

```
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); //azzerà 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. Ripetili').
```

```

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.FieldName('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.FieldName('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.FieldName('NMR').AsInteger:=
873:                             ABSQuery85.FieldName('NMR').AsInteger;
874:                         ABSTable15.FieldName('CODGA').AsInteger:=
875:                             ABSQuery85.FieldName('CODGA').AsInteger;
876:                         ABSTable15.FieldName('CODDSPL').AsInteger:=
877:                             ABSQuery85.FieldName('CODDSPL').AsInteger;
878:                         ABSTable15.FieldName('CODATL').AsInteger:=
879:                             ABSQuery85.FieldName('CODATL').AsInteger;
880:                         ABSTable15.FieldName('COGNOME').AsInteger:=
881:                             ABSQuery85.FieldName('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 //Ordina per tempi crescenti
1183:         Close;
1184:         SQL.Text:='Select * from Ranking Order By "TMC"';
1185:         ExecSQL;
1186:         Open
1187:     end;
1188:
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 //Ordinata per tempi crescenti
1198:         ABSTable16.Insert;
1199:         ABSTable16.FieldName('NMR').AsInteger:=
1200:             ABSQuery85.FieldName('NMR').AsInteger;
1201:         ABSTable16.FieldName('CODGA').AsInteger:=
1202:             ABSQuery85.FieldName('CODGA').AsInteger;
1203:         ABSTable16.FieldName('CODDSPL').AsInteger:=
1204:             ABSQuery85.FieldName('CODDSPL').AsInteger;
1205:         ABSTable16.FieldName('CODATL').AsInteger:=
1206:             ABSQuery85.FieldName('CODATL').AsInteger;
1207:         ABSTable16.FieldName('COGNOME').AsString:=
1208:             ABSQuery85.FieldName('COGNOME').AsString;
1209:         ABSTable16.FieldName('NMPET').AsInteger:=
1210:             ABSQuery85.FieldName('NMPET').AsInteger;
1211:         ABSTable16.FieldName('NMPOS').AsInteger:=
1212:             ABSQuery85.FieldName('NMPOS').AsInteger;
1213:         ABSTable16.FieldName('TAME1').AsDateTime:=
1214:             ABSQuery85.FieldName('TAME1').AsDateTime;
1215:         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))+#13+'?',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))+#13+'?',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?',mtconfirmation, [mbYes,mbNo], 0) = mrYes then
```



```

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:

```



```
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.
```