset;
52:     procedure FormCreate(Sender: TObject);
53:     procedure Button1Click(Sender: TObject);
54:     procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
55:     procedure cbb55Change(Sender: TObject);
56:     procedure dbgrd52MouseUp(Sender: TObject; Button: TMouseButton;
57:       Shift: TShiftState; X, Y: Integer);
58:     procedure dbnvgr51Click(Sender: TObject; Button: TNavigateBtn);
59:     procedure dbgrd55MouseDown(Sender: TObject; Button: TMouseButton;
60:       Shift: TShiftState; X, Y: Integer);
61:     procedure Pulizia (Sender: TObject);
62:     procedure dbgrd51CellClick(Column: TColumn);
63:     procedure Button2Click(Sender: TObject);
64:     procedure Button3Click(Sender: TObject);
65:     procedure Button4Click(Sender: TObject);
66:     procedure Button5Click(Sender: TObject);
67: You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)

```

```

68:     procedure Edit3Click(Sender: TObject);
69:     procedure Button7Click(Sender: TObject);
70:     procedure Button8Click(Sender: TObject);
71:     procedure Button9Click(Sender: TObject);
72:     procedure Button10Click(Sender: TObject);
73:
74:     private
75:     { Private declarations }
76:     public
77:     { Public declarations }
78:     end;
79:
80: var
81:     Form5: TForm5;
82:     AnaGAFlg:Boolean=True;
83:     inserisco5:Boolean=False;
84:     CanClose:Boolean=True;
85:     filtro,sg,tabl,garastr,giornostr:string;
86:     tabella:TDataset;
87:
88: implementation
89: uses
90:     Console1;
91:
92: {$R *.dfm}
93: {$D+}
94: procedure TForm5.FormCreate(Sender: TObject);    //Crea il Form5
95: begin
96:     Form2.ABSTable5.TableName:='AnaGa';
97:     Form2.ABSTable5.Open;
98:     Form2.ABSTable5.Edit;
99:     Form2.ABSTable6.TableName:='Base';
100:    Form2.ABSTable6.Open;
101:    Form2.ABSTable6.Edit;
102:    AnaGAFlg:=False;
103:    ds51.DataSet:=Form2.ABSTable5;//aggiunto per evitare circular DataLink error
104:    ds52.DataSet:=Form2.ABSTable4;
105:    Form2.ABSTable4.TableName:='AnaDspl';
106:    Form2.ABSTable4.Open;
107:    Form2.ABSTable4.Edit;
108:    Form2.ABSTable8.TableName:='GRPA';
109:    Form2.ABSTable8.Open;
110:    Form2.ABSTable8.Edit;
111:    Form2.ABSTable9.TableName:='GRPB';
112:    Form2.ABSTable9.Open;
113:    Form2.ABSTable9.Edit;
114:    shp1.Brush.Color:=clOlive; //per colorare la shape
115:    shp2.Brush.Color:=clRed; //per colorare la shape
116: end;
117:
118:
119:
120: procedure TForm5.Button1Click(Sender: TObject); //Esci
121: begin
122:     if inserisco5=True then //se ha inserito un record
123:         begin
124:             FormCloseQuery(Self,canclose);
125:         end
126:         else
127:             begin
128:                 Form5.Pulizia(Self);
129:                 Form2.ABSTable1.Open;
130:                 Form2.ABSTable1.Last;
131:                 if Form2.ABSTable1.FieldName('Password').AsString<>' ' then
132:                     Form2.ABSTable1.Insert;
133:                     Form2.ABSTable1.Edit;
134:                     Form2.ABSTable1.FieldName('OUTP').AsDateTime:=NOW;

```



```
136:         Form2.ABSTable1.Close;
137:     Form2.ABSTable4.Close;
138:     Form2.ABSTable5.Close;
139:     Form5.Close;    // NB chiama il can close
140:     AnaGAflg:=True; //per consentire di monitorizzare che Form5 è closed
141:     Form2.ABSTable6.Close;
142:     Form2.ABSTable8.Close;
143:     Form2.ABSTable9.Close;
144:     Form2.ABSTable10.Close;
145:     Form2.ABSTable10.EmptyTable;
146:     Form2.ABSTable11.Close;
147:     Form2.ABSTable11.EmptyTable;
148:     Form2.ABSTable12.Close;
149:     Form2.ABSTable12.EmptyTable;
150:     Form2.Visible:=True;
151: end;
152: end;
153:
154: procedure TForm5.Button2Click(Sender: TObject); //CERCA
155: label
156: vai;
157: var
158: chi:array[1..2]of shortint;
159: i:ShortInt;
160: begin
161:     ds51.DataSet:=ABSQuery6;
162:     i:=0;           //determina se data e/o luogo è in ricerca
163:     if edit1.text<>' ' then
164:         begin
165:             i:=i+1;
166:             chi[i]:=1;
167:         end;
168:     if edit2.Text<>' ' then
169:         begin
170:             i:=i+1;
171:             chi[i]:=2;
172:         end;
173:     if chi[2]>0 then //elabora i query
174:         begin
175:             filtro:='DATA LIKE ' + QuotedStr('%'+Edit1.Text+'%')+ ' AND LUOGO LIKE '
176:             +quotedStr('%'+edit2.text+'%');
177:             goto vai;
178:         end;
179:     if chi[1]>0 then
180:         begin
181:             if (edit1.Text<>' ') then
182:                 filtro:= 'DATA LIKE ' +quotedstr('%'+Edit1.Text+'%');
183:             if (edit2.Text<>' ') then
184:                 filtro:='LUOGO LIKE ' +QuotedStr('%'+Edit2.Text+'%');
185:             goto vai;
186:         end;
187:     vai :
188:         //effettua il query
189:         with ABSQuery6 do
190:             begin
191:                 close;
192:                 if i>0 then
193:                     begin
194:                         SQL.Text := 'select * from AnaGA where '+filtro; //Attenzione agli spazi prima e dopo
195:                         gli apici
196:                         ExecSql;
197:                     end else
198:                         begin
199:                             ShowMessage('Nessun elemento di ricerca inserito');
200:                             Exit;
201:                         end;
202:                         end;
```

```

203: end;
204:
205: procedure TForm5.Button3Click(Sender: TObject);    //Annulla
206: begin
207:     Edit1.Text:='';
208:     Edit2.Text:='';
209:     ds51.DataSet:=Form2.ABSTable5;
210: end;
211:
212:
213: procedure TForm5.Button4Click(Sender: TObject);    //AZZERAZIONE
214: begin
215:     if MessageDlg('Stai per azzerare l''intera Anagrafe delle Gare. Vuoi continuare ?',
216:         mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrYes then
217:         begin
218:             Form2.ABSTable5.Close;
219:             Form2.ABSTable5.EmptyTable;
220:             ShowMessage('Tutta l''anagrafe è stata cancellata !');
221:         end;
222: end;
223:
224:
225: procedure TForm5.cbb55Change(Sender: TObject); // Seleziona un dato della Combobox
226: begin                //effettuando una scelta viene definita la Table attiva
227:     if cbb55.Text='Base' then
228:         begin
229:             tabella:=Form2.ABSTable6;
230:             tabl:='Base';
231:             ds55.DataSet:=tabella;
232:             ShowMessage('Crea la Tabella Discipline Base'+#13+
233:                 'Clicca sul LUOGO della Gara'+#13 +'per inserire il suo codice.');// Tab 6
234:         end;
235:     if cbb55.Text='GruppoA' then
236:         begin
237:             tabella:=Form2.ABSTable8;
238:             tabl:='GRPA';
239:             ds55.DataSet := tabella;
240:             ShowMessage('Crea la Tabella Discipline Gruppo A ');//Tab 8
241:         end;
242:     if cbb55.Text='GruppoB' then
243:         begin
244:             tabella:=Form2.ABSTable9;
245:             tabl:='GRPB';
246:             ds55.DataSet := tabella; //in questo modo mostra gli accoppiamenti gia esistenti
247:             ShowMessage('Crea la Tabella Discipline Gruppo B ');//Tab 9
248:         end;
249: end;
250:
251:     //Accoppiamento GARA - DISCIPLIN
252:     // Inserimento codice Gara
253: procedure TForm5.dbgrd51CellClick(Column: TColumn);
254: Var
255:     CoGaSi:Boolean;
256: begin
257:     Self.ActiveControl := nil;    //per togliere eventuali focus precedenti
258:     if cbb55.ItemIndex>-1 then
259:         begin
260:             sg:= Form2.ABSTable5.FieldName('CODGA').AsString;
261:             //verifica l'esistenza di un precedente accoppiamento per quella Gara
262:             ds55.DataSet:=ABSQuery6;    //nella tabella degli accoppiamenti (TA)
263:             with ABSQuery6 do
264:                 begin
265:                     Close; //inserire una variabile nel query
266:                     SQL.Text := 'Select * from '+tabl+' where CODGA = '+' QuotedStr(sg);
267:                     ExecSql;
268:                     Open;
269:                 end;
270:
271: You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)

```

```

271:         if MessageDlg('Hai già elaborato questa Gara '+#13+'Vuoi aggiungere discipline ?',
272:             mtconfirmation, [mbYes,mbNo], 0) = mrNo then
273:             begin
274:                 if MessageDlg('Vuoi Rifarla del tutto ?',mtconfirmation, [mbYes,mbNo], 0) = mrYe
s then
275:                     begin
276:                         Form2.ABSTable6.Edit;
277:                         with Form2.ABSTable6 do // nel Query c'è già l'elenco degli accoppiamenti di
quella Gara
278:                             begin
279:                                 First;
280:                                 while(not Eof)do
281:                                     begin //seleziona e cancella
282:                                         CoGaSi:= Form2.ABSTable6.Locate('CODGA',sg,[1]);
283:                                         if CoGaSi=True then Delete;
284:                                         Next;
285:                                     end;
286:                                 end;
287:                             end;
288:                             ds55.DataSet:= tabella;
289:                             Exit;
290:                         end; //showmessage con posizione
291:                         ShowMessagePos('Questi sono gli accoppiamenti già inseriti',650,250);
292:
293:                         ds55.DataSet:= tabella; //ritorna al dataset primario
294:                         if MessageDlg('Codice Gara = '+sg+' ?', mtconfirmation, [mbYes,mbNo], 0) = mrYes then
295:                             begin
296:                                 tabella.Insert;
297:                                 tabella.FieldName('CODGA').AsInteger:=strToInt(sg);
298:                                 tabella.Post;
299:                                 showmessage('Clicca sulla DESCRIZIONE della disciplina da accoppiare');
300:                             end else
301:                                 begin
302:                                     sg:='';
303:                                     Exit ;
304:                                 end;
305:                             end
306:                         else
307:                             if inserisco5=False then
308:                                 ShowMessage('Effettua prima una scelta di Gruppo');
309:                             end;
310:
311: procedure TForm5.dbgrd52MouseUp(Sender: TObject; Button: TMouseButton;
312:     Shift: TShiftState; X, Y: Integer); //Accoppiamento GARA - DISCIPLINE
313:     var //Inserimento codice DISCIPLINE
314:         s:string;
315:     begin
316:         if cbb55.ItemIndex>-1 then
317:             begin
318:                 if sg='' then
319:                     begin //per definire su quale gara iserire le DSPL
320:                         showmessage('Clicca prima sul luogo della GARA');
321:                         Exit;
322:                     end;
323:                     s:= Form2.ABSTable4.FieldName('CODDSPL').AsString;
324:                     if MessageDlg('Codice Disciplina = '+s+' ?' , mtconfirmation,
325:                         [mbYes,mbNo], 0) = mrYes then
326:                         begin
327:                             //verifica che non sia già inserito
328:                             ds55.DataSet:=ABSQuery6;
329:                             with ABSQuery6 do
330:                                 begin
331:                                     Close; //inserire una variabile nel query
332:                                     SQL.Text := 'Select * from '+tabl+' where CODGA = '+ QuotedStr(sg)+
333:                                     ' and CODDSPL = '+ QuotedStr(s);
334:                                     ExecSql;
335:                                     Open;

```

```

337:         if ABSQuery6.RecordCount>0 then //caso in cui c'è già la DSPL in TA
338:             begin
339:                 ShowMessage('ERRORE - Questa disciplina è già inserita');
340:                 ds55.DataSet:=tabella; //riporta il dbgrid a leggere i valori
341:                 s:='';
342:                 Exit;
343:             end;
344:         ds55.DataSet:=tabella;
345:         tabella.Edit;
346:         tabella.FieldName('CODDSPL').AsInteger:=strToInt(s);
347:         tabella.Post;
348:         s:='';
349:         if MessageDlg('Vuoi inserire altre Discipline'+#13+'nella stessa Gara ?' ,
350:             mtconfirmation, [mbYes,mbNo], 0) = mrYes then
351:             begin
352:                 tabella.Insert;
353:                 tabella.FieldName('CODGA').AsInteger:=strToInt(sg);
354:                 tabella.Post;
355:                 showmessage('Clicca sulla DESCRIZIONE della disciplina da accoppiare');
356:             end else
357:             begin
358:                 sg:=''; //azzerà il codice gara precedente
359:                 s:='';
360:                 ds55.DataSet:=tabella;
361:                 Exit;
362:             end;
363:         end
364:         else Exit;
365:
366:     end
367:     else ShowMessage('Effettua prima una scelta di tabella');
368: end;
369:
370: procedure TForm5.dbgrd55MouseDown(Sender: TObject; Button: TMouseButton;
371:     Shift: TShiftState; X, Y: Integer); //verifica la scelta
372: begin
373:     if cbb55.ItemIndex=-1 then
374:         ShowMessage('Effettua prima una scelta di tabella');
375:     end;
376:
377: procedure TForm5.dbnvgr51Click(Sender: TObject; Button: TNavigateBtn);
378: begin //intercetta il dbnavigator
379:     case button of
380:         nbInsert: //se inserisce un nuovo record
381:             begin
382:                 Form2.ABSTable5.Last;
383:                 Form2.ABSTable5.Append;
384:                 inserisco5:=True; //inserisce il flag nel nuovo record
385:                 dbgrd51.SetFocus;
386:                 DBGrd51.SelectedField := Form2.ABSTable5.FieldName('DATA');
387:             end ;
388:         nbPost:
389:             begin //se lo carica nella tabella
390:                 inserisco5:=False ;
391:             end;
392:         nbDelete: //se cancella il record - serve anche per evitare di entrare in Query
393:             begin
394:                 inserisco5:=False;
395:             end;
396:     end;
397: end;
398:
399:
400: procedure TForm5.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
401: begin //Chiusura del Form5 Gara
402:     if inserisco5=True then
403:         if MessageDlg('Hai un dato da salvare ?' , mtconfirmation, [mbYes,mbNo], 0) = mrYes
404:             You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)

```

```

405:     begin
406:         CanClose:=False;
407:         Exit;
408:     end
409: else
410:     begin
411:         CanClose:=True;
412:         if inserisco5=True then //conferma il Falso di inserisco5
413:             inserisco5:=False;
414:             Button1Click(Self);
415:         end;
416: end;
417:
418:
419:
420: procedure TForm5.Edit3Click(Sender: TObject); //Acquisisce la gara per il report
421: begin                                // ripristina le tabelle
422:     Edit3.Text:='';
423:     Form2.ABSTable10.Close;
424:     Form2.ABSTable10.EmptyTable;
425:     Form2.ABSTable11.Close;
426:     Form2.ABSTable11.EmptyTable;
427:     Form2.ABSTable12.Close;
428:     Form2.ABSTable12.EmptyTable;
429:     Form2.ABSTable10.TableName:='repBase';
430:     Form2.ABSTable10.Open;
431:     Form2.ABSTable10.Edit;
432:     Form2.ABSTable11.TableName:='repGRPA';
433:     Form2.ABSTable11.Open;
434:     Form2.ABSTable11.Edit;
435:     Form2.ABSTable12.TableName:='repGRPB';
436:     Form2.ABSTable12.Open;
437:     Form2.ABSTable12.Edit;
438:
439: end;
440: procedure TForm5.Button5Click(Sender: TObject);    //prepara il report
441:     label
442:     GA,GB,FINE;
443:     var
444:     codgasi,dsplsi:Boolean;
445:     codgan,DSPLn:Integer;
446:     DSPLstr:string;
447: begin        //Riempire le 3 tabelle
448:     if Edit3.Text='' then
449:         begin
450:             ShowMessage('Devi inserire il codice della Gara');
451:             Exit;
452:         end else    //il codice GARA è presente
453:         begin      // Verifica l'esistenza della Gara negli accoppiamenti Base etc.
454:
455:             codgasi:= Form2.ABSTable5.Locate('CODGA',Edit3.text,[]);
456:             if codgasi=False then
457:                 begin
458:                     ShowMessage('Questo codice Gara è inesistente ');
459:                     Exit;
460:                 end else
461:                 begin        //Trovata la gara ed il giorno di esecuzione
462:                     codgan:=StrToInt(Edit3.Text);
463:                     garastr:= Form2.ABSTable5.FieldName('LUOGO').AsString;
464:                     giornostr := Form2.ABSTable5.FieldName('DATA').AsString;
465:                 end;
466:
467:                     //trovare le discipline della BASE
468:                     if Form2.ABSTable10.RecordCount>0 then //la tabella Base è già piena
469:                         begin
470:                             if MessageDlg('Il report delle discipline Base è già fatto. Vuoi rifarlo?' ,
471:                                 mtConfirmation, [mbYes,mbNo], 0) = mrYes then
472: You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)

```

```

473:         Form2.ABTable10.EmptyTable;
474:     end else goto GA;
475: end;
476: Form2.ABTable6.DisableControls;
477: try
478:     Form2.ABTable6.First;
479:     while not Form2.ABTable6.Eof do
480:         begin //per ogni codga ricavare il cod dspl e memorizzare la DESCR nella Tabella
a
481:             if codgan=Form2.ABTable6.FieldName('CODGA').AsInteger then
482:                 begin //se il codice gara corrisponde
483:                     DSPLn:=Form2.ABTable6.FieldName('CODDSPL').AsInteger;//memorizza la disc
iplina
484:                     dsplsi:=Form2.ABTable4.Locate('CODDSPL',DSPLn,[]);//cerca la disciplina su
ll'anagrafica
485:                     if dsplsi=True then //quando la trova
486:                         begin
487:                             DSPLstr:= Form2.ABTable4.FieldName('DESCR').AsString;//memorizza l
a descrizione
488:                             Form2.ABTable10.Insert; //la registra sulla tabella del report
489:                             Form2.ABTable10.FieldName('CODDSPL').AsInteger:=DSPLn;
490:                             Form2.ABTable10.FieldName('DESCR').AsString:=DSPLstr;
491:                             Form2.ABTable10.FieldName('TMAX').AsDateTime:=Form2.ABTable4.Fiel
dByName('TMAX').AsDateTime;
492:                             Form2.ABTable10.Post;
493:                         end;
494:                     end;
495:                     Form2.ABTable6.Next;
496:                 end;
497:             finally
498:                 Form2.ABTable6.EnableControls;
499:                 ShowMessage('Registrate le discipline Base');
500:             end;
501:
502:         GA :
503:             //trovare le discipline del GruppoA
504:         if Form2.ABTable11.RecordCount>0 then //è vuota la tabella GRPA ?
505:             begin
506:                 if MessageDlg('Il report delle discipline Gruppo A è già fatto. Vuoi rifarlo?' ,
507:                     mtconfirmation, [mbYes,mbNo], 0) = mrYes then
508:                     begin
509:                         Form2.ABTable11.EmptyTable;
510:                     end else goto GB;
511:                 end;
512:
513:             Form2.ABTable8.DisableControls;
514:             try
515:                 Form2.ABTable8.First;
516:                 while not Form2.ABTable8.Eof do
517:                     begin //per ogni codga ricavare il cod dspl e memorizzare la DESCR nella Tabella
518:                         if codgan=Form2.ABTable8.FieldName('CODGA').AsInteger then
519:                             begin //se il codice gara corrisponde
520:                                 DSPLn:=Form2.ABTable8.FieldName('CODDSPL').AsInteger;//memorizza la disc
iplina
521:                                 dsplsi:=Form2.ABTable4.Locate('CODDSPL',DSPLn,[]);//cerca la disciplina su
ll'anagrafica
522:                                 if dsplsi=True then //quando la trova
523:                                     begin
524:                                         DSPLstr:= Form2.ABTable4.FieldName('DESCR').AsString;//memorizza l
a descrizione
525:                                         Form2.ABTable11.Insert; //la registra sulla tabella del report
526:                                         Form2.ABTable11.FieldName('CODDSPL').AsInteger:=DSPLn;
527:                                         Form2.ABTable11.FieldName('DESCR').AsString:=DSPLstr;
528:                                         Form2.ABTable11.FieldName('TMAX').AsDateTime:=Form2.ABTable4.Fiel
dByName('TMAX').AsDateTime;
529:                                         Form2.ABTable11.Post;
530:                                     end;

```

