```
Imports System.IO
Imports System.Data.oledb
Public Class Form1
    Public Ofd As New OpenFileDialog
    Dim miofile As StreamReader
    Dim a As Double 'x minima
    Dim b As Double 'x massima
    Dim c As Double 'y minima
    Dim d As Double 'y massima
    Dim ymin, ymax As Double
    Dim gr As Graphics
    Dim FlagFunc As Boolean
    Dim integ As Integrale


    Public Function cx(ByVal x As Double) As Integer
        Dim cox As Integer
        Try
            'a = txtXMin.Text
            'b = txtXMax.Text
            cox = (((x - a)) / ((b - a))) * Panel1.Width 'calcolo l'ascissa sul pannello
            Return cox
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Function

    Public Function cy(ByVal y As Double) As Integer
        Dim coy As Integer
        coy = (1 - ((((y - c)) / ((d - c))))) * Panel1.Height 'calcolo l'ordinata sul pannello
        Return coy
    End Function

    Function f(ByVal x As Double) As Double
        Dim ya As Double
        Dim media As Double
        Dim scarto As Double
        If txtMedia.Text <> "" And txtScarto.Text <> "" And RadioButton1.Checked = True Then
            media = txtMedia.Text
            scarto = txtScarto.Text
            ya = (1 / scarto * Math.Sqrt(2 * 3.14159265)) * (2.718281828 ^ -((x - media) ^ 2) / 2 * (scarto ^ 2)) 'Funzione
        ElseIf RadioButton2.Checked = True And lblS.Text <> "" And lblM.Text <> "" Then
            media = lblM.Text
            scarto = lblS.Text
            ya = (1 / scarto * Math.Sqrt(2 * 3.14159265)) * (2.718281828 ^ -((x - media) ^ 2) / 2 * (scarto ^ 2)) 'Funzione
        End If
        Return ya
    End Function

    Sub funzione()
```

```
51          Dim x As Double
52          Dim ya As Single
53          gr = Panel1.CreateGraphics
54          Dim incr As Double
55          FlagFunc = True
56          incr = Math.Abs(Math.Abs(b - a) / (Panel1.Width * 2)) 'dimensione
              incremento dell'indice
57          For x = (a) To (b) Step incr 'da a fino a b incrementandosi di incr
58              Try
59                  ya = f(x) 'ritorna il valore della funzione su questa x
60                  'gr.DrawEllipse(Pens.Red, cx(x), cy(ya), 1, 1)
61                  gr.DrawLine(Pens.Blue, cx(x), (cy(ya)), cx(x + incr), cy(f(x +
                      incr))) 'disegna una linea da x,y correnti a x,y del ciclo
                    successivo
62              Catch ex As Exception
63
64              End Try
65          Next
66          creaReticolo()
67      End Sub
68
69      Sub rangeY()
70          Dim x, min, max As Double
71          Dim ya As Single
72          Dim incr As Double
73          Dim flag1, flag2 As Boolean
74          flag1 = False
75          flag2 = False
76          Try
77              incr = Math.Abs(Math.Abs(b - a) / (Panel1.Width * 2))
78              min = -0.0001 'massimo e minimo iniziali molto vicini allo 0
79              max = 0.0001
80              a = txtXMin.Text
81              b = txtXMax.Text
82              If (a < b) Then
83                  For x = a To b Step 1
84                      ya = f(x) 'ya assume il valore della funzione nell'ascissa
                         x
85                      If (ya < min) And (ya > (-100)) Then 'se ya e' piu'
                        piccola di min  e piu' grande del tetto minimo definito
                        come -100
86                          min = ya 'il minimo diventa ya
87                          flag1 = True
88                      End If
89                      If (ya > max) And (ya < 100) Then 'se ya e' piu' grande di
                         max  e piu' piccolo del tetto massimo definito come 100
90                          max = ya 'il massimo diventa max
91                          flag2 = True
92                      End If
93                  Next
94                  c = (min - 2) 'diminuisco il minimo ottenuto di 2 e aumento il
                       massimo di 2 per evitare di avere la funzione attaccata a
                      punti non compresi o compresi al pelo
95                  d = (max + 2)
96                  If flag1 = False Then 'se non ho mai cambiato il minimo
97                      min = ya 'min diventa l'ultima ya trovata che sara' > 0
```

```
98                      For x = a To b Step 1
99                          ya = f(x) 'ya assume il valore della variabile
                        dipendente di x
100                         If (ya < min) Then 'se ya e' minore dell'ultimo min
                        posto (l'ultima ya in partenza)
101                             min = ya 'allora min diventa la corrente ya
102                         End If
103                     Next
104                     c = min - 2 'la c (valore minimo dell'ordinata della
                        funzione) viene abbassato di 2 per rendere visibile la
                        funzione
105                 End If
106                 If flag2 = False Then 'stesso lavoro di sopra, ma al contrario
                    per ottenere l'ordinata massima nel caso l'ordinata massima
                    sia minore di 0
107                     max = ya
108                     For x = a To b Step 1
109                         ya = f(x)