```

532:         Form2.ABTable8.Next;
533:     end;
534: finally
535:     Form2.ABTable8.EnableControls;
536:     ShowMessage('Registrate le discipline Gruppo A');
537: end;
538:
539: GB :
540:
541: if Form2.ABTable12.RecordCount>0 then //è vuota la tabella Base ?
542: begin
543:     if MessageDlg('Il report delle discipline Gruppo B è già fatto. Vuoi rifarlo?' ,
544:         mtconfirmation, [mbYes,mbNo], 0) = mrYes then
545:         begin
546:             Form2.ABTable12.EmptyTable;
547:         end else goto FINE;
548:     end;
549:
550: Form2.ABTable9.DisableControls;
551: try
552:     Form2.ABTable9.First;
553:     while not Form2.ABTable9.Eof do
554:         begin //per ogni codga ricavare il cod dspl e memorizzare la DESCR nella Tabella
555:             if codgan=Form2.ABTable9.FieldName('CODGA').AsInteger then
556:                 begin //se il codice gara corrisponde
557:                     DSPLn:=Form2.ABTable9.FieldName('CODDSPL').AsInteger;//memorizza la disc
iplina
558:                     dsplsi:=Form2.ABTable4.Locate('CODDSPL',DSPLn,[]);//cerca la disciplina su
559:                     if dsplsi=True then //quando la trova
560:                         begin
561:                             DSPLstr:= Form2.ABTable4.FieldName('DESCR').AsString;//memorizza l
a descrizione
562:                             Form2.ABTable12.Insert; //la registra sulla tabella del report
563:                             Form2.ABTable12.FieldName('CODDSPL').AsInteger:=DSPLn;
564:                             Form2.ABTable12.FieldName('DESCR').AsString:=DSPLstr;
565:                             Form2.ABTable12.FieldName('TMAX').AsDateTime:=Form2.ABTable4.Fiel
dByName('TMAX').AsDateTime;
566:                             Form2.ABTable12.Post;
567:                         end;
568:                     end;
569:                     Form2.ABTable9.Next;
570:                 end;
571:             finally
572:                 Form2.ABTable9.EnableControls;
573:                 ShowMessage('Registrate le discipline Gruppo B');
574:             end;
575:
576:         FINE :
577:     end;
578: // Presentare il report  Nota bene come si passano le variabili
579: Form5.frxReport5.Variables['mia1'] := QuotedStr(giornostr);
580: Form5.frxReport5.Variables['mia2'] := QuotedStr(garastr);
581: // Form5.frxPDFExport51.FileName:='Accoppiamenti';
582: Form5.frxReport5.ShowReport;
583: Edit3.Text:='';
584: end;
585:
586: procedure TForm5.Button6Click(Sender: TObject);
587: begin
588:     Form5.frxReport5.Print;
589: end;
590:
591: // ELENCA o STAMPA L'anagrafe Gare
592: procedure TForm5.Button7Click(Sender: TObject);
593: begin
594:     Form5.frxReport51.ShowReport;

```



```

596:     then Exit else
597:         Form5.frxReport51.Print;
598: end;
599:
600: procedure TForm5.Button8Click(Sender: TObject); //Azzera gruppo Base
601: begin
602:     if MessageDlg('Stai per azzerare l''intera Anagrafe del Gruppo Base. Vuoi continuare ?',
603:         mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrYes then
604:         begin
605:             Form2.ABSTable6.Close;
606:             Form2.ABSTable6.EmptyTable;
607:             ShowMessage('Tutta l''anagrafe è stata cancellata !');
608:         end;
609: end;
610:
611: procedure TForm5.Button9Click(Sender: TObject); // Azzera gruppo A
612: begin
613:     if MessageDlg('Stai per azzerare l''intera Anagrafe del Gruppo A. Vuoi continuare ?',
614:         mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrYes then
615:         begin
616:             Form2.ABSTable8.Close;
617:             Form2.ABSTable8.EmptyTable;
618:             ShowMessage('Tutta l''anagrafe è stata cancellata !');
619:         end;
620: end;
621:
622: procedure TForm5.Button10Click(Sender: TObject); //Azzera gruppo B
623: begin
624:     if MessageDlg('Stai per azzerare l''intera Anagrafe del Gruppo B. Vuoi continuare ?',
625:         mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrYes then
626:         begin
627:             Form2.ABSTable9.Close;
628:             Form2.ABSTable9.EmptyTable;
629:             ShowMessage('Tutta l''anagrafe è stata cancellata !');
630:         end;
631: end;
632:
633: //Pulizia dei record vuoti
634: procedure TForm5.Pulizia(Sender:TObject); //Quando crei una Procedura premetti il Form
635: var
636: trovato,rimosso:Boolean ;
637: begin
638:     trovato:=False;
639:     rimosso:=False;
640:     trovato:=Form2.ABSTable6.Locate('CODDSPL',null,[]); //senza LocateOptions
641:     if trovato=True then
642:         begin
643:             Form2.ABSTable6.Delete;
644:             rimosso:=True;
645:             trovato:=False;
646:         end;
647:     trovato:=Form2.ABSTable6.Locate('CODGA',null,[]); //senza LocateOptions
648:     if trovato=True then
649:         begin
650:             Form2.ABSTable6.Delete;
651:             rimosso:=True;
652:             trovato:=False;
653:         end;
654:
655:     trovato:=Form2.ABSTable8.Locate('CODDSPL',null,[]); //senza LocateOptions
656:     if trovato=True then
657:         begin
658:             Form2.ABSTable8.Delete;
659:             rimosso:=True;
660:             trovato:=False;
661:         end;
662:     trovato:=Form2.ABSTable8.Locate('CODGA',null,[]); //senza LocateOptions
663: 
```



```
664:   begin
665:       Form2.ABSTable8.Delete;
666:       rimosso:=True;
667:       trovato:=False;
668:   end;
669:
670:   trovato:=Form2.ABSTable9.Locate('CODDSPL',null,[]); //senza LocateOptions
671:   if trovato=True then
672:       begin
673:           Form2.ABSTable9.Delete;
674:           rimosso:=True;
675:           trovato:=False;
676:       end;
677:       trovato:=Form2.ABSTable9.Locate('CODGA',null,[]); //senza LocateOptions
678:   if trovato=True then
679:       begin
680:           Form2.ABSTable9.Delete;
681:           rimosso:=True;
682:           trovato:=False;
683:       end;
684:   if rimosso=True then
685:       begin
686:           ShowMessage('Rimosso ogni record incompleto' );
687:           rimosso:=False;
688:       end;
689:   end;
690: end.
691:
```

```

1: unit Login1;    //03/04/2015
2:
3: interface
4:
5: uses
6:   Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, System.UITypes, Vcl.Graphics,
7:   Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Data.DB, ABSMain, System.Types;
8: type
9:   TForm1 = class(TForm)
10:    Edit1: TEdit;
11:    Label1: TLabel;
12:    procedure FormCreate(Sender: TObject);
13:    procedure Edit1KeyPress(Sender: TObject; var Key: Char);
14:    procedure FormClose(Sender: TObject; var Action: TCloseAction);
15:
16:    private
17:      { Private declarations }
18:    public
19:
20:      //class function Execute: Boolean;
21:    end;
22:
23: var
24:   Form1: TForm1;
25:   dir, pwstr, cnvstr, pwtemp: string;           //pwstr è la password in ingresso: cnvstr è la pw con
vertita
26:   nuovapw: Boolean = false;
27:   cicla: boolean = False;
28:   // ATTENZIONE VALORE INSERITO AL DEPLOY DEL PROGRAMMA è identico alla pw del Database
29:   Jolly: string='Corbi77'; //PW dell'amministratore usata anche in Console1
30: implementation
31:   uses
32:     Console1 ;
33:   {$R *.dfm}
34:   {$D+}
35:   procedure TForm1.FormCreate(Sender: TObject);    //preliminare login
36:   begin //verifica se c'è il db
37:     dir:=GetCurrentDir+'\'';
38:     pwstr:='';
39:     cnvstr:='';
40:     pwtemp:='';
41:     if FileExists(dir+'corbi.ABS')=False then //se non c'è il database
42:       begin
43:         MessageDlg('ATTENZIONE - File Dati corrotto o inesistente'+#13+' Il programma verrà chiu
so ', mtError, [mbOK], 0 );
44:         Application.Terminate;
45:       end else
46:         begin // se c'è il database
47:           Form2.ABSTable1.TableName:='Servizio';
48:           Form2.ABSTable1.Open;
49:           Form2.ABSTable1.First;
50:           Form2.ABSTable1.Edit;
51:           if (Form2.ABSTable1.FieldName('Password').IsNull) then //se non c'è ancora la pw CR
EARE LA PW
52:             begin
53:               ShowMessage('Assegnate una password: max 8 caratteri ');
54:               nuovapw:=True; //flag per nuova pw
55:               form1.Visible:=True;
56:             end
57:             else form1.Visible:=True; // continua se c'è la pw con la routine di riconoscimento
58:           end;
59:       end;
60:
61:   procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);    // carico login
62:   var
63:     n: Integer;
64:   You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)

```

```

65: riconosce:string;
66: begin
67:   if Key=#13 then // all'atto dell'entry della pw
68:     begin
69:       pwstr:=Edit1.Text; // pwstr è la password attualmente digitata
70:       if pwstr=Jolly then //Procedura per ripristinare la pw
71:         begin
72:           ShowMessage('ATTENZIONE - Ripristino password '+#13+'Il programma verrà chiuso, rilancialo e'+#13
73:             +'Digita una nuova password');
74:           Form2.ABSTable1.First; //Attenzione sembra che porre xx.first dopo edit tolga dalla
75:             condizione di edit
76:           Form2.ABSTable1.Edit;
77:           Form2.ABSTable1.FieldName('password').AsString:=''; //cancella la vecchia password
78:             per ripristinarla
79:           Form2.ABSTable1.Post;
80:           Application.Terminate;
81:         end;
82:       L:= Length(pwstr);
83:       if L>8 then //verifica la lunghezza massima
84:         begin
85:           ShowMessage('Numero di caratteri superiore ad 8, ripeti');
86:           Edit1.Text:='';
87:           Exit;
88:         end;
89:       if (nuovapw=True) and (cicla=False)then // Se è nuova password
90:         begin
91:           cnvstr:='';
92:           pwtemp:=pwstr; //si pone la pw nella memoria temporanea
93:           Edit1.Text:=''; //verifica pw
94:           ShowMessage('Conferma la password inserita');
95:           cicla:=True;
96:           Exit;
97:         end;
98:       if (nuovapw=True) and (cicla=True) then //sempre la nuova nel secondo ciclo
99:         begin
100:          if pwtemp=pwstr then
101:            begin
102:              for n := 1 to L do //procedura di cryptaggio
103:                begin
104:                  p:= chr(ord(pwstr[n])+(L+1-n));
105:                  cnvstr:=cnvstr + p;
106:                end;
107:              Form2.ABSTable1.Open;
108:              if Form2.ABSTable1.RecordCount=0 then
109:                Form2.ABSTable1.Insert else
110:                Form2.ABSTable1.First;
111:                Form2.ABSTable1.Edit;
112:                Form2.ABSTable1.FieldName('password').AsString:=cnvstr;
113:                Form2.ABSTable1.Post;
114:                Form2.ABSTable1.Close;
115:                form1.Visible:=False;
116:                cicla:=False;
117:                pwtemp:='';
118:                pwstr:='';
119:                nuovapw:=False;
120:                ShowMessage('La password è stata regolarmente registrata'+#13+'Memorizzala in luogo sicuro');
121:                Form1.Visible:=False;
122:                Form1.Close;
123:                Form2.Visible:=True;
124:              end else
125:                begin
126:                  MessageDlg('ATTENZIONE - I due inserimenti non coincidono'+#13+'Ripeti con attenzione', mtError,[mbOK],0 );
127:                  Edit1.Text:='';
128:                  cicla:=False;

```

```
128:         pwstr:='';
129:         Exit;
130:     end;
131: end
132: else //la pw non è nuova quindi PROCEDURA DI RICONOSCIMENTO
133: begin
134:     for n := 1 to L do //procedura di cryptaggio
135:     begin
136:         p:= chr(ord(pwstr[n])+(L+1-n));
137:         cnvstr:=cnvstr + p;
138:     end;
139:     Form2.ABSTable1.Open;
140:     Form2.ABSTable1.First;
141:     riconosce:=Form2.ABSTable1.FieldName('password').AsString;
142:     if cnvstr=ricosce then
143:     begin
144:         ShowMessage('Accesso consentito') ; // ACCESSO
145:         Form2.ABSTable1.Open;
146:         Form2.ABSTable1.INSERT;
147:         Form2.ABSTable1.FieldName('IN').AsDateTime:= NOW;
148:         Form2.ABSTable1.FieldName('what').AsString:='Accesso alla Consolle';
149:         Form2.ABSTable1.FieldName('OUT').AsString:='';
150:         Form2.ABSTable1.Post;
151:         Form2.ABSTable1.Close;
152:         Form1.Visible:=False ;
153:         Form1.Close;
154:         Form2.Visible:=True;
155:     end
156: else //La password inserita è sbagliata
157: begin
158:     if cicla=True then //ha sbagliato per la seconda volta
159:     begin
160:         if MessageDlg('Password errata ! Vuoi ripristinarla ?',
161:             mtConfirmation, [mbYes, mbNo], 0, mbYes) = mrYes then
162:         begin
163:             ShowMessage('Digita la password originale di registrazione');
164:             Edit1.Text:='';
165:             Exit;
166:         end else Application.terminate;
167:     end
168: else
169:     begin
170:         MessageDlg('ATTENZIONE - Password errata', mtError,[mbOK],0 );
171:         pwstr:='';
172:         cnvstr:='';
173:         edit1.text:='';
174:         cicla:=True ;
175:         Exit;
176:     end;
177: end;
178:
179: end;
180: end;
181: end;
182:
183: procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
184: begin
185:     Action:=caFree;
186: end;
187:
188: end.
189:
```

```
1: unit Gara;
2:
3: interface
4:
5: uses
6:   Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
7:   Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Mask, Vcl.DBCtrls,
8:   Vcl.ExtCtrls, Vcl.Grids, Vcl.DBGrids, Data.DB, ABSMain, Vcl.ComCtrls,
9:   Vcl.Imaging.jpeg, Vcl.Buttons, frxClass, frxDBSet, frxExportPDF,
10:  Datasnap.Provider, Datasnap.DBClient;
11:
12: type
13:   TForm6 = class(TForm)
14:     Button1: TButton;
15:     Button2: TButton;
16:     Button3: TButton;
17:     Button4: TButton;
18:     dbgrd72: TDBGrid;
19:     dbnvgr7: TDBNavigator;
20:     ds7: TDataSource;
21:     ds72: TDataSource;
22:     AbsQuery72: TABSQuery;
23:     cbb72: TComboBox;
24:     Label7: TLabel;
25:     Timer1: TTimer;
26:     Label8: TLabel;
27:     shp1: TShape;
28:     Label9: TLabel;
29:     dbgrd7: TDBGrid;
30:     dbgrd74: TDBGrid;
31:     ds74: TDataSource;
32:     Label1: TLabel;
33:     Edit1: TEdit;
34:     Edit2: TEdit;
35:     ABSQuery7: TABSQuery;
36:     Button5: TButton;
37:     ABSQuery74: TABSQuery;
38:     img2: TImage;
39:     cbb74: TComboBox;
40:     Label2: TLabel;
41:     rb1: TRadioButton;
42:     rb2: TRadioButton;
43:     btn1: TBitBtn;
44:     img3: TImage;
45:     Label3: TLabel;
46:     Label4: TLabel;
47:     frxdbdtst13: TfrxDBDataset;
48:     ABSQuery73: TABSQuery;
49:     ABSTable13: TABSTable;
50:     ds13: TDataSource;
51:     frxrprt7: TfrxReport;
52:     frxPDFExport7: TfrxPDFExport;
53:     Button6: TButton;
54:     Edit3: TEdit;
55:     Button7: TButton;
56:     ABSTable14: TABSTable;
57:     Button8: TButton;
58:     Bevel1: TBevel;
59:     Label5: TLabel;
60:     Label6: TLabel;
61:     Button9: TButton;
62:     img1: TImage;
63:     Edit4: TEdit;
64:     Label10: TLabel;
65:     Button10: TButton;
66:     frxReport14: TfrxReport;
```

```

68:     ABSQueryDUP: TABSQuery;
69:     ABSQueryDUP2: TABSQuery;
70:     procedure Button1Click(Sender: TObject);
71:     procedure FormCreate(Sender: TObject);
72:     procedure dbgrd72CellClick(Column: TColumn);
73:     procedure Button2Click(Sender: TObject);
74:     procedure dbnvgr7Click(Sender: TObject; Button: TNavigateBtn);
75:     procedure cbb72Select(Sender: TObject);
76:     procedure Timer1Timer(Sender: TObject);
77:     procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
78:     procedure Button5Click(Sender: TObject);
79:     procedure cbb74Change(Sender: TObject);
80:     procedure rb2Click(Sender: TObject);
81:     procedure rb1Click(Sender: TObject);
82:     procedure Button6Click(Sender: TObject);
83:     procedure btn1Click(Sender: TObject);
84:     procedure Edit1Change(Sender: TObject);
85:     procedure Button3Click(Sender: TObject);
86:     procedure Button4Click(Sender: TObject);
87:     procedure Button7Click(Sender: TObject);
88:     procedure Button8Click(Sender: TObject);
89:     procedure ABSTable14BeforeInsert(DataSet: TDataSet);
90:     procedure ABSTable14BeforeDelete(DataSet: TDataSet);
91:     procedure Button9Click(Sender: TObject);
92:     procedure dbgrd7CellClick(Column: TColumn);
93:     procedure GetText(Sender: TField; var Text: string; DisplayText: Boolean);
94:     procedure Button10Click(Sender: TObject);
95:     private
96:     { Private declarations }
97:     public
98:     { Public declarations }
99:     end;
100:         //NOTA BENE l'entry del codice gara non triggera alcun evento
101:     var
102:         Form6: TForm6;
103:         GaFlg:Boolean =True;
104:         inserisco7:Boolean=False;
105:         ATL:Integer;
106:         scelta:Boolean=False; // se si sceglie con la combobox > True
107:         filtGa:Boolean=False; // idem se filtra Gara
108:         CanClose:Boolean=True;
109:         LastGa:SmallInt;
110:         numgara:Integer;
111:         tempo:string;
112:         rab1,rab2:Boolean;
113:     implementation
114:         uses
115:             Console1,aiuto,Tempi ;
116:         {$R *.dfm}
117:         {$D+}
118:     procedure TForm6.FormCreate(Sender: TObject); //Crea il form
119:     begin
120:         Form2.ABSTable7.TableName:='Gara';
121:         Form2.ABSTable7.Open; //FORMATTARE CAMPI DBGRID
122:         TDateTimeField(Form2.ABSTable7.FieldByName('TAME1')).DisplayFormat:='nn:ss,zzz';
123:         TDateTimeField(Form2.ABSTable7.FieldByName('TBME2')).DisplayFormat:='nn:ss,zzz';
124:         TDateTimeField(Form2.ABSTable7.FieldByName('PEN')).DisplayFormat:='nn:ss,zzz';
125:         TBooleanField(Form2.ABSTable7.FieldByName('DQ')).DisplayValues:='Yes;No';
126:         ABSTable14.TableName:='Corrente';
127:         ABSTable14.Open;
128:         TDateTimeField(ABSTable14.FieldByName('TAME1')).DisplayFormat:='nn:ss,zzz';
129:         TDateTimeField(ABSTable14.FieldByName('TBME2')).DisplayFormat:='nn:ss,zzz';
130:         TDateTimeField(ABSTable14.FieldByName('PEN')).DisplayFormat:='nn:ss,zzz';
131:         TBooleanField(ABSTable14.FieldByName('DQ')).DisplayValues:='Yes;No';
132:         GaFlg:=False;
133:         ds7.DataSet:=Form2.ABSTable7;//aggiunto per evitare circular DataLink error
134:         Form2.ABSTable7.Edit;

```

```

136:     Form2.ABSTable1.Open;
137:     Form2.ABSTable1.INSERT;
138:     Form2.ABSTable1.FieldName('IN').AsDateTime:= NOW;
139:     Form2.ABSTable1.FieldName('what').AsString:='Accesso alle Iscrizioni';
140:     Form2.ABSTable1.FieldName('OUT').AsString:='';
141:     Form2.ABSTable1.Post;
142:     Form2.ABSTable1.Close;
143:     cbb74.Text:='Scegli';
144:     if rb1.Checked=True then
145:     begin
146:         Edit2.Visible:=False;
147:         Label1.Visible:=False;
148:     end;
149:     Form2.ABSTable5.TableName:='AnaGa';
150:     Form2.ABSTable5.Open;
151:     Form2.ABSTable5.Last;
152:     LastGa:=Form2.ABSTable5.FieldName('CODGA').AsInteger ;
153:     Label4.Caption:=IntToStr(LastGa);
154:     Button2.Click;    //Refresh
155:     Button5.Click;
156:     ATL:=0;
157:
158: end;
159:
160: procedure TForm6.GetText(Sender: TField; var Text: string; DisplayText: Boolean);
161: begin
162:     if ABSTable14.FieldName('DQ').AsBoolean then
163:         Text := 'Yes'
164:     else
165:         Text := 'No';
166:
167: end;
168:
169:
170:
171: procedure TForm6.rb1Click(Sender: TObject); //AUTOMATICA
172: begin
173:     Edit2.Visible:=False;    //toglie edit2
174:     Label1.Visible:=False;
175:     cbb74.Visible:=True;
176:     Label2.Visible:=True;
177: end;
178:
179: procedure TForm6.rb2Click(Sender: TObject); // MANUALE
180: begin
181:     if Edit1.Text='' then
182:     begin
183:         ShowMessage('Inserisci il Codice della Gara ');
184:         rb1.Checked:=True;
185:         Exit;
186:     end;
187:     cbb74.Visible:=False;
188:     Label2.Visible:=False;
189:     Edit2.Visible:=True;
190:     Label1.Visible:=True;
191:     ds74.DataSet:=Form2.ABSTable4;
192: end;
193:
194: procedure TForm6.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
195: begin
196:     if inserisco7=True then
197:     if MessageDlg('Hai un dato da salvare ?' , mtconfirmation, [mbYes,mbNo], 0) = mrYes
198:     then
199:     begin
200:         CanClose:=False;
201:         Exit;
202:     end

```

```

204:      begin
205:          CanClose:=True;
206:          if inserisco7=True then //conferma il Falso di inserisco5
207:              inserisco7:=False;
208:              Button1Click(Self);
209:      end;
210: end;
211:
212:
213: procedure TForm6.Timer1Timer(Sender: TObject);    //Flashing
214: begin
215:     if Edit1.Text='' then
216:         begin
217:             if label8.Font.Color = clRed then
218:                 label8.Font.Color :=  clNavy
219:             else
220:                 label8.Font.color := clRed;
221:             end else Label8.Visible:=False;
222:     if Edit2.Text='' then
223:         begin
224:             if label11.Font.Color = clRed then
225:                 label11.Font.Color :=  clNavy
226:             else
227:                 label11.Font.color := clRed;
228:             end else Label11.Visible:=False;
229:     //condizioni per garantire che label2 sia invisibile, altrimenti il colore vince
230:     if ((cbb74.ItemIndex=0) or (cbb74.Text='Scegli'))and (cbb74.Visible=True) then
231:         begin
232:             if label2.Font.Color = clRed then
233:                 label2.Font.Color :=  clNavy
234:             else
235:                 label2.Font.color := clRed;
236:                 Label2.Visible:=True;
237:             end else Label2.Visible:=False;
238:
239:     end;
240:
241: procedure TForm6.ABSTable14BeforeDelete(DataSet: TDataSet);
242: begin
243:     if ds7.DataSet=ABSTable14 then
244:         begin
245:             ShowMessage('Prima di rimuovere un record premi "Refresh" '+#13+
246:                 'ed identifica il record con il suo codice');
247:             Abort;
248:             Exit;
249:         end;
250:     end;
251:
252: procedure TForm6.ABSTable14BeforeInsert(DataSet: TDataSet);
253: begin
254:     if ds7.DataSet=ABSTable14 then
255:         begin
256:             ShowMessage('Per inserire nuovi record premi "Refresh" '+#13+
257:                 'o esegui l''inserimento automatico');
258:             Abort;
259:             Exit;
260:         end;
261:     end;
262:
263: procedure TForm6.btn1Click(Sender: TObject);
264: begin
265:     // Form7.Create(self);
266:     Form7.Show;
267: end;
268:
269: procedure TForm6.Button1Click(Sender: TObject);    //Esci
270:

```



```

272:
273:     if filtGa =True then
274:     if MessageDlg('Devi sincronizzare la tabella delle iscrizioni ?'
275: , mtconfirmation, [mbYes,mbNo], 0) = mrYes then Exit;
276:     filtGa:=False;
277:     if inserisco7=True then //se ha inserito un record
278:     begin
279:         FormCloseQuery(Self,canclose);
280:     end
281:     else
282:     begin
283:         Form2.ABSTable7.Close;
284:         Form6.Close; // NB chiama il can close
285:         GaFlg:=True; //per consentire di monitorizzare che Form6 è closed
286:         Form2.Visible:=True;
287:         Edit1.Text:=''; //rimuove il codice della gara
288:         Edit2.Text:=''; // '' della DSPL
289:         Form2.ABSTable10.Close; //Chiude non cancella
290:         Form2.ABSTable11.Close;
291:         Form2.ABSTable12.Close;
292:         Form6.ABSTable13.Close;
293:         Form6.ABSTable13.EmptyTable;
294:         Form6.ABSTable14.Close;
295:         Form6.ABSTable14.EmptyTable;
296:         Form2.ABSTable2.Close;
297:         Form2.ABSTable4.Close;
298:         Form2.ABSTable5.Close;
299:         Form2.ABSTable6.Close;
300:         Form2.ABSTable8.Close;
301:         Form2.ABSTable9.Close;
302:     end;
303:
304: end;
305:
306: procedure TForm6.Button2Click(Sender: TObject); //Refresh dbgrid Atleti
307: begin
308:     ds72.DataSet:=AbsQuery72;
309:     with AbsQuery72 do
310:     begin
311:         Close; //inserisce una variabile nel query
312:         SQL.Text := 'Select * from AnaAtl';
313:         ExecSql;
314:         Open;
315:     end;
316: end;
317:
318: procedure TForm6.Button3Click(Sender: TObject); //Rapporto
319: var
320: CODGAn:Integer;
321: DSPLsi,ATLsi:Boolean;
322: CODDSPL13,CODATL13:Integer;
323: begin
324:     Form2.ABSTable4.Open;
325:     //Form2.ABSTable4.Edit;
326:     Form2.ABSTable2.Open;
327:     //Form2.ABSTable2.Edit;
328:     CODGAn:=StrToInt(inputBox('Rapporto Iscrizioni','Inserire il codice della Gara',Edit1.Text));
329:     ds7.DataSet := ABSQuery7; //Costruisce una tabella SQL
330:     with ABSQuery7 do
331:     begin
332:         Close; //porto su DBGRID solo la gara scelta con tutti gli iscritti
333:         SQL.Text := 'Select * from Gara where CODGA = '+QuotedStr(IntToStr(CODGAn))+ '
ORDER BY CODATL Asc, CODDSPL Asc';
334:         ExecSql;
335:         Open; //Estrae i dati di iscrizione solo per una gara
336:         First.

```

```

338:         Form6.ABSTable13.Close;
339:         Form6.ABSTable13.EmptyTable;
340:         Form6.ABSTable13.Open;
341:         TBooleanField(ABSTable13.FieldByName('DQ')).DisplayValues:='Yes;No';
342:         TDateTimeField(ABSTable13.FieldByName('TAME1')).DisplayFormat:='nn:ss,zzz';
343:         TDateTimeField(ABSTable13.FieldByName('TBME2')).DisplayFormat:='nn:ss,zzz';
344:         TDateTimeField(ABSTable13.FieldByName('PEN')).DisplayFormat:='nn:ss,zzz';
345:         Form6.ABSTable13.Edit;           // Crea Tabella Master per il report = Ma
estra
346:         while (not Eof) do //sempre sotto ABSQuery7
347:         begin                // copia il Query in una Table
348:             Form6.ABSTable13.Insert;
349:             Form6.ABSTable13.FieldByName('CODGA').AsInteger:=ABSQuery7.FieldByName('CODG
A').AsInteger;
350:             Form6.ABSTable13.FieldByName('CODDSPL').AsInteger:=ABSQuery7.FieldByName('CO
DDSPL').AsInteger;
351:             Form6.ABSTable13.FieldByName('CODATL').AsInteger:=ABSQuery7.FieldByName('COD
ATL').AsInteger;
352:             Form6.ABSTable13.FieldByName('COGNOME').AsString:=ABSQuery7.FieldByName('COG
NOME').AsString;
353:             Form6.ABSTable13.FieldByName('NMPET').AsInteger:=ABSQuery7.FieldByName('NMPE
T').AsInteger;
354:             Form6.ABSTable13.FieldByName('NMPOS').AsInteger:=ABSQuery7.FieldByName('NMPO
S').AsInteger;
355:             Form6.ABSTable13.FieldByName('TAME1').AsDateTime:=ABSQuery7.FieldByName('TAM
E1').AsDateTime;
356:             Form6.ABSTable13.FieldByName('TBME2').AsDateTime:=ABSQuery7.FieldByName('TBM
E2').AsDateTime;
357:             Form6.ABSTable13.FieldByName('DQ').AsBoolean:=ABSQuery7.FieldByName('DQ').As
Boolean;
358:             Form6.ABSTable13.FieldByName('PEN').AsDateTime:=ABSQuery7.FieldByName('PEN')
.AsDateTime;
359:             Form6.ABSTable13.FieldByName('MANL').AsBoolean:=ABSQuery7.FieldByName('MANL')
.AsBoolean;
360:             Form6.ABSTable13.FieldByName('CAT').AsInteger:=ABSQuery7.FieldByName('CAT').
AsInteger;
361:             Form6.ABSTable13.Post;
362:             ABSQuery7.Next;
363:         end;
364:
365:         end;           //aggiungo i campi DESCR e TMAX
366:         ABSTable13.First;
367:         while not ABSTable13.EOF do
368:         begin
369:             CODDSPL13:=ABSTable13.FieldByName('CODDSPL').AsInteger;
370:             DSPLsi:=Form2.ABSTable4.Locate('CODDSPL',CODDSPL13,[ ] );//si blocca sul record con p
ari codice DSPL
371:             if DSPLsi=False then ShowMessage('Valore non trovato'); //NB nel locate il valore d
a ricercare è variant (non stringa)
372:             ABSTable13.Edit;
373:             ABSTable13.FieldByName('DESCR').AsString:=Form2.ABSTable4.FieldByName('DESCR').AsStr
ing;
374:             ABSTable13.FieldByName('TMAX').AsDateTime:=Form2.ABSTable4.FieldByName('TMAX').AsDat
eTime;
375:             ABSTable13.Post;
376:             ABSTable13.NEXT
377:         end;
378:         // Aggiungo il nome e nazione e il telefono
379:         ABSTable13.First;
380:         while not ABSTable13.EOF do
381:         begin
382:             CODATL13:=ABSTable13.FieldByName('CODATL').AsInteger;
383:             ATLsi:=Form2.ABSTable2.Locate('CODATL',CODATL13,[ ] );//si blocca sul record con pari
codice DSPL
384:             if ATLsi=False then ShowMessage('Valore non trovato');
// NB nel locate il valore da ricercare è variant (non stringa)
385:             ABSTable13.Edit;
386:             ABSTable13.FieldByName('NOME').AsString:=Form2.ABSTable2.FieldByName('NOME').AsStrin

```

```

g;
387:         ABSTable13.FieldName('NAZIONE').AsString:=Form2.ABSTable2.FieldName('NAZIONE').A
sString;
388:         ABSTable13.FieldName('TLF').AsString:=Form2.ABSTable2.FieldName('TLF').AsString;
389:         ABSTable13.Post;
390:         ABSTable13.NEXT
391:     end;
392:     Form6.frxrprt7.ShowReport;
393: end;
394:
395: procedure TForm6.Button4Click(Sender: TObject); //Stampa Report
396: begin
397:     Form6.frxrprt7.Print;
398: end;
399:
400: procedure TForm6.Button5Click(Sender: TObject); //Refresh tabella iscrizioni
401: begin
402:     dbgrd7.Columns[0].FieldName:='NMREC'; //applicare il campo relativo al cambio di Dataset
403:     Label6.Caption:='Dati relativi a tutte le Gare';
404:     ds7.DataSet:=Form2.ABSTable7;
405: end;
406:
407: procedure TForm6.Button6Click(Sender: TObject); // Filtro una gara          LAVORI IN CORSO
CODICE DA RIVEDERE
408: var
409:     quanti,btscelta :Integer;
410:     cosa:TStringList;
411:     ca,cd,ga:Integer;
412: begin
413:     if Edit3.Text='' then
414:     begin
415:         ShowMessage('Inserisci il codice della Gara nella casella a destra !');
416:         Exit;
417:     end;
418:     Form6.ABSTable14.TableName:=('Corrente');
419:     Form6.ABSTable14.Open;
420:     Form6.ABSTable14.Edit;
421:     quanti:=ABSTable14.RecordCount;
422:     filtGa:=True;
423:     if quanti>0 then //Continua solo se non esiste già il buffer
424:     begin
425:         numgara:=ABSTable14.FieldName('CODGA').AsInteger;
426:         if numgara<> StrToInt(Edit3.Text) then //numero gara residente DIVERSO dal richiest
o
427:         begin
428:             if MessageDlg('La tabella contiene dati per una Gara diversa '+#13#10+
429:                 'e con codice : '+ IntToStr(numgara)+' Desideri cancellarli ?', mtconfirmatio
n,
430:                 [mbYes,mbNo], 0) = mrYes then
431:                 begin
432:                     Button7Click(self);
433:                     Edit3.Text:='';
434:                     Exit;
435:                 end else
436:                 begin //Gara diversa da quella richiesta non cancellata
437:                     ShowMessage('Puoi esaminare una gara per volta, ripeti la scelta !');
438:                     Edit3.Text:='';
439:                     Exit;
440:                 end;
441:             end else
442:             begin
443:                 btscelta := MessageDlg('La Tabella per la Gara n. '+IntToStr(numgara)+' ha '+Int
ToStr(quanti)+
444:                 ' dati.'+#13+' Vuoi usarla(Yes) o cancellarla(No) o rinunciare(Cancel) ? ',mtCu
stom,
445:                 [mbYes,mbNo,mbCancel], 0);
446:                 if btscelta = mrYes then //usa il precedente buffer e continua

```

```

448:         dbgrd7.Columns[0].FieldName:='NM'; //applicare il campo relativo al cambio di
Dataset
449:         ds7.DataSet:=ABSTable14;
450:         Label6.Caption:='Dati relativi ad una Gara';
451:         ShowMessage('E'' possibile effettuare modifiche');
452:         Edit3.Text:=''; //Punto in cui si è scelta la GARA da Aggiornare
453:         Exit;
454:     end;
455:     if btscelta = mrNo then // cancella il buffer per rifarlo
456:     begin
457:         Button7Click(Self);
458:         Edit3.Text:='';
459:         exit;
460:     end;
461:     if btscelta = mrCancel then //esce dalla procedura
462:     begin
463:         Edit3.Text:='';
464:         Exit;
465:     end;
466: end;
467: end;
468:
469: dbgrd7.Columns[0].FieldName:='NM'; //applicare il campo relativo al cambio di Dataset
470: ds7.DataSet := ABSQuery7; //Costruisce una tabella SQL
471: with ABSQuery7 do
472:     begin
473:         Close; //porto su DBGRID solo la gara scelta con tutti gli iscritti
474:         SQL.Text := 'Select * from Gara where CODGA = '+QuotedStr(Edit3.Text)+ ' ORDER
BY CODATL Asc, CODDSPL Asc';
475:         ExecSql;
476:         Open; //Estrae i dati di iscrizione solo per una gara
477:         First;
478:         while (not Eof) do //sempre sotto ABSQuery7
479:         begin // copia il Query in una Table 14 Buff
er gara corrente
480:             Form6.ABSTable14.Insert;
481:             Form6.ABSTable14.FieldName('NM').AsInteger:=ABSQuery7.FieldName('NMREC')
.AsInteger;
482:             Form6.ABSTable14.FieldName('CODGA').AsInteger:=ABSQuery7.FieldName('CODG
A').AsInteger;
483:             Form6.ABSTable14.FieldName('CODDSPL').AsInteger:=ABSQuery7.FieldName('CO
DDSPL').AsInteger;
484:             Form6.ABSTable14.FieldName('CODATL').AsInteger:=ABSQuery7.FieldName('COD
ATL').AsInteger;
485:             Form6.ABSTable14.FieldName('COGNOME').AsString:=ABSQuery7.FieldName('COG
NOME').AsString;
486:             Form6.ABSTable14.FieldName('NMPET').AsInteger:=ABSQuery7.FieldName('NMPE
T').AsInteger;
487:             Form6.ABSTable14.FieldName('NMPOS').AsInteger:=ABSQuery7.FieldName('NMPO
S').AsInteger;
488:             Form6.ABSTable14.FieldName('TAME1').AsDateTime:=ABSQuery7.FieldName('TAM
E1').AsDateTime;
489:             Form6.ABSTable14.FieldName('TBME2').AsDateTime:=ABSQuery7.FieldName('TBM
E2').AsDateTime;
490:             Form6.ABSTable14.FieldName('DQ').AsBoolean:=ABSQuery7.FieldName('DQ').As
Boolean;
491:             Form6.ABSTable14.FieldName('PEN').AsDateTime:=ABSQuery7.FieldName('PEN')
.AsDateTime;
492:             Form6.ABSTable14.FieldName('MANL').AsBoolean:=ABSQuery7.FieldName('MANL')
.AsBoolean;
493:             Form6.ABSTable14.FieldName('CAT').AsInteger:=ABSQuery7.FieldName('CAT').
AsInteger;
494:             Form6.ABSTable14.Post;
495:             ABSQuery7.Next;
496:         end;
497:     end;
498:     Label6.Caption:='Dati relativi ad una Gara';

```

```

500:      //Controllo presenza dati essenziali
501:      cosa:=TStringList.Create;
502:      cosa.Clear;
503:      ABSTable14.First;
504:      while not ABSTable14.Eof do
505:      begin
506:          if ABSTable14.FieldName('CAT').AsInteger=0 then
507:          begin
508:              ga:=ABSTable14.FieldName('CODGA').AsInteger;
509:              ca:=ABSTable14.FieldName('CODATL').AsInteger;
510:              cd:=ABSTable14.FieldName('CODDSPL').AsInteger;
511:              cosa.Add('Gara : '+IntToStr(ga)+' Codice Atl. : '+IntToStr(ca)+'- Disciplina:
'+IntToStr(cd)+'> Categoria');
512:          end;
513:          if ABSTable14.FieldName('NMPET').AsInteger=0 then
514:          begin
515:              ga:=ABSTable14.FieldName('CODGA').AsInteger;
516:              ca:=ABSTable14.FieldName('CODATL').AsInteger;
517:              cd:=ABSTable14.FieldName('CODDSPL').AsInteger;
518:              cosa.Add('Gara : '+IntToStr(ga)+' Codice Atl. : '+IntToStr(ca)+'- Disciplina:
'+IntToStr(cd)+'> Pettorale');
519:          end;
520:          if ABSTable14.FieldName('NMPOS').AsInteger=0 then
521:          begin
522:              ga:=ABSTable14.FieldName('CODGA').AsInteger;
523:              ca:=ABSTable14.FieldName('CODATL').AsInteger;
524:              cd:=ABSTable14.FieldName('CODDSPL').AsInteger;
525:              cosa.Add('Gara : '+IntToStr(ga)+' Codice Atl. : '+IntToStr(ca)+'- Disciplina:
'+IntToStr(cd)+'> Posizione');
526:          end;
527:          ABSTable14.Next;
528:      end;
529:      if cosa.Count>0 then
530:      begin
531:          cosa.SaveToFile('.\mancanti.txt');
532:          if (cosa.Count<50) then
533:              ShowMessage('Codici Atleti con dati mancanti-Annotarli ' + #13+#13+ cosa.Text+#13+
534:              ' Dati salvati in "mancanti.txt") else
535:              ShowMessage('La lista degli Atleti con dati mancanti è salvata in "Mancanti.txt".
536:      ');
537:      //Correzione DATI Mancanti
538:      if MessageDlg('Desideri fornire ora i dati mancanti essenziali ?'+#13+#13
539:      + 'Ricorda di caricare il dato e sincronizzare',mtconfirmation, [mbYes,mbNo], 0)
540:      = mrYes then
541:      begin
542:          ds7.DataSet:=ABSQuery7;
543:          with ABSQuery7 do
544:          begin
545:              Close;
546:              SQL.Text := 'Select * from Corrente where CAT = '+QuotedStr(IntToStr(0)) ;
547:              ExecSql;
548:              Open;
549:          end;
550:      end;
551:
552:      ShowMessage('E' possibile effettuare modifiche');
553:      Edit3.Text:='';
554:      cosa.Clear;
555:  end;
556:
557:
558: procedure TForm6.Button7Click(Sender: TObject);    // Resetta la tabella della Gara corrente
559: begin
560:     if MessageDlg('Stai per azzerare i dati memorizzati nel Buffer. Vuoi continuare ?',
561:     mtConfirmation, [mbYes, mbNo], 0, mbNo) = mrYes then
562:     You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)

```

```

563:         ABSTable14.Close;
564:         ABSTable14.EmptyTable;
565:         ShowMessage('Tutti i dati di iscrizione contenuti nel buffer sono stati cancellat
i, '
566:             + #13+ ' i dati sincronizzati sono presenti nel Database');
567:     end;
568:     ds7.DataSet:=Form2.ABSTable7;
569: end;
570:
571: procedure TForm6.Button8Click(Sender: TObject);    //Sincronizza buffer con Gara
572: var
573: doppio:Boolean;
574: cerco:integer;
575: begin
576:     if filtGa=False then
577:     begin
578:         ShowMessage('Non c'è nulla da sincronizzare. ');
579:         Exit;
580:     end;
581:     if MessageDlg('Confermi di voler aggiornare i Dati Base ?', mtconfirmation,
582:         [mbYes,mbNo], 0) = mrNo then Exit;
583:
584:     Form2.ABSTable7.Open;
585:     ABSTable14.Open;
586:     ABSTable14.First;
587:     ABSTable14.Edit;
588:     while not ABSTable14.eof do
589:     begin
590:         cerco:=ABSTable14.FieldName('NM').AsInteger;
591:         doppio:=Form2.ABSTable7.Locate('NMREC',cerco,[]);
592:         if doppio=True then
593:         begin //record preesistente da aggiornare
594:             Form2.ABSTable7.Edit;
595:             Form2.ABSTable7.FieldName('NMPET').AsInteger:=ABSTable14.FieldName('NMPET').As
Integer;
596:             Form2.ABSTable7.FieldName('NMPOS').AsInteger:=ABSTable14.FieldName('NMPOS').As
Integer;
597:             Form2.ABSTable7.FieldName('TAME1').AsDateTime:=ABSTable14.FieldName('TAME1').A
sDateTime;
598:             Form2.ABSTable7.FieldName('TBME2').AsDateTime:=ABSTable14.FieldName('TBME2').A
sDateTime;
599:             Form2.ABSTable7.FieldName('DQ').AsBoolean:=ABSTable14.FieldName('DQ').AsBoolea
n;
600:             Form2.ABSTable7.FieldName('PEN').AsDateTime:=ABSTable14.FieldName('PEN').AsDat
eTime;
601:             Form2.ABSTable7.FieldName('MANL').AsBoolean:=ABSTable14.FieldName('MANL').AsBo
olean;
602:             Form2.ABSTable7.FieldName('CAT').AsInteger:=ABSTable14.FieldName('CAT').AsInte
ger;
603:             Form2.ABSTable7.Post;
604:         end else
605:         begin
606:             ShowMessage('ERRORE ! Il record con il codice '+IntToStr(cerco)+
607:                 ' non esiste nella Tabella Gare Generale. ');
608:         end;
609:         ABSTable14.Next;
610:     end;
611:     ShowMessage('La tabella Gare Generale è stata aggiornata.'+#13+
612:         'Puoi cliccare "Azzerà" per cancellare il buffer. ');
613: end;
614:
615: procedure TForm6.Button9Click(Sender: TObject);
616: begin
617:     if filtGa=False then
618:     begin
619:         ShowMessage('Non c'è alcuna Gara selezionata');
620:         Exit;

```



```

622:     Form9.Show;
623: end;
624:
625:
626: procedure TForm6.Button10Click(Sender: TObject); //Rapporto iscritti
627: begin
628:     if Edit3.Text='' then
629:     begin
630:         ShowMessage('Inserisci il codice della Gara nella casella a destra !');
631:         Exit;
632:     end;
633:
634:     frxReport14.ShowReport;
635:
636: end;
637:
638: //Inserimento atleti nella gara cliccando su un Atleta
639: procedure TForm6.dbgrd72CellClick(Column: TColumn);
640: var
641:     Nom,Cogn:string;
642:     volte:smallInt;
643:     Sorge:TABSTable;
644:     Rect2:Boolean;
645:     trovato:Boolean;
646:     n:Integer;
647: begin
648:     if Edit1.Text<>' ' then
649:     begin
650:         // o usa la Tab2=ATL o Query2 = ATL da cognome
651:         Form2.ABSTable2.Open;
652:         if scelta=False then
653:         begin
654:             ATL:= dbgrd72.Fields[0].AsInteger;//CODATL dal campo 0 della tabella di cui ho cliccat
o una colonna
655:             end else ATL:=AbsQuery72.FieldByName('CODATL').AsInteger; //prende ATL dalla scelta
656:
657:             if Form2.ABSTable2.RecordCount >0 then
658:                 Rect2:=Form2.ABSTable2.Locate('CODATL',ATL,[]) else
659:                 begin
660:                     ShowMessage('La tabella degli Atleti è vuota. ');
661:                     Exit;
662:                 end;
663:             if Rect2=False then
664:             begin
665:                 ShowMessage('Errore Imprevisto: l''Atleta non esiste nel DataBase - Controlla');
666:                 Exit;
667:             end;
668:             Cogn:= Form2.ABSTable2.FieldByName('COGNOME').AsString;
669:             Nom:= Form2.ABSTable2.FieldByName('NOME').AsString;
670:
671:             if rb1.Checked=True then //automatica
672:             begin
673:                 if MessageDlg('Confermi l''iscrizione automatica ?',
674:                     mtconfirmation, [mbYes,mbNo], 0) = mrYes then
675:                 begin
676:                     if cbb74.ItemIndex=-1 then //Att. cbb.Text='xx' funge dopo una scelta
677:                     begin
678:                         ShowMessage('Scegli il Gruppo di iscrizione');
679:                         Exit;
680:                     end;
681:                 end;
682:                 //SCELTA AUTOMATICA dopo la scelta della Gara e del Gruppo
683:                 if MessageDlg('Confermi l''iscrizione AUTOMATICA di '+Nom+ ' '+ Cogn+
684:                     ' alle discipline del gruppo '+cbb74.Text,
685:                     mtconfirmation, [mbYes,mbNo], 0) = mrNo then
686:                 begin
687:                     Button2.Click;
688:                 end;
689:             end;
690:         end;
691:     end;
692: end;

```

```

689:         end else
690:         begin
691:             if cbb74.Text='Base' then
692:                 begin
693:                     Sorge:=Form2.ABSTable10;
694:                     Sorge.Open;
695:                 end;
696:
697:             if cbb74.Text='GruppoA' then
698:                 begin
699:                     Sorge:=Form2.ABSTable11;
700:                     Sorge.Open;
701:                 end;
702:
703:             if cbb74.Text='GruppoB' then
704:                 begin
705:                     Sorge:=Form2.ABSTable12;
706:                     Sorge.Open;
707:                 end;
708:             //CONTROLLO CATEGORIA
709:             if Edit4.Text='' then
710:                 begin
711:                     ShowMessage(' Inserisci la Categoria');
712:                     Exit;
713:                 end;
714:             //Verifica se esiste già l'iscrizione
715:             Sorge.First;
716:             while not Sorge.Eof do
717:                 begin
718:                     n:=Sorge.FieldName('CODDSPL').AsInteger;
719:                     trovato:=Form2.ABSTable7.Locate(' CODGA;CODDSPL;CODATL',VarArrayOf([Edit1.Text
,IntToStr(n),IntToStr(ATL)]),[]);
720:                     if trovato=True then
721:                         begin
722:                             ShowMessage('ATTENZIONE ! - Questo Atleta risulta già iscritto '
723:                             +#13+'ad una o più gare del Gruppo. Non è consentita l''iscrizione automat
ica');
724:                             Exit;
725:                         end;
726:                     Sorge.Next;
727:                 end;
728:
729:             //Carica il nuovo record
730:             Sorge.First;
731:             while not Sorge.EOF do // Modo corretto di loop EOF
732:                 begin
733:                     Form2.ABSTable7.Insert;
734:                     Form2.ABSTable7.FieldName('CODGA').AsInteger:=StrToInt(Edit1.Text);
735:                     Form2.ABSTable7.FieldName('CODDSPL').AsInteger:=Sorge.FieldName('CODDSPL').
AsInteger;
736:                     Form2.ABSTable7.FieldName('CODATL').AsInteger:= ATL;
737:                     Form2.ABSTable7.FieldName('COGNOME').AsString:=Cogn;
738:                     Form2.ABSTable7.FieldName('NMPET').AsInteger:=0;
739:                     Form2.ABSTable7.FieldName('NMPOS').AsInteger:=0;
740:                     Form2.ABSTable7.FieldName('TAME1').AsDateTime:=StrToTime('00:00:00');
741:                     Form2.ABSTable7.FieldName('TBME2').AsDateTime:=StrToTime('00:00:00');
742:                     Form2.ABSTable7.FieldName('DQ').AsBoolean:=False;
743:                     Form2.ABSTable7.FieldName('PEN').AsDateTime:=StrToTime('00:00:00,000');
744:                     Form2.ABSTable7.FieldName('MANL').AsBoolean:=False;
745:                     Form2.ABSTable7.FieldName('CAT').AsInteger:=StrToInt(Edit4.Text);
746:                     Form2.ABSTable7.Post;
747:                     Sorge.Next;
748:                 end;
749:                 Button5.Click;
750:             end; //Iscrizione automatica terminata
751:         end;
752:

```



```

754:
755:   if (rb2.Checked=True) and (Edit2.Text<>'') then    //verifica se già inserito
756:   begin
757:       ds7.DataSet:=ABSQuery7;
758:       with AbsQuery7 do
759:       begin
760:           Close; //cerca i doppioni
761:           SQL.Text := 'Select * from Gara where CODATL = '+ QuotedStr(IntToStr(ATL))+
762:           ' AND CODDSPL = '+QuotedStr(Edit2.Text);
763:           ExecSql;
764:           Open;
765:           volte:=AbsQuery7.RecordCount;
766:           if volte>0 then
767:           begin
768:               ShowMessage(Nom+' '+Cogn+' è già iscritto '+IntToStr(volte)+' volte a ques
769: ta disciplina');
770:               Button5.Click;
771:               Form2.ABSTable2.First;
772:               Exit;
773:           end;
774:       end;
775:       if Edit4.Text='' then
776:       begin
777:           ShowMessage(' Inserisci la Categoria');
778:           Exit;
779:       end;
780:       if MessageDlg('Confermi l''iscrizione di '+Nom+ ' '+ Cogn,
781: mtconfirmation, [mbYes,mbNo], 0) = mrYes then
782:       begin
783:           Form2.ABSTable7.Insert;
784:           Form2.ABSTable7.FieldName('CODATL').AsInteger:=ATL;
785:           Form2.ABSTable7.FieldName('CODGA').AsInteger:= StrToInt(Edit1.Text);
786:           Form2.ABSTable7.FieldName('CODDSPL').AsInteger:=StrToInt(Edit2.Text);
787:           Form2.ABSTable7.FieldName('COGNOME').AsString:= Cogn;
788:           Form2.ABSTable7.Post;
789:           Button5.Click;          //per ripristinare l'evento si ds72
790:           Form2.ABSTable2.First;
791:           inserisco7:=False;
792:       end else Exit;
793:   end else
794:   begin
795:       if rb2.Checked=True then
796:       begin
797:           ShowMessage('Iscrizione manuale, indica prima il codice disciplina ');
798:           Button2.Click;
799:           exit;
800:       end;
801:   end else
802:   begin
803:       ShowMessage('Inserisci il Codice della Gara. ');
804:       Exit;
805:   end;
806:   end;
807:   //INSERISCE I Tempi(da unit Tempi)
808:   procedure TForm6.dbgrd7CellClick(Column: TColumn);
809:   begin
810:       if inserisco7=True then Exit;
811:       if tempo='' then
812:       begin
813:           ShowMessage('Il valore da inserire è nullo');
814:           Exit;
815:       end;
816:       ABSTable14.Edit;
817:       TDateTimeField(ABSTable14.FieldName('TAME1')).DisplayFormat:='nn:ss,zzz';
818:       TDateTimeField(ABSTable14.FieldName('TBME2')).DisplayFormat:='nn:ss,zzz';
819:       if (rb1.Checked=True) then

```

```

821:         if ABSTable14.FieldName('TAME1').AsDateTime<>0 then
822:             if MessageDlg('Esiste già un dato, vuoi sostituirlo?',
823:                 mtconfirmation, [mbYes,mbNo], 0,mbNo) = mrNo then Exit;
824:             ABSTable14.FieldName('TAME1').AsDateTime:=StrToTime(tempo);
825:         end;
826:         if (rab2 = True) then
827:             begin
828:                 if ABSTable14.FieldName('TAME1').AsDateTime<>0 then
829:                     if MessageDlg('Esiste già un dato, vuoi sostituirlo?',
830:                         mtconfirmation, [mbYes,mbNo], 0,mbNo) = mrNo then Exit;
831:                     ABSTable14.FieldName('TBME2').AsDateTime:=StrToTime(tempo);
832:                 end;
833:                 ABSTable14.Post;
834:                 tempo:='';
835:                 //trucco per poter fire l'onclick sulla stessa riga due volte
836:                 Form6.dbgrd7.SelectedField:=ABSTable14.FieldName('NM');
837:             end;
838:
839:
840: // Ricerca per cognome
841:
842:             // seleziona una lettera dalla combobox per poi inserire l'atleta
843: procedure TForm6.cbb72Select(Sender: TObject);
844: var
845: c:string;
846: begin
847:     if Edit1.Text<>' ' then
848:         begin
849:             c:=cbb72.Items[cbb72.ItemIndex]; //estrae la scelta dalla combobox
850:             scelta:=True; //scelta fatta per cognome
851:             ds72.DataSet:=AbsQuery72;
852:             with AbsQuery72 do
853:                 begin
854:                     Close; //inserisce una variabile nel query selezionandola per primo carattere
855:                     SQL.Text := 'Select * from AnaAtl where substring(COGNOME,1,1) = '+ QuotedStr(c);
856:                     ExecSql;
857:                     Open;
858:                     cbb72.ItemIndex:=-1;
859:                 end;
860:             end else
861:                 begin
862:                     cbb72.ItemIndex:=-1; //riporta la combobox all'inizio
863:                     ShowMessage('Inserisci il Codice della Gara');
864:                     Exit;
865:                 end;
866:         end;
867:
868: procedure TForm6.cbb74Change(Sender: TObject); //SCEGLIE IL GRUPPO
869: var //realizza per ogni gruppo scelto una tabella delle DSPL per quella gara
870: dsplSi:Boolean;
871: DSPLn:Integer;
872: baseSi:Boolean;
873: begin
874:     Form2.ABSTable6.Open;
875:     Form2.ABSTable10.TableName:='repBase';
876:     Form2.ABSTable10.Open;
877:     Form2.ABSTable10.Edit;
878:     Form2.ABSTable11.TableName:='repGRPA';
879:     Form2.ABSTable11.Open;
880:     Form2.ABSTable11.Edit;
881:     Form2.ABSTable12.TableName:='repGRPB';
882:     Form2.ABSTable12.Open;
883:     Form2.ABSTable12.Edit;
884:     if Edit1.Text<>' ' then
885:         begin
886:             if cbb74.Text='Base' then          // GRUPPO BASE
887:                 begin

```

```

    nto di Base
889:         begin                                     // esiste già
890:             baseSi:= Form2.ABSTable6.Locate('CODGA',strToInt(Edit1.Text),[]); //Verifica l'
esistenza
891:         if baseSi=True then                         //del codice della Gara negli accoppiamenti di
Base
892:             begin
893:                 if MessageDlg('Il Gruppo delle discipline Base di questa Gara'+#13+
894:                     'è già stato creato. Vuoi rifarlo ?', mtconfirmation, [mbYes,mbNo], 0,mbNo
) = mrYes then
895:                     begin
896:                         ShowMessage('Esci dalla Sezione Gara Corrente e lancia Anagrafica Gare'+
+#13+
897:                             'per rifare gli accoppiamenti Gruppo Base');
898:                         Exit;
899:                     end;
900:                 end else
901:                 begin
902:                     ShowMessage('Esci dalla Sezione Gara Corrente e lancia Anagrafica Gare'+#
13+
903:                         'per definire gli accoppiamenti Gruppo Base');
904:                         Exit;
905:                     end;
906:
907:                 end;
908:                 Form2.ABSTable10.Close;
909:                 Form2.ABSTable10.EmptyTable;
910:                 ds74.DataSet := ABSQuery74; //Costruisce una tabella SQL isolando solo le disciplin
e
911:                 with ABSQuery74 do                     // del Gruppo Base di quella Gara
912:                 begin
913:                     Close; //estrae le discipline relative alla gara ed al Gruppo Base
914:                     SQL.Text := 'Select * from Base where CODGA = '+QuotedStr(edit1.Text);
915:                     ExecSql;
916:                     Open; //la Tabella è formata da tutti i codici disciplina della Base
917:                     First;
918:                     Form2.ABSTable10.Open;
919:                     Form2.ABSTable10.Edit;
920:                     Form2.ABSTable4.Open;
921:                     while(not Eof)do //Crea le tabelle complete delle discipline per quel Gruppo
922:                     begin
923:                         DSPLn:=dbgrd74.Columns[0].Field.AsInteger; //preleva il cod della DSPL
924:                         dsplSi:= Form2.ABSTable4.Locate('CODDSPL',DSPLn,[]); // usa il cod per localizz
are il record di ABS/4
925:                         if dsplSi=True then
926:                             begin
927:                                 Form2.ABSTable10.Insert; //Costruisce la tabella completa delle discipl
ine
928:                                 Form2.ABSTable10.FieldName('CODDSPL').AsInteger:=Form2.ABSTable4.FieldName
ame('CODDSPL').AsInteger;
929:                                 Form2.ABSTable10.FieldName('DESCR').AsString:=Form2.ABSTable4.FieldName
('DESCR').AsString;
930:                                 Form2.ABSTable10.FieldName('TMAX').AsDateTime:=Form2.ABSTable4.FieldName
e('TMAX').AsDateTime;
931:                                 Form2.ABSTable10.Post
932:                             end else
933:                             begin
934:                                 ShowMessage('Attenzione ! Manca questo Codice dalla Tabella delle Dis
cipline : '
935:                                     + IntToStr(DSPLn));
936:                                 Exit;
937:                             end;
938:                             Next;
939:                         end;
940:
941:                     ds74.DataSet:=Form2.ABSTable10;
942:                 end;

```

```

944:         ShowMessage('La tabella mostra le discipline del Gruppo Base'+#13+
945:         ' per la Gara indicata.');
```

946: end;

```

947:
948:     if cbb74.Text='GruppoA' then
949:     begin
950:         Form2.ABSTable8.Open;
951:         if Form2.ABSTable8.RecordCount>0 then //Controlla se per la Gara scelta l'accoppiame
nto di GruppoA
952:         begin                                     // esiste già
953:             baseSi:= Form2.ABSTable8.Locate('CODGA',strToInt(Edit1.Text),[]); //Verifica l
esistenza
954:             if baseSi=True then                     //del codice della Gara negli accoppiamenti di Base
955:             begin
956:                 if MessageDlg('Il Gruppo A delle discipline di questa Gara'+#13+
957:                 'è già stato creato. Vuoi rifarlo ?', mtconfirmation, [mbYes,mbNo], 0,mbNo
958:                 ) = mrYes then
959:                 begin
960:                     ShowMessage('Esci dalla Sezione Gara Corrente e lancia Anagrafica Gare'
961:                     +#13+
962:                     'per rifare gli accoppiamenti del Gruppo A');
963:                     Exit;
964:                 end;
965:                 end else
966:                 begin
967:                     ShowMessage('Esci dalla Sezione Gara Corrente e lancia Anagrafica Gare'+#1
968:                     3+
969:                     'per definire gli accoppiamenti Gruppo A');
970:                     Exit;
971:                 end;
972:             end;
973:             Form2.ABSTable11.Close;
974:             Form2.ABSTable11.EmptyTable;
975:             ds74.DataSet := ABSQuery74; //Costruisce una tabella SQL isolando solo le disciplin
e
976:             with ABSQuery74 do                       // del Gruppo A di quella Gara
977:             begin
978:                 Close; //estrae le discipline relative alla gara ed al Gruppo A
979:                 SQL.Text := 'Select * from GRPA where CODGA = '+QuotedStr(edit1.Text);
980:                 ExecSql;
981:                 Open; //la Tabella è formata da tutti i codici disciplina della Base
982:                 First;
983:                 Form2.ABSTable11.Open;
984:                 Form2.ABSTable11.Edit;
985:                 while(not Eof)do //Crea le tabelle complete delle discipline per quel Gruppo
986:                 begin
987:                     DSPLn:=dbgrd74.Columns[0].Field.AsInteger; //preleva il cod della DSPL
988:                     dsplSi:= Form2.ABSTable4.Locate('CODDSPL',DSPLn,[]); // usa il cod per localizz
are il record di ABS/4
989:                     if dsplSi=True then
990:                     begin
991:                         Form2.ABSTable11.Insert;          //Costruisce la tabella completa delle discipl
ine
992:                         Form2.ABSTable11.FieldName('CODDSPL').AsInteger:=Form2.ABSTable4.FieldName
993:                         ame('CODDSPL').AsInteger;
994:                         Form2.ABSTable11.FieldName('DESCR').AsString:=Form2.ABSTable4.FieldName
995:                         ('DESCR').AsString;
996:                         Form2.ABSTable11.FieldName('TMAX').AsDateTime:=Form2.ABSTable4.FieldName
997:                         e('TMAX').AsDateTime;
998:                         Form2.ABSTable11.Post
999:                     end else
1000:                     begin
1001:                         ShowMessage('Attenzione ! Manca questo Codice dalla Tabella delle Dis
1002:                         cipline : '
1003:                         + IntToStr(DSPLn));
1004:                     end;
1005:                 end;
1006:             end;
1007:         end;
1008:     end;
1009: 
```

```

1000:         Next;
1001:     end;
1002:
1003:     ds74.DataSet:=Form2.ABSTable11;
1004: end;
1005:
1006:     ShowMessage('La tabella mostra le discipline del Gruppo A');
1007: end;
1008:
1009:     if cbb74.Text='GruppoB' then
1010: begin
1011:     Form2.ABSTable9.Open;
1012:     if Form2.ABSTable9.RecordCount>0 then //Controlla se per la Gara scelta l'accoppiame
nto di Gruppo B
1013:     begin // esiste già
1014:         baseSi:= Form2.ABSTable9.Locate('CODGA',strToInt(Edit1.Text),[]); //Verifica l'
esistenza
1015:         if baseSi=True then //del codice della Gara negli accoppiamenti di Base
1016:         begin
1017:             if MessageDlg('Il Gruppo B delle discipline di questa Gara'+#13+
1018:                 'è già stato creato. Vuoi rifarlo ?', mtconfirmation, [mbYes,mbNo], 0,mbNo
) = mrYes then
1019:             begin
1020:                 ShowMessage('Esci dalla Sezione Gara Corrente e lancia Anagrafica Gare
'+#13+
1021:                     'per rifare gli accoppiamenti del Gruppo B');
1022:                 Exit ;
1023:             end;
1024:         end else
1025:         begin
1026:             ShowMessage('Esci dalla Sezione Gara Corrente e lancia Anagrafica Gare'+#1
3+
1027:                 'per definire gli accoppiamenti del Gruppo B');
1028:                 Exit ;
1029:             end;
1030:         end;
1031:     Form2.ABSTable12.Close;
1032:     Form2.ABSTable12.EmptyTable;
1033:     ds74.DataSet := ABSQuery74; //Costruisce una tabella SQL isolando solo le disciplin
e
1034:     with ABSQuery74 do // del Gruppo B di quella Gara
1035:     begin
1036:         Close; //estrae le discipline relative alla gara ed al Gruppo B
1037:         SQL.Text := 'Select * from GRPB where CODGA = '+QuotedStr(edit1.Text);
1038:         ExecSql;
1039:         Open; //la Tabella è formata da tutti i codici disciplina della Base
1040:         First;
1041:         Form2.ABSTable12.Open;
1042:         Form2.ABSTable12.Edit;
1043:         while(not Eof)do //Crea le tabelle complete delle discipline per quel Gruppo
1044:         begin
1045:             DSPLn:=dbgrd74.Columns[0].Field.AsInteger; //preleva il cod della DSPL
1046:             dsplSi:= Form2.ABSTable4.Locate('CODDSPL',DSPLn,[]); // usa il cod per localizz
are il record di ABS/4
1047:             if dsplSi=True then
1048:             begin
1049:                 Form2.ABSTable12.Insert; //Costruisce la tabella completa delle discipl
ine
1050:                 Form2.ABSTable12.FieldByName('CODDSPL').AsInteger:=Form2.ABSTable4.FieldByN
ame('CODDSPL').AsInteger;
1051:                 Form2.ABSTable12.FieldByName('DESCR').AsString:=Form2.ABSTable4.FieldByName
('DESCR').AsString;
1052:                 Form2.ABSTable12.FieldByName('TMAX').AsDateTime:=Form2.ABSTable4.FieldByNam
e('TMAX').AsDateTime;
1053:                 Form2.ABSTable12.Post
1054:             end else
1055:             begin

```

```
cipline : '
1057:         + IntToStr(DSPLn));
1058:         Exit;
1059:     end;
1060:     Next;
1061: end;
1062:
1063:     ds74.DataSet:=Form2.ABSTable12;
1064: end;
1065:     ShowMessage('La tabella mostra le discipline del Gruppo B');
1066: end;
1067: end else
1068: begin
1069:     ShowMessage('Inserisci il Codice della Gara ');
1070:     cbb74.ItemIndex:=0;
1071:     Exit;
1072: end;;
1073: end;                                     //NAVIGATORE iscrizioni
1074:
1075: procedure TForm6.dbnvgr7Click(Sender: TObject; Button: TNavigateBtn);
1076: begin
1077:     case button of
1078:         nbInsert: //se inserisce un nuovo record
1079:         begin
1080:             if (Edit1.Text='') OR (Edit2.Text='') then
1081:                 begin
1082:                     ShowMessage('Inserisci il Codice della Gara e/o della Disciplina');
1083:                     Exit;
1084:                 end;
1085:             Form2.ABSTable7.Last;
1086:             Form2.ABSTable7.Append;
1087:             inserisco7:=True; //inserisce il flag nel nuovo record
1088:             dbgrd7.SetFocus;
1089:             DBGrd7.SelectedField := Form2.ABSTable7.FieldName('CODATL');
1090:         end;
1091:         nbPost:
1092:         begin //se lo carica nella tabella
1093:             inserisco7:=False ;
1094:         end;
1095:         nbDelete: //se cancella il record - serve anche per evitare di entrare in Query
1096:         begin
1097:             inserisco7:=False;
1098:         end;
1099:         nbEdit:
1100:             inserisco7:=True;
1101:         end;
1102:     end;
1103:
1104: procedure TForm6.Edit1Change(Sender: TObject);
1105: begin
1106:     if rb1.Checked = True then
1107:         cbb74.ItemIndex:=0;
1108:     end;
1109:
1110: end.
1111:
```

```
1: unit Classifiche;
2:
3: interface
4:
5: uses
6:   Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,
7:   Vcl.Graphics, Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.CheckLst,
8:   Vcl.Grids, System.DateUtils, Data.DB, Vcl.DBGrids, Vcl.DBCtrls, Vcl.ExtCtrls,
9:   ABSMain, frxClass, frxDBSet, Datasnap.DBClient, Datasnap.Provider, Vcl.HtmlHelpViewer;
10:
11: type
12:   TForm8 = class(TForm)
13:     Esci: TButton;
14:     DBGrid8: TDBGrid;
15:     DataSource8: TDataSource;
16:     Button1: TButton;
17:     Button2: TButton;
18:     Button3: TButton;
19:     Label1: TLabel;
20:     ListBox8: TListBox;
21:     Button4: TButton;
22:     Button6: TButton;
23:     Shape1: TShape;
24:     Label3: TLabel;
25:     Label4: TLabel;
26:     Label5: TLabel;
27:     Shape2: TShape;
28:     Shape3: TShape;
29:     Button7: TButton;
30:     RadioButton1: TRadioButton;
31:     RadioButton2: TRadioButton;
32:     RadioButton3: TRadioButton;
33:     ComboBox1: TComboBox;
34:     Label6: TLabel;
35:     ABSQuery80: TABSQuery;
36:     DataSourceQry: TDataSource;
37:     ABSTable15: TABSTable;
38:     ABSQuery85: TABSQuery;
39:     frxReport8: TfrxReport;
40:     frxDBDataset85: TfrxDBDataset;
41:     ABSQuery83: TABSQuery;
42:     ABSTable16: TABSTable;
43:     DataSourceQry2: TDataSource;
44:     ABSTable18: TABSTable;
45:     frxReport8D: TfrxReport;
46:     frxDBDataset8D: TfrxDBDataset;
47:     frxReport16: TfrxReport;
48:     frxDBDataset16: TfrxDBDataset;
49:     frxReport16T: TfrxReport;
50:     frxDBDataset16T: TfrxDBDataset;
51:     frxReport16T2: TfrxReport;
52:     frxDBDataset16t2: TfrxDBDataset;
53:     ABSQueryRep: TABSQuery;
54:     ABSTable19: TABSTable;
55:     frxReportTot: TfrxReport;
56:     frxDBDatasetTot: TfrxDBDataset;
57:     Bevel1: TBevel;
58:     ABSTable16R: TABSTable;
59:     frxReportR: TfrxReport;
60:     frxDBDatasetR: TfrxDBDataset;
61:     ABSQueryR: TABSQuery;
62:     procedure FormCreate(Sender: TObject);
63:     procedure EsciClick(Sender: TObject);
64:     procedure Button1Click(Sender: TObject);
65:     procedure DBGrid8CellClick(Column: TColumn);
66:     procedure Button2Click(Sender: TObject);
67:     procedure Button3Click(Sender: TObject);
```



```

69:     procedure Button6Click(Sender: TObject);
70:     procedure Calcola(Sender: TObject);
71:     procedure Button7Click(Sender: TObject);
72:     private
73:     { Private declarations }
74:     public
75:     { Public declarations }
76:     end;
77:
78: var
79:     Form8: TForm8;                                //ArGare Gare selezionate
80:     ClassFlg:Boolean=True;                        //ArDspl Discipline selezionate
81:     ArGare,ArDspl,ArCat:array of Integer; //ArCat Categorie selezionate
82:     SelGare:Boolean=False;
83:     SelDspl:Boolean=False;
84:     ClassDp:Boolean=False;
85:     i,idx,mom,momd,MaxIdxGare,MaxIdxDspl,MaxIdxCat:Integer;
86: implementation
87: uses
88: Console1 ;
89: {$R *.dfm}
90: {$D+}
91: procedure TForm8.FormCreate(Sender: TObject);
92: begin
93:     ClassFlg:=False;
94:     Form2.ABSTable3.TableName:='Selezionati';
95:     Form2.ABSTable3.Open;
96:     Form2.ABSTable3.Edit;
97:     ABSTable15.TableName:='Classifica';
98:     ABSTable15.Open;
99:     ABSTable15.Edit;
100:    Form2.ABSTable1.Open;
101:    Form2.ABSTable1.INSERT;
102:    Form2.ABSTable1.FieldName('IN').AsDateTime:= NOW;
103:    Form2.ABSTable1.FieldName('what').AsString:='Accesso alle Classifiche';
104:    Form2.ABSTable1.FieldName('OUT').AsString:='';
105:    Form2.ABSTable1.Post;
106:    Form2.ABSTable1.Close;
107:    if ABSTable15.RecordCount>0 then
108:        begin
109:            TDateTimeField(Form8.ABSTable15.FieldName('TAME1')).DisplayFormat:='nn:ss,zzz';
110:            TDateTimeField(Form8.ABSTable15.FieldName('TBME2')).DisplayFormat:='nn:ss,zzz';
111:            TDateTimeField(Form8.ABSTable15.FieldName('PEN')).DisplayFormat:='nn:ss,zzz';
112:        end;
113:    ABSTable16.TableName:='Ranking';
114:    ABSTable16R.TableName:='R16R';
115:    ABSTable18.TableName:='Temp8';
116:    Form2.ABSTable5.Open;
117:    Form2.ABSTable4.Open;
118:    SetLength(ArGare,20);
119:    SetLength(ArDspl,20);
120:    SetLength(ArCat,10);
121:    ABSTable19.TableName:='totale';
122:    Shape1.Brush.Color := clRed;
123:    Shape2.Brush.Color := clRed;
124:    Shape3.Brush.Color := clRed;
125:    RadioButton1.Color:= clRed;
126:    RadioButton2.Color:= clRed;
127:    RadioButton3.Color:= clRed;
128: end;
129:
130: procedure TForm8.Button1Click(Sender: TObject); //Seleziona le gare
131: begin
132:     DataSource8.DataSet:=Form2.ABSTable5;          //display gare su dbgrid
133:     DBGrid8.DataSource:=DataSource8;
134:     DBGrid8.Width:=376;
135:     DBGrid8.Columns[0].Width:=44.

```



```

137:     DBGrid8.Columns[2].Width:=228;
138:     SelGare:=True;
139:     SelDspl:=False;
140:     SetLength(ArGare,0);
141:     SetLength(ArGare,20);
142:     Shape1.Brush.Color := clRed;
143: end;
144:
145: procedure TForm8.DBGrid8CellClick(Column: TColumn); //Inserisci-Scegli
146: begin
147:     ListBox8.Items.Add(DBGrid8.Fields[0].AsString);
148:     Self.ActiveControl:=nil;
149: end;
150:
151: procedure TForm8.Button2Click(Sender: TObject); //Rimuovere
152: begin
153:     if ListBox8.Items.Count > 0 then //verifica che non c'è selezionato
154:     begin
155:         if ListBox8.Itemindex = -1 then
156:         begin
157:             ShowMessage('Nessun elemento selezionato');
158:             Exit;
159:         end;
160:     end;
161:     ListBox8.DeleteSelected;
162: end;
163:
164: procedure TForm8.Button3Click(Sender: TObject); //Salva i valori negli array
165: var
166: k:ShortInt;
167: kk:Integer;
168: begin
169:     for i := ListBox8.Items.Count - 1 downto 0 do //toglie gli spazi vuoti
170:     begin
171:         if (Trim(ListBox8.Items[i]) = '') then
172:             ListBox8.Items.Delete(i);
173:     end;
174:
175:     if MessageDlg('Confermi gli elementi scelti ?',mtconfirmation,[mbYes,mbNo], 0)
176:     = mrNo then
177:     begin
178:         if SelGare=True then // verifica la scelta
179:         begin
180:             SetLength(ArGare,0);
181:             SetLength(ArGare,20);
182:             Shape1.Brush.Color := clRed;
183:         end;
184:         if SelDspl=True then
185:         begin
186:             SetLength(ArDspl,0);
187:             SetLength(ArDspl,20);
188:             Shape2.Brush.Color := clRed;
189:         end;
190:         ListBox8.Clear;
191:         Exit;
192:     end;
193:
194:     if SelGare=True then //Salva le Gare scelte nell'ARRAY
195:     begin
196:         for k := 0 to ListBox8.Items.Count-1 do
197:         begin
198:             kk:= StrToInt(ListBox8.Items[k]); // NOTA BENE gli array non consentono
199:             ArGare[k]:= kk; // di eliminare il passaggio tramite kk
200:         end;
201:
202:         for i := 0 to High(ArGare) do //sistema per verificare se l'array è vuoto
203:             if ArGare[i]=0 then // se è uguale a zero, indica precedente

```

```

205:         mom := i-1;
206:         MaxIdxGare:=mom;
207:         Break;
208:     end;
209:     if mom=-1 then
210:     begin
211:         ShowMessage('Non vi sono elementi da salvare ! - ripeti');
212:         Exit;
213:     end else
214:         begin
215:             ShowMessage(' Dati salvati');
216:             Shape1.Brush.Color := clLime;
217:         end ;
218:
219:         for k := ListBox8.Items.Count - 1 downto 0 do //resetta la listBox
220:             ListBox8.Items.Delete(k);
221:             SelGare:=False;
222:         end;
223:
224:     if SelDspl=True then // memorizza le discipline
225:     begin
226:         for k := 0 to ListBox8.Items.Count-1 do
227:         begin
228:             kk:= StrToInt(ListBox8.Items[k]); // NOTA BENE gli array non consentono
229:             ArDspl[k]:= kk; // di eliminare il passaggio tramite kk
230:         end;
231:
232:         for i := 0 to High(ArDspl) do //Questo è l'unico sistema per verificare
233:             if ArDspl[i]=0 then // se l'array è vuoto
234:             begin
235:                 mom := i-1;
236:                 MaxIdxDspl:=mom;
237:                 Break;
238:             end;
239:             if mom=-1 then
240:             begin
241:                 ShowMessage('Non vi sono elementi da salvare ! - ripeti');
242:                 Exit;
243:             end else
244:                 begin
245:                     ShowMessage(' Dati salvati');
246:                     Shape2.Brush.Color := clLime;
247:                 end ;
248:                 for k := ListBox8.Items.Count - 1 downto 0 do //resetta la listBox
249:                     ListBox8.Items.Delete(k);
250:                     SelDspl:=False;
251:                 end;
252:
253:     end;
254:
255: procedure TForm8.Button4Click(Sender: TObject); //Seleziona le discipline
256: begin
257:     DataSource8.DataSet:=Form2.ABSTable4; //display gare su dbgrid
258:     DBGrid8.DataSource:=DataSource8;
259:     DBGrid8.Width:=376;
260:     DBGrid8.Columns[0].Width:=44;
261:     DBGrid8.Columns[1].Width:=241;
262:     DBGrid8.Columns[2].Width:=50;
263:     Shape2.Brush.Color := clRed;
264:     SelDspl:=True;
265:     SelGare:=False;
266:     SetLength(ArDspl,0);
267:     SetLength(ArDspl,20);
268:     for i := 0 to High(ArGare) do //Controlla che le gare siano state indicate
269:         if ArGare[i]=0 then
270:         begin
271:             mom := i-1; //mom = all'ultima elemento dell'array memorizzato

```

```

273:         end;
274:     if mom=-1 then
275:     begin
276:         Shapel.Brush.Color := clRed;
277:         ShowMessage('Scegli prima una o più gare');
278:         Exit;
279:     end;
280:
281: end;
282:
283: procedure TForm8.Button6Click(Sender: TObject);          // CLASSIFICARE PER GARE
284: var
285:     garesi,dsplsi,trovato:Boolean;
286:     QryStr:string;
287:     Cgara,Cdspl,qtrl:Integer;
288: begin
289:     Form2.ABSTable3.Close;
290:     Form2.ABSTable3.EmptyTable; //resetta la tabella gare selezionate
291:     Form2.ABSTable3.Open;
292:     for i := 0 to High(ArGare) do
293:     if ArGare[i]=0 then
294:     begin
295:         mom := i-1; //mom = all'ultimo elemento dell'array memorizzato
296:         Break;
297:     end;
298:     if mom=-1 then
299:     begin
300:         ShowMessage('Non vi sono elementi memorizzati ! - ripeti');
301:         Shapel.Brush.Color := clRed;
302:         Exit;
303:     end;
304:     for i:= 0 to mom do          //crea la tabella 3
305:     begin
306:         idx:=ArGare[i]; //Trasferisce i dati dall'Array alla Tabella
307:         Form2.ABSTable3.Insert;
308:         Form2.ABSTable3.FieldName('Codice').AsInteger:=idx;
309:         garesi:=Form2.ABSTable5.Locate('CODGA',idx,[]);
310:         Form2.ABSTable3.FieldName('Descrizione').AsString:=
311:             Form2.ABSTable5.FieldName('LUOGO').AsString;
312:         Form2.ABSTable3.Post;
313:     end;
314:
315:     DataSource8.DataSet:=Form2.ABSTable3;          //display gare su dbgrid
316:     DBGrid8.DataSource:=DataSource8;
317:     DBGrid8.Width:=376;
318:     DBGrid8.Columns[0].Width:=44;
319:     DBGrid8.Columns[1].Width:=291;
320:     if MessageDlg('Confermi di effettuare la classifica su questi elementi ?',
321:         mtconfirmation, [mbYes,mbNo], 0) = mrNo then
322:     begin
323:         Form2.ABSTable3.Close;
324:         Form2.ABSTable3.EmptyTable;
325:         Shapel.Brush.Color := clRed;
326:         SetLength(ArGare,0); //azzerare l'array delle Gare non confermando la scelta
327:         SetLength(ArGare,20);
328:         Exit;
329:     end;
330:     if comboBox1.ItemIndex=-1 then
331:     begin
332:         ShowMessage('Devi scegliere una Categoria');
333:         Exit;
334:     end else
335:     begin
336:         if MessageDlg('Confermi di effettuare la classifica per la Categoria '+comboBox1.Text+
337:             ' ?',
338:             mtconfirmation, [mbYes,mbNo], 0) = mrNo then
339:             begin

```

```
340:         Exit ;
341:     end ;
342:     Shape3.Brush.Color := clLime;
343: end;
344:
345:
346:     //GRUPPO BASE
347: if RadioButton1.Checked=True then
348:     if MessageDlg('Confermi di effettuare la classifica sul Gruppo Base ?',
349:         mtconfirmation, [mbYes,mbNo], 0) = mrNo then
350:         begin
351:             ShowMessage('Cambia il Gruppo');
352:             Exit;
353:         end else
354:         begin //converte l'array delle gare in string per il Query
355:             if Form2.ABSTable6.RecordCount=0 then
356:                 begin
357:                     ShowMessage('Attenzione il Gruppo scelto non contiene discipline !');
358:                     Exit;
359:                 end;
360:                 QryStr:='(';
361:                 for i := 0 to MaxIdxGare-1 do
362:                     QryStr:=QryStr+IntToStr(ArGare[i])+',';
363:                     QryStr:=QryStr+IntToStr(ArGare[MaxIdxGare]);
364:                     QryStr:=QryStr+')';
365:
366:                 with ABSQuery80 do //Estrae dal Base le discipline gruppo Base
367:                     begin //per tutte le gare indicate e estrae le
368:                         Close; // coppie di valori n.gara e n.disciplina
369:                         SQL.Text:='Select * from Base where CODGA IN '+QryStr; //dall'ARRAY
370:                         ExecSQL;
371:                         Open;
372:                     end;
373:                     if ABSQuery80.RecordCount=0 then
374:                         begin
375:                             ShowMessage('Attenzione il Gruppo scelto non contiene discipline !');
376:                             Exit;
377:                         end;
378:                         //Fase di raccolta record da classificare in T15 Classifica
379:                         if ABSTable15.RecordCount>0 then //Azzerò la tabella Classifica
380:                             begin // preesistente per non causare
381:                                 ABSTable15.Close; // problemi di indice unique
382:                                 ABSTable15.EmptyTable;
383:                                 ABSTable15.Open;
384:                             end;
385:                             ABSTable15.Last;
386:                             ABSQuery80.First; //ABSQuery80 è una tabella con gare e discipline
387:                             qtrl:=ABSQuery80.RecordCount;
388:                             while not ABSQuery80.EOF do // selezionate
389:                                 begin
390:                                     Cgara:=ABSQuery80.FieldName('CODGA').AsInteger;
391:                                     Cdspl:=ABSQuery80.FieldName('CODDSPL').AsInteger;
392:                                     with ABSQuery85 do //Estrae dalle Gare le discipline da classif
393:                                         begin //Costruisce Classifica con tutti i record gare e discipline
394:                                             Close; //selezionate e gli Atleti che le hanno in comune
395:                                             SQL.Text:='INSERT INTO Classifica SELECT * From Gara where CODGA='
396:                                                 +QuotedStr(IntToStr(Cgara))+ ' AND CODDSPL='+QuotedStr(IntToStr(Cdspl))
397:                                                 + 'AND CAT =' + quotedstr(comboBox1.text);
398:                                             ExecSQL;
399:                                         end;
400:                                         qtrl:=qtrl-1;
401:                                         ABSQuery80.Next;
402:                                     end;
403:                                     ComboBox1.ItemIndex:=-1;
404:                                     if qtrl>0 then ShowMessage('ABSQuery80 errore eof');
405:
406:                                     with ABSQuery85 do
```

```
408:         Close;
409:         SQL.Text:='Select * from Classifica Order By "COGNOME","CODDSPL";
410:         ExecSQL;
411:         Open;
412:     end;
413:     ABSTable15.Close;    //30/04/15 18:06 devo cancellare per poter sostituire
414:     ABSTable15.EmptyTable; //con i dati sortati e senza problemi di indice
415:     ABSTable15.Open;
416:     ABSTable15.Last;
417:     ABSQuery85.First;
418:     qtrl:=ABSQuery85.RecordCount;
419:     while not ABSQuery85.EOF do //copia il query ordinato nella Tabella 15
420:     begin
421:         ABSTable15.Insert;
422:         ABSTable15.FieldByName('NMR').AsInteger:=
423:             ABSQuery85.FieldByName('NMR').AsInteger;
424:         ABSTable15.FieldByName('CODGA').AsInteger:=
425:             ABSQuery85.FieldByName('CODGA').AsInteger;
426:         ABSTable15.FieldByName('CODDSPL').AsInteger:=
427:             ABSQuery85.FieldByName('CODDSPL').AsInteger;
428:         ABSTable15.FieldByName('CODATL').AsInteger:=
429:             ABSQuery85.FieldByName('CODATL').AsInteger;
430:         ABSTable15.FieldByName('COGNOME').AsString:=
431:             ABSQuery85.FieldByName('COGNOME').AsString;
432:         ABSTable15.FieldByName('NMPET').AsInteger:=
433:             ABSQuery85.FieldByName('NMPET').AsInteger;
434:         ABSTable15.FieldByName('NMPOS').AsInteger:=
435:             ABSQuery85.FieldByName('NMPOS').AsInteger;
436:         ABSTable15.FieldByName('TAME1').AsDateTime:=
437:             ABSQuery85.FieldByName('TAME1').AsDateTime;
438:         ABSTable15.FieldByName('TBME2').AsDateTime:=
439:             ABSQuery85.FieldByName('TBME2').AsDateTime;
440:         ABSTable15.FieldByName('DQ').AsBoolean:=
441:             ABSQuery85.FieldByName('DQ').AsBoolean;
442:         ABSTable15.FieldByName('PEN').AsDateTime:=
443:             ABSQuery85.FieldByName('PEN').AsDateTime;
444:         ABSTable15.FieldByName('MANL').AsBoolean:=
445:             ABSQuery85.FieldByName('MANL').AsBoolean;
446:         ABSTable15.FieldByName('CAT').AsInteger:=
447:             ABSQuery85.FieldByName('CAT').AsInteger;
448:         qtrl:=qtrl-1;
449:         ABSTable15.Post;
450:         ABSQuery85.Next;
451:     end;
452:     if qtrl>0 then ShowMessage('ABSQuery85 errore eof');
453:
454:     if MessageDlg('Desideri esaminare un rapporto dei Dati Base per la Classifica ?'
455:     ,mtconfirmation, [mbYes,mbNo], 0) = mrYes then
456:     begin
457:         frxReport8.ShowReport;    //dati da ABSTable15 (Classifica)
458:         if MessageDlg('Confermi i dati a base della classifica? '+#13+
459:         'In caso negativo ESCI e correggi le scelte fatte.',
460:         ,mtconfirmation, [mbYes,mbNo], 0) = mrNO then
461:         begin
462:             ABSTable15.Close;
463:             ABSTable15.EmptyTable;
464:             Exit;
465:         end;
466:     end;
467:
468:     RadioButton1.Color:= clLime;
469:     Calcola(Self);
470:     Exit;
471: end;
472:
473: //GRUPPO A
474: if RadioButton2.Checked=True then
```

```
476:         mtconfirmation, [mbYes,mbNo], 0) = mrNo then
477: begin
478:     ShowMessage('Cambia il Gruppo');
479:     Exit;
480: end else
481: begin //converte l'array delle gare in string per il Query
482:     if Form2.ABSTable8.RecordCount=0 then
483:         begin
484:             ShowMessage('Attenzione il Gruppo scelto non contiene discipline !');
485:             Exit;
486:         end;
487:         QryStr:='(';
488:         for i := 0 to MaxIdxGare-1 do
489:             QryStr:=QryStr+IntToStr(ArGare[i])+',';
490:             QryStr:=QryStr+IntToStr(ArGare[MaxIdxGare]);
491:             QryStr:=QryStr+')';
492:
493:         with ABSQuery80 do //Estrae dal Base le discipline gruppo A
494:             begin //per tutte le gare indicate e estrae le
495:                 Close; // coppie di valori n.gara e n.disciplina
496:                 SQL.Text:='Select * from GRPA where CODGA IN '+QryStr; //dall'ARRAY
497:                 ExecSQL;
498:                 Open;
499:             end; //Fase di raccolta record da classificare in T15 Classifica
500:         if ABSQuery80.RecordCount=0 then
501:             begin
502:                 ShowMessage('Attenzione il Gruppo scelto non contiene discipline !');
503:                 Exit;
504:             end;
505:
506:         if ABSTable15.RecordCount>0 then //Azzero la tabella Classifica
507:             begin // preesistente per non causare
508:                 ABSTable15.Close; // problemi di indice unique
509:                 ABSTable15.EmptyTable;
510:                 ABSTable15.Open;
511:             end;
512:         ABSTable15.Last;
513:         ABSQuery80.First; //ABSQuery80 è una tabella con gare e discipline
514:         qtrl:=ABSQuery80.RecordCount;
515:         while not ABSQuery80.EOF do // selezionate
516:             begin
517:                 Cgara:=ABSQuery80.FieldByName('CODGA').AsInteger;
518:                 Cdspl:=ABSQuery80.FieldByName('CODDSPL').AsInteger;
519:                 with ABSQuery85 do //Estrae dalle Gare le discipline da classif
520:                     begin //Costruisce Classifica con tutti i record gare e discipline
521:                         Close; //selezionate e gli Atleti che le hanno in comune
522:                         SQL.Text:='INSERT INTO Classifica SELECT * From Gara where CODGA='
523:                         +QuotedStr(IntToStr(Cgara))+ ' AND CODDSPL='+QuotedStr(IntToStr(Cdspl))
524:                         + 'AND CAT =' + quotedstr(comboBox1.text);
525:                         ExecSQL;
526:                     end;
527:                     qtrl:=qtrl-1;
528:                     ABSQuery80.Next;
529:                 end;
530:                 ComboBox1.ItemIndex:=-1;
531:         if qtrl>0 then ShowMessage('ABSQuery80 errore eof');
532:
533:         with ABSQuery85 do
534:             begin
535:                 Close;
536:                 SQL.Text:='Select * from Classifica Order By "COGNOME","CODDSPL"';
537:                 ExecSQL;
538:                 Open;
539:             end;
540:             ABSTable15.Close; //30/04/15 18:06 devo cancellare per poter sostituire
541:             ABSTable15.EmptyTable; //con i dati sortati e senza problemi di indice
542:             ABSTable15.Open;
```



```
544:      ABSQuery85.First;
545:      qtrl:=ABSQuery85.RecordCount;
546:      while not ABSQuery85.EOF do //copia il query ordinato nella Tabella 15
547:      begin
548:          ABSTable15.Insert;
549:          ABSTable15.FieldByName('NMR').AsInteger:=
550:              ABSQuery85.FieldByName('NMR').AsInteger;
551:          ABSTable15.FieldByName('CODGA').AsInteger:=
552:              ABSQuery85.FieldByName('CODGA').AsInteger;
553:          ABSTable15.FieldByName('CODDSPL').AsInteger:=
554:              ABSQuery85.FieldByName('CODDSPL').AsInteger;
555:          ABSTable15.FieldByName('CODATL').AsInteger:=
556:              ABSQuery85.FieldByName('CODATL').AsInteger;
557:          ABSTable15.FieldByName('COGNOME').AsString:=
558:              ABSQuery85.FieldByName('COGNOME').AsString;
559:          ABSTable15.FieldByName('NMPET').AsInteger:=
560:              ABSQuery85.FieldByName('NMPET').AsInteger;
561:          ABSTable15.FieldByName('NMPOS').AsInteger:=
562:              ABSQuery85.FieldByName('NMPOS').AsInteger;
563:          ABSTable15.FieldByName('TAME1').AsDateTime:=
564:              ABSQuery85.FieldByName('TAME1').AsDateTime;
565:          ABSTable15.FieldByName('TBME2').AsDateTime:=
566:              ABSQuery85.FieldByName('TBME2').AsDateTime;
567:          ABSTable15.FieldByName('DQ').AsBoolean:=
568:              ABSQuery85.FieldByName('DQ').AsBoolean;
569:          ABSTable15.FieldByName('PEN').AsDateTime:=
570:              ABSQuery85.FieldByName('PEN').AsDateTime;
571:          ABSTable15.FieldByName('MANL').AsBoolean:=
572:              ABSQuery85.FieldByName('MANL').AsBoolean;
573:          ABSTable15.FieldByName('CAT').AsInteger:=
574:              ABSQuery85.FieldByName('CAT').AsInteger;
575:          qtrl:=qtrl-1;
576:          ABSTable15.Post;
577:          ABSQuery85.Next;
578:      end;
579:      if qtrl>0 then ShowMessage('ABSQuery85 errore eof');
580:
581:      if MessageDlg('Desideri esaminare un rapporto dei Dati Base per la Classifica ?'
582:          ,mtconfirmation, [mbYes,mbNo], 0) = mrYes then
583:      begin
584:          frxReport8.ShowReport;
585:          if MessageDlg('Confermi i dati a base della classifica? '+#13+
586:              'In caso negativo ESCI e correggi le scelte fatte.',
587:              mtconfirmation, [mbYes,mbNo], 0) = mrNO then
588:              begin
589:                  ABSTable15.Close;
590:                  ABSTable15.EmptyTable;
591:                  Exit;
592:              end;
593:          end;
594:
595:          RadioButton2.Color:= clLime;
596:          Calcola(Self);
597:          Exit;
598:      end;
599:
600:
601:      //GRUPPO B
602:      if RadioButton3.Checked=True then
603:          if MessageDlg('Confermi di effettuare la classifica sul Gruppo B ?',
604:              mtconfirmation, [mbYes,mbNo], 0) = mrNo then
605:              begin
606:                  ShowMessage('Cambia il Gruppo');
607:                  Exit;
608:              end else
609:              begin //converte l'array delle gare in string per il Query
610:                  if Form2.ABSTable8.RecordCount=0 then
```

```

612:         ShowMessage('Attenzione il Gruppo scelto non contiene discipline !');
613:         Exit;
614:     end;
615:     QryStr:='(';
616:     for i := 0 to MaxIdxGare-1 do
617:         QryStr:=QryStr+IntToStr(ArGare[i])+',';
618:         QryStr:=QryStr+IntToStr(ArGare[MaxIdxGare]);
619:         QryStr:=QryStr+')';
620:
621:     with ABSQuery80 do          //Estrae dal Base le discipline gruppo B
622:     begin                      //per tutte le gare indicate e estrae le
623:         Close;                // coppie di valori n.gara e n.disciplina
624:         SQL.Text:='Select * from GRPB where CODGA IN '+QryStr; //dall'ARRAY
625:         ExecSQL;
626:         Open;
627:     end; //Fase di raccolta record da classificare in T15 Classifica
628:     if ABSQuery80.RecordCount=0 then
629:     begin
630:         ShowMessage('Attenzione il Gruppo scelto non contiene discipline !');
631:         Exit;
632:     end;
633:
634:     if ABSTable15.RecordCount>0 then //Azzerò la tabella Classifica
635:     begin                          // preesistente per non causare
636:         ABSTable15.Close;          // problemi di indice unique
637:         ABSTable15.EmptyTable;
638:         ABSTable15.Open;
639:     end;
640:     ABSTable15.Last;
641:     ABSQuery80.First; //ABSQuery80 è una tabella con gare e discipline
642:     qtrl:=ABSQuery80.RecordCount;
643:     while not ABSQuery80.EOF do    // selezionate
644:     begin
645:         Cgara:=ABSQuery80.FieldByName('CODGA').AsInteger;
646:         Cdspl:=ABSQuery80.FieldByName('CODDSPL').AsInteger;
647:         with ABSQuery85 do        //Estrae dalle Gare le discipline da classif
648:         begin //Costruisce Classifica con tutti i record gare e discipline
649:             Close; //selezionate e gli Atleti che le hanno in comune
650:             SQL.Text:='INSERT INTO Classifica SELECT * From Gara where CODGA='
651:             +QuotedStr(IntToStr(Cgara))+ ' AND CODDSPL='+QuotedStr(IntToStr(Cdspl))
652:             + 'AND CAT =' + quotedstr(comboBox1.text);
653:             ExecSQL;
654:         end;
655:         qtrl:=qtrl-1;
656:         ABSQuery80.Next;
657:     end;
658:     ComboBox1.ItemIndex:=-1;
659:     if qtrl>0 then ShowMessage('ABSQuery80 errore eof');
660:
661:     with ABSQuery85 do
662:     begin
663:         Close;
664:         SQL.Text:='Select * from Classifica Order By "COGNOME","CODDSPL"';
665:         ExecSQL;
666:         Open;
667:     end;
668:     ABSTable15.Close; //30/04/15 18:06 devo cancellare per poter sostituire
669:     ABSTable15.EmptyTable; //con i dati sortati e senza problemi di indice
670:     ABSTable15.Open;
671:     ABSTable15.Last;
672:     ABSQuery85.First;
673:     qtrl:=ABSQuery85.RecordCount;
674:     while not ABSQuery85.EOF do //copia il query ordinato nella Tabella 15
675:     begin
676:         ABSTable15.Insert;
677:         ABSTable15.FieldByName('NMR').AsInteger:=
678:         ABSQuery85.FieldByName('NMR').AsInteger;

```



```
680:         ABSQuery85.FieldName('CODGA').AsInteger;
681:     ABSTable15.FieldName('CODDSPL').AsInteger:=
682:         ABSQuery85.FieldName('CODDSPL').AsInteger;
683:     ABSTable15.FieldName('CODATL').AsInteger:=
684:         ABSQuery85.FieldName('CODATL').AsInteger;
685:     ABSTable15.FieldName('COGNOME').AsString:=
686:         ABSQuery85.FieldName('COGNOME').AsString;
687:     ABSTable15.FieldName('NMPET').AsInteger:=
688:         ABSQuery85.FieldName('NMPET').AsInteger;
689:     ABSTable15.FieldName('NMPOS').AsInteger:=
690:         ABSQuery85.FieldName('NMPOS').AsInteger;
691:     ABSTable15.FieldName('TAME1').AsDateTime:=
692:         ABSQuery85.FieldName('TAME1').AsDateTime;
693:     ABSTable15.FieldName('TBME2').AsDateTime:=
694:         ABSQuery85.FieldName('TBME2').AsDateTime;
695:     ABSTable15.FieldName('DQ').AsBoolean:=
696:         ABSQuery85.FieldName('DQ').AsBoolean;
697:     ABSTable15.FieldName('PEN').AsDateTime:=
698:         ABSQuery85.FieldName('PEN').AsDateTime;
699:     ABSTable15.FieldName('MANL').AsBoolean:=
700:         ABSQuery85.FieldName('MANL').AsBoolean;
701:     ABSTable15.FieldName('CAT').AsInteger:=
702:         ABSQuery85.FieldName('CAT').AsInteger;
703:     qtrl:=qtrl-1;
704:     ABSTable15.Post;
705:     ABSQuery85.Next;
706: end;
707: if qtrl>0 then ShowMessage('ABSQuery85 errore eof');
708:
709: if MessageDlg('Desideri esaminare un rapporto dei Dati Base per la Classifica ?'
710: ,mtconfirmation, [mbYes,mbNo], 0) = mrYes then
711:     begin
712:         frxReport8.ShowReport;
713:         if MessageDlg('Confermi i dati a base della classifica? '+#13+
714:         'In caso negativo ESCI e correggi le scelte fatte.',
715:         mtconfirmation, [mbYes,mbNo], 0) = mrNO then
716:             begin
717:                 ABSTable15.Close;
718:                 ABSTable15.EmptyTable;
719:                 Exit;
720:             end;
721:         end;
722:
723:     RadioButton3.Color:= clLime;
724:     Calcola(Self);
725:     Exit;
726: end;
727: end;
728:
729: procedure TForm8.Button7Click(Sender: TObject);
730: var
731:     dspls, garesi, trovato: Boolean;
732:     qtrl, qtrl2, cgara, cdspl: Integer;
733: begin
734:     ClassDp:=True;
735:     Form2.ABSTable3.Close;
736:     Form2.ABSTable3.EmptyTable; //resetta la tabella gare selezionate
737:     Form2.ABSTable3.Open;      // e carica la Tabella con le Gare scelte per
738:     for i := 0 to High(ArGare) do //la ricerca per discipline
739:         if ArGare[i]=0 then
740:             begin
741:                 mom := i-1; //mom = all'ultimo elemento dell'array memorizzato
742:                 Break;
743:             end;
744:         if mom=-1 then
745:             begin
746:                 ShowMessage('Non vi sono elementi memorizzati. E' ripetibile.');
```

```

748:         Exit;
749:     end;
750: for i:= 0 to mom do      //crea la tabella 3
751:     begin
752:         idx:=ArGare[i]; //Trasferisce i dati dall'Array alla Tabella
753:         Form2.ABSTable3.Insert;
754:         Form2.ABSTable3.FieldName('Codice').AsInteger:=idx;
755:         garesi:=Form2.ABSTable5.Locate('CODGA',idx,[]);
756:         Form2.ABSTable3.FieldName('Descrizione').AsString:=
757:             Form2.ABSTable5.FieldName('LUOGO').AsString;
758:         Form2.ABSTable3.Post;
759:     end;
760:
761:
762: Form2.ABSTable3D.Close;
763: Form2.ABSTable3D.EmptyTable; //resetta la tabella gare selezionate
764: Form2.ABSTable3D.Open;
765: for i := 0 to High(ArDspl) do
766:     if ArDspl[i]=0 then
767:         begin
768:             momd := i-1; //mom = all'ultimo elemento dell'array memorizzato
769:             Break;
770:         end;
771:     if momd=-1 then
772:         begin
773:             ShowMessage('Non vi sono elementi memorizzati ! - ripeti');
774:             Exit;
775:         end;
776: for i:= 0 to momd do      //crea la tabella 3D
777:     begin
778:         idx:=ArDspl[i]; //Trasferisce i dati dall'Array alla Tabella
779:         Form2.ABSTable3D.Insert;
780:         Form2.ABSTable3D.FieldName('CODDSPL').AsInteger:=idx;
781:         dspls:=Form2.ABSTable4.Locate('CODDSPL',idx,[]);
782:         Form2.ABSTable3D.FieldName('DESCR').AsString:=
783:             Form2.ABSTable4.FieldName('DESCR').AsString;
784:         Form2.ABSTable3D.FieldName('TMAX').AsDateTime:=
785:             Form2.ABSTable4.FieldName('TMAX').AsDateTime;
786:         Form2.ABSTable3D.Post;
787:     end;
788:
789: DataSource8.DataSet:=Form2.ABSTable3D;      //display gare su dbgrid
790: DBGrid8.DataSource:=DataSource8;
791: DBGrid8.Width:=376;
792: DBGrid8.Columns[0].Width:=44;
793: DBGrid8.Columns[1].Width:=291;
794: if MessageDlg('Confermi di effettuare la classifica su questi elementi ?',
795:     mtconfirmation, [mbYes,mbNo], 0) = mrNo then
796:     begin
797:         Form2.ABSTable3D.Close;
798:         Form2.ABSTable3D.EmptyTable;
799:         Shape1.Brush.Color := clRed;
800:         SetLength(ArDspl,0); //azzerare l'array delle Discipline non confermando la scelta
801:         SetLength(ArDspl,20);
802:         Exit;
803:     end;
804:     if comboBox1.ItemIndex=-1 then
805:         begin
806:             ShowMessage('Devi scegliere una Categoria');
807:             Exit;
808:         end else
809:         begin
810:             if MessageDlg('Confermi di effettuare la classifica per la Categoria '+comboBox1.Text+
811:                 ' ?',
812:                 mtconfirmation, [mbYes,mbNo], 0) = mrNo then
813:                 begin
814:                     Shape2.Brush.Color := clRed;

```

```
815:         end ;
816:         Shape3.Brush.Color := clLime;
817:     end;
818:         //inizia a raccogliere i record da classificare
            //lavori in corso
819:
820:         if ABSTable15.RecordCount>0 then //Azzerò la tabella Classifica
821:             begin // preesistente per non causare
822:                 ABSTable15.Close; // problemi di indice unique
823:                 ABSTable15.EmptyTable;
824:                 ABSTable15.Open;
825:             end;
826:             ABSTable15.Last;
827:             Form2.ABSTable3.First; //Form2.ABSTable3 è una tabella con gare
828:             qtrl:=Form2.ABSTable3.RecordCount;
829:         while not Form2.ABSTable3.EOF do //per ogni gara selezionata
830:             begin
831:                 Cgara:=Form2.ABSTable3.FieldByName('Codice').AsInteger;
832:                 qtrl2:=Form2.ABSTable3D.RecordCount;
833:                 Form2.ABSTable3D.First;
834:                 while not Form2.ABSTable3D.EOF do //per ogni disciplina selezionata
835:                     begin
836:                         Cdspl:=Form2.ABSTable3D.FieldByName('CODDSPL').AsInteger;
837:                         with ABSQuery85 do //Estrae dalle Gare le discipline da classif
838:                             begin //Costruisce Classifica con tutti i record gare e discipline
839:                                 Close; //selezionate e gli Atleti che le hanno in comune
840:                                 SQL.Text:='INSERT INTO Classifica SELECT * From Gara where CODGA='
841:                                     +QuotedStr(IntToStr(Cgara))+ ' AND CODDSPL='+QuotedStr(IntToStr(Cdspl))
842:                                     + 'AND CAT =' + quotedstr(comboBox1.text);
843:                                 ExecSQL;
844:                             end;
845:
846:                             qtrl2:=qtrl2-1;
847:                             Form2.ABSTable3D.Next
848:                         end;
849:                     if qtrl2>0 then ShowMessage('ABSTable3D errore eof');
850:                     qtrl:=qtrl-1;
851:                     Form2.ABSTable3.Next;
852:                 end;
853:                 ComboBox1.ItemIndex:=-1;
854:                 if qtrl>0 then ShowMessage('ABSTable3 errore eof');
855:
856:                 with ABSQuery85 do
857:                     begin
858:                         Close;
859:                         SQL.Text:='Select * from Classifica Order By "COGNOME","CODDSPL"';
860:                         ExecSQL;
861:                         Open;
862:                     end;
863:                 ABSTable15.Close; //30/04/15 18:06 devo cancellare per poter sostituire
864:                 ABSTable15.EmptyTable; //con i dati sortati e senza problemi di indice
865:                 ABSTable15.Open;
866:                 ABSTable15.Last;
867:                 ABSQuery85.First;
868:                 qtrl:=ABSQuery85.RecordCount;
869:                 while not ABSQuery85.EOF do //copia il query ordinato nella Tabella 15
870:                     begin
871:                         ABSTable15.Insert;
872:                         ABSTable15.FieldByName('NMR').AsInteger:=
873:                             ABSQuery85.FieldByName('NMR').AsInteger;
874:                         ABSTable15.FieldByName('CODGA').AsInteger:=
875:                             ABSQuery85.FieldByName('CODGA').AsInteger;
876:                         ABSTable15.FieldByName('CODDSPL').AsInteger:=
877:                             ABSQuery85.FieldByName('CODDSPL').AsInteger;
878:                         ABSTable15.FieldByName('CODATL').AsInteger:=
879:                             ABSQuery85.FieldByName('CODATL').AsInteger;
880:                         ABSTable15.FieldByName('COGNOME').AsInteger:=
881:                             ABSQuery85.FieldByName('COGNOME').AsInteger;
```

```
882:         ABSTable15.FieldName('NMPET').AsInteger:=
883:             ABSQuery85.FieldName('NMPET').AsInteger;
884:         ABSTable15.FieldName('NMPOS').AsInteger:=
885:             ABSQuery85.FieldName('NMPOS').AsInteger;
886:         ABSTable15.FieldName('TAME1').AsDateTime:=
887:             ABSQuery85.FieldName('TAME1').AsDateTime;
888:         ABSTable15.FieldName('TBME2').AsDateTime:=
889:             ABSQuery85.FieldName('TBME2').AsDateTime;
890:         ABSTable15.FieldName('DQ').AsBoolean:=
891:             ABSQuery85.FieldName('DQ').AsBoolean;
892:         ABSTable15.FieldName('PEN').AsDateTime:=
893:             ABSQuery85.FieldName('PEN').AsDateTime;
894:         ABSTable15.FieldName('MANL').AsBoolean:=
895:             ABSQuery85.FieldName('MANL').AsBoolean;
896:         ABSTable15.FieldName('CAT').AsInteger:=
897:             ABSQuery85.FieldName('CAT').AsInteger;
898:         qtrl:=qtrl-1;
899:         ABSTable15.Post;
900:         ABSQuery85.Next;
901:     end;
902:     if qtrl>0 then ShowMessage('ABSQuery85 errore eof');
903:
904:     if MessageDlg('Desideri esaminare un rapporto dei Dati Base per la Classifica ?'
905: ,mtconfirmation, [mbYes,mbNo], 0) = mrYes then
906:         begin
907:             frxReport8.ShowReport;
908:             if MessageDlg('Confermi i dati a base della classifica? '+#13+
909: 'In caso negativo ESCI e correggi le scelte fatte.',
910: mtconfirmation, [mbYes,mbNo], 0) = mrNO then
911:                 begin
912:                     ABSTable15.Close;
913:                     ABSTable15.EmptyTable;
914:                     Exit;
915:                 end;
916:             end;
917:
918:             Calcola(Self);
919:             Exit;
920:         end;
921:
922: procedure TForm8.Calcola(Sender: TObject);
923: var
924: t1,t2,p1,tmx,tmcn:TDateTime;
925: h1,m1,s1,z1,h2,m2,s2,z2,h3,m3,s3,z3:Word;
926: sql,esclusioni,trovato,ATLsi:Boolean;
927: esclusi:TStringList;
928: ca,dp,ga,qtrl,ct,i,imax,temp2,tt,tt2:Integer;
929: dacancA:array of smallInt;
930: dacancD:array of smallInt;
931: dacancG:array of smallInt;
932: cgn:string;
933: RnkMax,rn,CODATL16:Integer;           //Fase segnalazione atleti in difetto
934: begin
935:     trovato:=False;
936:     SetLength(dacancA,0);
937:     SetLength(dacancA,300);
938:     SetLength(dacancD,0);
939:     SetLength(dacancD,300);
940:     SetLength(dacancG,0);
941:     SetLength(dacancG,300);
942:     esclusi:=TStringList.Create;//NOTA BENE L'ESCLUSIONE VALE SOLO NEL RUN IN CORSO
943:     ABSTable16.Close;
944:     ABSTable16.EmptyTable;
945:     ABSTable15.Open;
946:     ABSTable15.Edit;
947:     ABSTable15.Last;    //per ottenere un conteggio corretto ???!!!
948:     ABSTable15.First;
```

```

950: qtrl:=ABSTable15.RecordCount;
951: i:=1;
952: esclusioni:=False;
953:
954:
955: while not ABSTable15.Eof do //memorizzazione e correzione
956: begin
957:   t1:=TimeOf(ABSTable15.FieldByName('TAME1').AsDateTime);
958:   t2:=TimeOf(ABSTable15.FieldByName('TBME2').AsDateTime);
959:   sql:=ABSTable15.FieldByName('DQ').AsBoolean;
960:   ct:=ABSTable15.FieldByName('CAT').AsInteger;
961:   ca:= ABSTable15.FieldByName('CODATL').AsInteger; //memorizza i record
962:   cgn:=ABSTable15.FieldByName('COGNOME').AsString;
963:   dp:= ABSTable15.FieldByName('CODDSPL').AsInteger;
964:   ga:=ABSTable15.FieldByName('CODGA').AsInteger;
965:   if ct=0 then
966:     begin
967:       ShowMessage('Attenzione l''Atleta '+cgn+' codice '+IntToStr(ca)+' gara '
968:         +IntToStr(ga)+' disciplina '+IntToStr(dp)+#13+#13+
969:         ' non ha la categoria. Correggi');
970:       Exit;
971:     end;
972:   if ((t1=0) and (t2=0))or(sql=True) then //Individue le ragioni
973:     begin
974:       esclusi.Add('Gara '+Inttostr(ga)+' codice: '+intToStr(ca)+' '+cgn+
975:         ' Disc. '+intToStr(dp));
976:       dacancA[i]:=ca; //memorizza i dati da-completare;
977:       dacancD[i]:=dp;
978:       dacancG[i]:=ga;
979:       i:=i+1;
980:       esclusioni:=True;
981:     end;
982:   qtrl:=qtrl-1;
983:   imax:=i;
984:
985:   ABSTable15.Next;
986: end;
987: if qtrl>0 then ShowMessage('ABSTable15 errore eof');
988:
989: if (esclusi.Count<50) and (esclusi.Count>0) then
990:   ShowMessage(' Atleti con dati mancanti' +#13+#13+ esclusi.Text+#13+#13+
991:     ' Dati salvati su Mancanti.txt') else
992:   begin
993:     if esclusi.count>0 then
994:       ShowMessage('Lista Atleti con dati mancanti, salvata in "Mancanti.txt".');
995:   end;
996: esclusi.SaveToFile('.\Mancanti.txt');
997: esclusi.Clear;
998: //Calcolo dei tempi usa Selezionati T3 e classifiche T15
999: Form2.ABSTable3.Open;
1000: Form2.ABSTable3.First;
1001: while not Form2.ABSTable3.eof do //Quì inizia il giro delle gare scelte
1002: begin
1003:   ga:=Form2.ABSTable3.FieldByName('Codice').AsInteger;
1004:   ABSTable18.Close; //azzerla la T18 per riceve i dati della nuova gara
1005:   ABSTable18.EmptyTable;
1006:   // DataSourceQry.DataSet:=ABSQuery85;
1007:   with ABSQuery85 do //seleziona tutte le discipline della gara scelta
1008:     begin
1009:       Close; //quì nasce la ABSTable18 con tutte le discipline di una gara
1010:       SQL.Text:='INSERT INTO Temp8 Select * from Classifica where CODGA = '
1011:         + QuotedStr(IntToStr(ga))+ ' Order By "CODDSPL"';
1012:       ExecSQL;
1013:       //Open;
1014:     end; //creata la Temp8 = le discipline della gara
1015:     //Crea le discipline da classificare per la gara scelta
1016:

```

```

1018: Begin
1019:   with ABSQuery80 do
1020:     begin
1021:       Close;
1022:       if RadioButton1.Checked=True then
1023:         SQL.Text:= 'Select CODDSPL from Base where CODGA = '
1024:           + QuotedStr(IntToStr(ga))+ ' Order By "CODDSPL"';
1025:       if RadioButton2.Checked=True then
1026:         SQL.Text:= 'Select CODDSPL from GRPA where CODGA = '
1027:           + QuotedStr(IntToStr(ga))+ ' Order By "CODDSPL"';
1028:       if RadioButton3.Checked=True then
1029:         SQL.Text:= 'Select CODDSPL from GRPB where CODGA = '
1030:           + QuotedStr(IntToStr(ga))+ ' Order By "CODDSPL"';
1031:       ExecSQL;
1032:       Open;
1033:     end;
1034:   end else
1035:   begin
1036:     with ABSQuery80 do //per il caso della classifica per discipline
1037:       begin
1038:         Close;
1039:         SQL.Text:= 'Select distinct CODDSPL from Classifica where CODGA = '
1040:           + QuotedStr(IntToStr(ga))+ ' Order By "CODDSPL"';
1041:         ExecSQL;
1042:         Open;
1043:       end;
1044:     end;
1045:
1046:     ABSTable18.Open;
1047:     TDateTimeField(Form8.ABSTable18.FieldByName('TAME1')).DisplayFormat:='nn:ss,zzz';
1048:     TDateTimeField(Form8.ABSTable18.FieldByName('TBME2')).DisplayFormat:='nn:ss,zzz';
1049:     TDateTimeField(Form8.ABSTable18.FieldByName('PEN')).DisplayFormat:='nn:ss,zzz';
1050:
1051:     // T18 tutte le dspl di una gara
1052:
1053:     ABSQuery80.First; // Tutte le dspl del gruppo scelto
1054:     while not ABSQuery80.eof do
1055:       begin
1056:         dp:=ABSQuery80.FieldByName('CODDSPL').AsInteger;
1057:         with ABSQuery83 do //Tutte le discipline
1058:           begin
1059:             Close; //la ABSTable18 fornisce i dati per la disciplina
1060:             SQL.Text:='Select * from Temp8 where CODDSPL = '+ QuotedStr(IntToStr(dp))+
1061:               ' Order By "CODATL"';
1062:             ExecSQL;
1063:             Open;
1064:           end; //Crea Query83 = i record di una disciplina per volta
1065:           //quì ci sono tutti i record relativi ad una gara su una disciplina
1066:           //con i tempi di ciascun Atleta e si può valutare il vincitore
1067:           //ATTENZIONE PUO' CAPITARE CHE NESSUN ATLETA è ISCRITTO
1068:           if ABSQuery83.RecordCount=0 then
1069:             begin
1070:               ShowMessage('Attenzione nessun atleta iscritto alla disciplina '+IntToStr(dp)
+#13+
1071:                 'nella Gara : '+inttostr(ga));
1072:               ABSQuery80.Next;
1073:               Continue;
1074:             end;
1075:
1076:           ABSTable16.Close;
1077:           ABSTable16.EmptyTable;
1078:           ABSTable16.Open;
1079:           TDateTimeField(Form8.ABSTable16.FieldByName('TAME1')).
1080:             DisplayFormat:='nn:ss,zzz';
1081:           TDateTimeField(Form8.ABSTable16.FieldByName('TBME2')).
1082:             DisplayFormat:='nn:ss,zzz';
1083:           TDateTimeField(Form8.ABSTable16.FieldByName('PEN')).
1084:             DisplayFormat:='nn:ss,zzz';

```



```

1085:      TDateTimeField(Form8.ABSTable16.FieldName('TMC')).
1086:          DisplayFormat:='nn:ss,zzz';
1087:      ABSQuery83.Last; //CREA LA T16 DOVE SI ASSEGNANO I TEMPI e i DQ
1088:      ABSQuery83.First;
1089:      qtrl:=ABSQuery83.RecordCount;
1090:      while not ABSQuery83.EOF do //query ordinato nella Tabella 16 Una Gara
1091:          begin
1092:              ABSTable16.Insert;
1093:              ABSTable16.FieldName('NMR').AsInteger:=
1094:                  ABSQuery83.FieldName('NMR').AsInteger;           // Una disciplna
1095:              ABSTable16.FieldName('CODGA').AsInteger:=
1096:                  ABSQuery83.FieldName('CODGA').AsInteger;
1097:              ABSTable16.FieldName('CODDSPL').AsInteger:=
1098:                  ABSQuery83.FieldName('CODDSPL').AsInteger;
1099:              ABSTable16.FieldName('CODATL').AsInteger:=
1100:                  ABSQuery83.FieldName('CODATL').AsInteger;
1101:              ABSTable16.FieldName('COGNOME').AsString:=
1102:                  ABSQuery83.FieldName('COGNOME').AsString;
1103:              ABSTable16.FieldName('NMPET').AsInteger:=
1104:                  ABSQuery83.FieldName('NMPET').AsInteger;
1105:              ABSTable16.FieldName('NMPOS').AsInteger:=
1106:                  ABSQuery83.FieldName('NMPOS').AsInteger;
1107:              ABSTable16.FieldName('TAME1').AsDateTime:=
1108:                  ABSQuery83.FieldName('TAME1').AsDateTime;
1109:              ABSTable16.FieldName('TBME2').AsDateTime:=
1110:                  ABSQuery83.FieldName('TBME2').AsDateTime;
1111:              ABSTable16.FieldName('DQ').AsBoolean:=
1112:                  ABSQuery83.FieldName('DQ').AsBoolean;
1113:              ABSTable16.FieldName('PEN').AsDateTime:=
1114:                  ABSQuery83.FieldName('PEN').AsDateTime;
1115:              ABSTable16.FieldName('MANL').AsBoolean:=
1116:                  ABSQuery83.FieldName('MANL').AsBoolean;
1117:              ABSTable16.FieldName('CAT').AsInteger:=
1118:                  ABSQuery83.FieldName('CAT').AsInteger;
1119:              qtrl:=qtrl-1;
1120:              ABSTable16.Post;
1121:              ABSQuery83.Next;
1122:          end;
1123:      if qtrl>0 then ShowMessage('ABSQuery83_2 errore eof');
1124:
1125:
1126:      RnkMax:=ABSTable16.RecordCount;
1127:      rn:=RnkMax;
1128:
1129:      ABSTable16.First;
1130:          //ricava TMAX e il cod. disciplina
1131:      dp:= ABSTable16.FieldName('CODDSPL').AsInteger;
1132:      trovato:=Form2.ABSTable4.Locate('CODDSPL',dp,[]);
1133:      if trovato =True then tmx:=Form2.ABSTable4.FieldName('TMAX').AsDateTime
1134:          else
1135:              begin
1136:                  ShowMessage('ERRORE, Disciplina o TMAX non trovato - ca: '+inttostr(ca)+' d
1137: p : '+inttostr(dp));
1138:                  Exit;
1139:              end;
1140:      while not ABSTable16.eof do //scorre tutti i record e inserisce il tempo medio
1141:          begin
1142:              ABSTable16.Edit;
1143:              t1:=ABSTable16.FieldName('TAME1').AsDateTime;
1144:              t2:=ABSTable16.FieldName('TBME2').AsDateTime;
1145:              p1:=ABSTable16.FieldName('PEN').AsDateTime;
1146:              sql:=ABSTable16.FieldName('DQ').AsBoolean;
1147:              ca:= ABSTable16.FieldName('CODATL').AsInteger;
1148:              // controlla proprietà dei tempi
1149:              DecodeTime(t1,h1,m1,s1,z1);
1150:              DecodeTime(t2,h2,m2,s2,z2);
1151:              if (h1<0) or (h2<0) then

```

```

1152:         ShowMessage('Impossibile classificare: presenti valori'+
1153:         ' incompatibili per l''Atleta : '+ inttostr(ca));
1154:         Exit;
1155:     end;
1156:         //tempo medio
1157:     if ((t1=0)and(t2<>0)) then
1158:     begin
1159:         ABSTable16.FieldName('TMC').AsDateTime:= (t2+p1);
1160:         TDateTimeField(Form8.ABSTable16.FieldName('TMC')).DisplayFormat:='nn:ss,
zzz';
1161:     end;
1162:
1163:     if ((t1<>0)and(t2=0)) then
1164:     begin
1165:         ABSTable16.FieldName('TMC').AsDateTime:= (t1+p1);
1166:         TDateTimeField(Form8.ABSTable16.FieldName('TMC')).DisplayFormat:='nn:ss
s,zzz';
1167:     end;
1168:
1169:         // inserisce il valore medio calcolato
1170:     if ((t1<>0)and(t2<>0))then
1171:     begin
1172:         tmcn:= (((t1+t2)/2)+p1);
1173:         ABSTable16.FieldName('TMC').AsDateTime:= tmcn;
1174:         TDateTimeField(Form8.ABSTable16.FieldName('TMC')).DisplayFormat:='nn:ss
,zzz';
1175:     end;
1176:     ABSTable16.Post;
1177:     ABSTable16.Next; //nuovo atleta
1178: end;
1179:
1180:                                     //Assegnazione Del Ranking
1181:     with ABSQuery85 do
1182:     begin
1183:         //Ordina per tempi crescenti
1184:         Close;
1185:         SQL.Text:='Select * from Ranking Order By "TMC"';
1186:         ExecSQL;
1187:         Open;
1188:     end;
1189:     ABSTable16.Close;
1190:     ABSTable16.EmptyTable;
1191:     ABSTable16.Open;
1192:     ABSTable16.Edit;
1193:     ABSQuery85.Last;
1194:     ABSQuery85.First;
1195:     qtrl:=ABSQuery85.RecordCount;
1196:     while not ABSQuery85.EOF do
1197:     begin
1198:         //Ordinata per tempi crescenti
1199:         ABSTable16.Insert;
1200:         ABSTable16.FieldName('NMR').AsInteger:=
1201:             ABSQuery85.FieldName('NMR').AsInteger;
1202:         ABSTable16.FieldName('CODGA').AsInteger:=
1203:             ABSQuery85.FieldName('CODGA').AsInteger;
1204:         ABSTable16.FieldName('CODDSPL').AsInteger:=
1205:             ABSQuery85.FieldName('CODDSPL').AsInteger;
1206:         ABSTable16.FieldName('CODATL').AsInteger:=
1207:             ABSQuery85.FieldName('CODATL').AsInteger;
1208:         ABSTable16.FieldName('COGNOME').AsString:=
1209:             ABSQuery85.FieldName('COGNOME').AsString;
1210:         ABSTable16.FieldName('NMPET').AsInteger:=
1211:             ABSQuery85.FieldName('NMPET').AsInteger;
1212:         ABSTable16.FieldName('NMPOS').AsInteger:=
1213:             ABSQuery85.FieldName('NMPOS').AsInteger;
1214:         ABSTable16.FieldName('TAME1').AsDateTime:=
1215:             ABSQuery85.FieldName('TAME1').AsDateTime;
1216:         ABSTable16.FieldName('TAME2').AsDateTime:=

```


128 You created this PDF from an application that is not licensed to print to novaPDF printer (<http://www.novapdf.com>)

```
1285:         ABSTable16.FieldName('NMPOS').AsInteger:=
1286:             ABSQuery85.FieldName('NMPOS').AsInteger;
1287:         ABSTable16.FieldName('TAME1').AsDateTime:=
1288:             ABSQuery85.FieldName('TAME1').AsDateTime;
1289:         ABSTable16.FieldName('TBME2').AsDateTime:=
1290:             ABSQuery85.FieldName('TBME2').AsDateTime;
1291:         ABSTable16.FieldName('DQ').AsBoolean:=
1292:             ABSQuery85.FieldName('DQ').AsBoolean;
1293:         ABSTable16.FieldName('PEN').AsDateTime:=
1294:             ABSQuery85.FieldName('PEN').AsDateTime;
1295:         ABSTable16.FieldName('MANL').AsBoolean:=
1296:             ABSQuery85.FieldName('MANL').AsBoolean;
1297:         ABSTable16.FieldName('CAT').AsInteger:=
1298:             ABSQuery85.FieldName('CAT').AsInteger;
1299:         ABSTable16.FieldName('TMC').AsDateTime:=
1300:             ABSQuery85.FieldName('TMC').AsDateTime;
1301:         ABSTable16.FieldName('RNK').AsInteger:=
1302:             ABSQuery85.FieldName('RNK').AsInteger;
1303:         ABSTable16.FieldName('TMC').AsDateTime:=
1304:             ABSQuery85.FieldName('TMC').AsDateTime;
1305:
1306:         qtrl:=qtrl-1;
1307:         ABSTable16.Post;
1308:         ABSQuery85.Next;
1309:     end;
1310:
1311:         // Aggiungo il nome e nazione
1312:     ABSTable16.First;
1313:     while not ABSTable16.EOF do
1314:     begin
1315:         CODATL16:=ABSTable16.FieldName('CODATL').AsInteger;
1316:         ATLSi:=Form2.ABSTable2.Locate('CODATL',CODATL16,[] );
1317:         if ATLSi=False then //si blocca sul record con pari codice DSPL
1318:         begin
1319:             ShowMessage('Errore Atleta non trovato nell''Anagrafica');
1320:             Exit;
1321:         end;
1322:         ABSTable16.Edit;
1323:         ABSTable16.FieldName('NOME').AsString:=Form2.ABSTable2.FieldName('NOME').AsStri
ng;
1324:         ABSTable16.FieldName('NAZIONE').AsString:=Form2.ABSTable2.FieldName('NAZIONE').
AsString;
1325:         ABSTable16.Post;
1326:         ABSTable16.NEXT
1327:     end;
1328:     // Added by Manlio Laschena 17/05/2015 13:05:35 ci sono i DATI dei punti
1329:         //TRASCRIZIONE su T16T STORICO
1330:
1331:     Form2.ABSTable16T.TableName:='T16T';
1332:     Form2.ABSTable16T.Open;
1333:     ABSTable16.First;
1334:     while not ABSTable16.EOF do
1335:     begin
1336:         //Raccolta dati di codice
1337:         ca:=ABSTable16.FieldName('CODATL').AsInteger;
1338:         ga:=ABSTable16.FieldName('CODGA').AsInteger;
1339:         dp:=ABSTable16.FieldName('CODDSPL').AsInteger;
1340:         trovato:=Form2.ABSTable16T.Locate('CODATL;CODGA;CODDSPL',VarArrayOf([IntToStr(ca),
IntToStr(ga),IntToStr(dp)]),[]);
1341:         if trovato=False then Form2.ABSTable16T.Insert;
1342:         Form2.ABSTable16T.Edit;
1343:         Form2.ABSTable16T.FieldName('CODATL').AsInteger:=ca;
1344:         Form2.ABSTable16T.FieldName('CODGA').AsInteger:=ga;
1345:         Form2.ABSTable16T.FieldName('CODDSPL').AsInteger:=dp;
1346:         Form2.ABSTable16T.FieldName('COGNOME').AsString:=ABSTable16.FieldName('COGNOME').AsString;
1347:         Form2.ABSTable16T.FieldName('NOME').AsString:=ABSTable16.FieldName('NOME').A
```

```
1348:         Form2.ABTable16T.FieldByName('NAZIONE').AsString:=ABTable16.FieldByName('NAZIONE').AsString;
1349:         Form2.ABTable16T.FieldByName('TMC').AsDateTime:=ABTable16.FieldByName('TMC').AsDateTime;
1350:         TDateTimeField(Form2.ABTable16T.FieldByName('TMC')).DisplayFormat:='nn:ss,zzz';
1351:         Form2.ABTable16T.FieldByName('RNK').AsInteger:=ABTable16.FieldByName('RNK').AsInteger;
1352:         Form2.ABTable16T.FieldByName('DQ').AsBoolean:=ABTable16.FieldByName('DQ').AsBoolean;
1353:         Form2.ABTable16T.FieldByName('CAT').AsInteger:=ABTable16.FieldByName('CAT').AsInteger;
1354:         //Aggiunge dati mancanti
1355:         trovato:=Form2.ABTable4.Locate('CODDSPL',dp,[]); //Descrizione
1356:         if trovato=True then
1357:             Form2.ABTable16T.FieldByName('DESCR').AsString:= Form2.ABTable4.FieldByName('DESCR').AsString else
1358:             begin
1359:                 ShowMessage('Errore Disciplina non trovata nell''Anagrafica @T16T');
1360:                 Exit;
1361:             end;
1362:         trovato:=Form2.ABTable5.Locate('CODGA',ga,[]); //Data Gara
1363:         if trovato=True then
1364:             Form2.ABTable16T.FieldByName('DATAGA').AsDateTime:= Form2.ABTable5.FieldByName('DATAGA').AsDateTime else
1365:             begin
1366:                 ShowMessage('Errore Gara non trovata nell''Anagrafica @T16T');
1367:                 Exit;
1368:             end;
1369:
1370:         Form2.ABTable16T.Post;
1371:         ABTable16.Next;
1372:     end;
1373:     //determina il punteggio totale per atleta
1374:
1375:     if MessageDlg('Desideri esaminare la Classifica della Disciplina '+
1376:         QuotedStr(IntToStr(dp))+#13+'Gara : '+QuotedStr(IntToStr(ga))+ ' ?',mtconfirmation,
1377:         [mbYes,mbNo], 0) = mrYes then
1378:         begin
1379:             frxReport16.Variables['max']:= quotedstr(TimeToStr(tmx));
1380:             frxReport16.ShowReport; //Classifiche per singole discipline
1381:             end; //Dati da ABTable16 (Ranking)
1382:
1383:     with ABSQueryR do
1384:         begin
1385:             Close;
1386:             SQL.Text:= 'INSERT INTO R16R SELECT * FROM Ranking';
1387:             ExecSQL;
1388:         end;
1389:     ABSQuery80.Next; //Altra disciplina
1390: end;
1391: //qui finisce le discipline della Gara
1392: if MessageDlg('Desideri esaminare i risultati complessivi per le discipline selezionate '+
1393:     'della Gara : '+QuotedStr(IntToStr(ga))+ ' ?',mtconfirmation, [mbYes,mbNo], 0) = mrYes then
1394:     frxReportR.ShowReport;
1395:
1396:
1397:     Form2.ABTable3.Next;//adesso inserisce una nuova gara
1398: end;
1399:
1400:
1401: if (ClassDp=False) then
1402:     begin
1403:         if MessageDlg('Desideri esaminare la Classifica Riepilogativa '+
1404:             'della Gara : '+QuotedStr(IntToStr(ga))+#13+'per tutte le discipline praticate ?',mt
```

```

1405:     begin
1406:         with ABSQueryRep do
1407:             begin
1408:                 Close;
1409:                 SQL.Text:='Select * from T16T Where CODGA = '+ QuotedStr(IntToStr(ga))+ ' Order by
CODDSPL, RNK desc';
1410:                 ExecSQL;
1411:                 Open;
1412:             end;
1413:
1414:             frxReport16T.ShowReport; //Classifica riepilogativa gara
1415:             with ABSQuery85 do //Dati da QueryRep
1416:                 begin
1417:                     Close; //tabelle con nome numerico devono iniziare con una lettera
1418:                     SQL.Text:='Select CODATL,COGNOME,DATAGA, sum(RNK) AS TOTRNK from T16T Where CODGA
= '+'
1419:                     QuotedStr(IntToStr(ga))+ ' Group By CODATL,COGNOME,DATAGA Order by TOTRNK desc';
1420:                     ExecSQL;
1421:                     Open
1422:                 end;
1423:                 frxReport16T2.ShowReport; // Classifica riepilogativa punteggio
1424:             end; //Dati da Query85 >> 16T (T16T)
1425:         end;
1426:
1427:
1428:     if ClassDp=False then
1429:         begin
1430:
1431:             if MessageDlg('Desideri esaminare Il Punteggio complessivo raggiunto '+#13
1432:                 +' per le Gare selezionate ?'+#13+
1433:                 'NB. Non cambia per una sola Gara',mtconfirmation, [mbYes,mbNo], 0) = mrYes then
1434:                 begin
1435:                     ABSTable19.Open;
1436:                     ABSTable19.Edit;
1437:                     ABSTable19.First;
1438:                     Form2.ABSTable3.First;
1439:                     while not Form2.ABSTable3.Eof do
1440:                         begin //Colleziona i dati di tutte le gare scelte
1441:                             ga:=Form2.ABSTable3.FieldName('codice').AsInteger;
1442:                             with ABSQueryRep do
1443:                                 begin
1444:                                     Close;
1445:                                     SQL.Text:='Select CODATL,COGNOME, NOME,NAZIONE, SUM(RNK) AS TOTRNK from T16T W
here CODGA = '
1446:                                     + QuotedStr(IntToStr(ga))+ 'AND CAT ='+QuotedStr(IntToStr(ct))+ ' Group By CODA
TL,COGNOME,NOME,NAZIONE ';
1447:                                     ExecSQL;
1448:                                     Open;
1449:                                 end;
1450:                                 ABSQueryRep.First;
1451:                                 while not ABSQueryRep.eof do
1452:                                     begin
1453:                                         temp2:=ABSQueryRep.FieldName('CODATL').AsInteger;
1454:                                         trovato:=ABSTable19.Locate('CODATL',temp2,[]);
1455:                                         if trovato=False then
1456:                                             ABSTable19.Insert;
1457:                                             ABSTable19.Edit;
1458:                                             ABSTable19.FieldName('CODATL').AsInteger:=ABSQueryRep.FieldName('CODATL')
.AsInteger;
1459:                                             ABSTable19.FieldName('COGNOME').AsString:=ABSQueryRep.FieldName('COGNOME')
.AsString;
1460:                                             ABSTable19.FieldName('NOME').AsString:=ABSQueryRep.FieldName('NOME').AsSt
ring;
1461:                                             ABSTable19.FieldName('NAZIONE').AsString:=ABSQueryRep.FieldName('NAZIONE')
.AsString;
1462:                                             tt:= ABSQueryRep.FieldName('TOTRNK').AsInteger;
1463:                                             tt2:=ABSTable19.FieldName('RNK').AsInteger;
1464:                                             You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)

```

```
1465:         ABSTable19.Post;
1466:         ABSQueryRep.Next;
1467:     end;
1468:     Form2.ABSTable3.Next;
1469: end;
1470:     with ABSQueryRep do
1471:     begin
1472:         Close;
1473:         SQL.Text:= 'Select * from totale Order by RNK desc';
1474:         ExecSQL;
1475:         Open;
1476:     end;
1477:
1478:     frxReportTot.ShowReport;
1479:         //dati da QueryRep >> Totale T19
1480: end;
1481: end;
1482:
1483: ClassDp:=False;
1484: RadioButton1.Color:= clRed;
1485: RadioButton2.Color:= clRed;
1486: RadioButton3.Color:= clRed;
1487: Shape1.Brush.Color := clRed; //Azzerata e resetta la scelta Gare nell'Array
1488: SetLength(ArGare,0);
1489: SetLength(ArGare,20);
1490: Shape2.Brush.Color := clRed; //Azzerata e resetta la scelta Discipline nell'Array
1491: SetLength(ArDspl,0);
1492: SetLength(ArDspl,20);
1493: Shape3.Brush.Color:= clRed;
1494: ComboBox1.ItemIndex:=-1;
1495: ABSTable19.Close;
1496: ABSTable19.EmptyTable;
1497: ABSTable18.Close;
1498: ABSTable18.EmptyTable;
1499:
1500: end;
1501:
1502: procedure TForm8.Esciclick(Sender: TObject); //Esci
1503: begin
1504:     Form2.ABSTable1.Open;
1505:     Form2.ABSTable1.Last;
1506:     if Form2.ABSTable1.FieldName('Password').AsString<>' ' then
1507:     Form2.ABSTable1.Insert;
1508:     Form2.ABSTable1.Edit;
1509:     Form2.ABSTable1.FieldName('OUT').AsDateTime:=NOW;
1510:     Form2.ABSTable1.Post;
1511:     Form2.ABSTable1.Close;
1512:     Form8.Close; // NB chiama il can close
1513:     ClassFlg:=True; //per consentire di monitorizzare che Form8 è closed
1514:     Form2.Visible:=True;
1515:     SetLength(ArGare,0);
1516:     SetLength(ArDspl,0);
1517:     SetLength(ArCat,0);
1518:     Form2.ABSTable3.Close;
1519:     Form2.ABSTable3.EmptyTable; //resetta la tabella Selezionati
1520:     ABSQuery85.Close;
1521:     ABSTable15.Close; //per consentire lo svuotamento messo su ObjInsp Exclusive=True
1522:     ABSTable15.EmptyTable; //resetta la tabella Classifica
1523:     ABSTable16R.Close;
1524:     ABSTable16R.EmptyTable;
1525: end;
1526: end.
```

```

1: unit Tempi;
2:
3: interface
4:
5: uses
6:   Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
7:   Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls;
8:
9: type
10:   TForm9 = class(TForm)
11:     Edit1: TEdit;
12:     Button1: TButton;
13:     Label1: TLabel;
14:     Label2: TLabel;
15:     Edit2: TEdit;
16:     Edit3: TEdit;
17:     Button3: TButton;
18:     RadioButton1: TRadioButton;
19:     RadioButton2: TRadioButton;
20:     Label3: TLabel;
21:     Label4: TLabel;
22:     Label5: TLabel;
23:     Label6: TLabel;
24:     procedure Button1Click(Sender: TObject);
25:     procedure Button3Click(Sender: TObject);
26:     procedure Edit2KeyPress(Sender: TObject; var Key: Char);
27:   private
28:     { Private declarations }
29:   public
30:     { Public declarations }
31:   end;
32:
33: var
34:   Form9: TForm9;
35:
36: implementation
37:
38: {$R *.dfm}
39: {$D+}
40: uses
41:   Gara;
42: procedure TForm9.Button1Click(Sender: TObject); //ACCETTA
43: begin
44:   if (Edit1.Text='') and (Edit2.Text='') then // 0 min 0 sec 0 ms
45:     begin
46:       ShowMessage('Nulla da accettare');
47:       if MessageDlg('Vuoi Uscire ?',
48:         mtconfirmation, [mbYes,mbNo], 0) = mrYes then
49:         begin
50:           Form9.Close;
51:           Form9.Release;
52:           end else Exit;
53:       end;
54:   if (Edit2.Text<>'') and (Length(Edit2.Text)<6) then
55:     begin
56:       ShowMessage('I secondi devono avere il formato YY.ZZZ');
57:       Exit;
58:     end;
59:
60:   Gara.rab1:=False;
61:   Gara.rab2:=False;
62:   if (Edit1.Text<>'') and (Edit2.Text='') then // 1 min 0 sec
63:     begin
64:       Edit3.Text:='';
65:       Edit3.Text:='00:'+Edit1.Text+'00,000';
66:       Edit1.Text:='';

```

```
68:
69:     end;
70:     if (Edit1.Text='') and (Edit2.Text<>'') then // 0 min 1 sec
71:     begin
72:         Edit3.Text:='';
73:         Edit3.Text:= '00:00:'+Edit2.Text;
74:         Edit1.Text:='';
75:         Edit2.Text:='';
76:     end;
77:     if True then
78:     if (Edit1.Text<>'') and (Edit2.Text<>'') then // 1 min 1 sec
79:     begin
80:         Edit3.Text:='';
81:         Edit3.Text:='00:'+Edit1.Text+':'+Edit2.Text;
82:         Edit1.Text:='';
83:         Edit2.Text:='';
84:     end;
85:
86:     Gara.tempo:=Edit3.Text;
87:     if RadioButton1.Checked=True then
88:     Gara.rab1:=True else
89:     begin
90:         Gara.rab2:= True;
91:         RadioButton1.Checked:=True;
92:     end;
93:
94:
95: end;
96:
97: procedure TForm9.Button3Click(Sender: TObject); // Esci
98: begin
99:     Form9.Close;
100:     Form9.Release;
101: end;
102:
103: procedure TForm9.Edit2KeyPress(Sender: TObject; var Key: Char);
104: begin
105:     if (Key=#13) AND (length(edit2.text)<6) then
106:         showmessage('Il formato deve essere YY,ZZZ ');
107:     end;
108:
109: end.
110:
111:
```

```
1: unit aiuto;
2:
3: interface
4:
5: uses
6:   Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
7:   Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls;
8:
9: type
10:   TForm7 = class(TForm)
11:     mmol: TMemo;
12:     Button1: TButton;
13:     procedure Button1Click(Sender: TObject);
14:   private
15:     { Private declarations }
16:   public
17:     { Public declarations }
18:   end;
19:
20: var
21:   Form7: TForm7;
22:
23: implementation
24:   uses
25:     Gara;
26:   {$R *.dfm}
27:   {$D+}
28: procedure TForm7.Button1Click(Sender: TObject);
29: begin
30:   Form7.visible:=False;
31: end;
32:
33: end.
```