110                         If (ya > max) Then
111                             max = ya
112                         End If
113                     Next
114                     d = max + 2
115                 End If
116             Else
117                 MsgBox("XMin deve essere minore di Xmax!")
118             End If
119         Catch ex As Exception
120             MsgBox(ex.Message)
121         End Try
122     End Sub
123
124     Sub assi()
125         Dim gr As Graphics
126         gr = Panel1.CreateGraphics
127         gr.Clear(Color.White)
128         Try
129             If (b > 0) And (a < 0) Then
130                 gr.DrawLine(Pens.Black, cx(0), (cy(c)), cx(0), (cy(d))) 'asse
                      y
131             End If
132             If (d > 0) And (c < 0) Then
133                 gr.DrawLine(Pens.Black, cx(a), cy(0), cx(b), cy(0)) 'asse x
134             End If
135             creaReticolo()
136         Catch ex As Exception
137
138         End Try
139     End Sub
140
141     Private Sub BtnDisegna_Click(ByVal sender As System.Object, ByVal e As
          System.EventArgs) Handles BtnDisegna.Click
142         rangeY()
143         Button1.Visible = True
144         If a < b Then
145             assi() 'disegna assi
```

```
146             funzione() 'disegna funzione
147             Label3.Text = "Y Minima= " & Math.Truncate(c) 'mostra il valore
                  dell'ord min troncato all'intero
148             Label4.Text = "Y Massima= " & Math.Truncate(d) 'mostra il valore
                  dell'ord max troncato all'intero
149         End If
150
151         If c > 0 Then 'casi in cui gli assi non sono visibili
152             BtnVisAssi.Visible = True 'rendo visibile un pulsante per
                  allargare la funzione e visualizzare gli assi
153         End If
154         If d < 0 Then
155             BtnVisAssi.Visible = True
156         End If
157         If a > 0 Then
158             BtnVisAssi.Visible = True
159         End If
160         If b < 0 Then
161             BtnVisAssi.Visible = True
162         End If
163
164     End Sub
165
166     Private Sub BtnVisAssi_Click(ByVal sender As System.Object, ByVal e As
          System.EventArgs) Handles BtnVisAssi.Click
167         If c > 0 Then 'modifico il valore degli estremi per rendere visibili
              gli assi e ridisegno la funzione
168             c = -2
169         End If
170         If d < 0 Then
171             d = 2
172         End If
173         If a > 0 Then
174             a = -2
175         End If
176         If b < 0 Then
177             b = 2
178         End If
179
180
181         assi()
182         funzione()
183         Label3.Text = "Y Minima= " & Math.Truncate(c) 'mostra il valore
              dell'ord min troncato all'intero
184         Label4.Text = "Y Massima= " & Math.Truncate(d) 'mostra il valore
              dell'ord max troncato all'intero
185
186
187         BtnVisAssi.Visible = False
188     End Sub
189
190     'Private Sub BtnIndietro_Click(ByVal sender As System.Object, ByVal e As
          System.EventArgs)
191     '    assi()
192     '    funzione()
193     '    Label3.Text = "Y Minima= " & c
```

```
194     '    Label4.Text = "Y Massima= " & d
195     'End Sub
196
197     Public Sub creaReticolo()
198         Dim i As Single
199         Dim gr As Graphics
200         gr = Panel1.CreateGraphics
201         Dim incr As Double
202         Dim incr2 As Double
203         Dim incrY As Double
204         incr = Math.Abs(b - a) / 10
205         incr2 = cx(incr) - cx(0)
206         For i = incr2 To Panel1.Width Step incr2
207             Try
208                 If (a < 0) And (b > 0) Then
209                     gr.DrawLine(Pens.Fuchsia, cx(0) + i, 0, cx(0) + i,
                        Panel1.Height)
210                     gr.DrawLine(Pens.Fuchsia, cx(0) - i, 0, cx(0) - i,
                        Panel1.Height)
211                 Else
212                     gr.DrawLine(Pens.Fuchsia, cx(a) + i, 0, cx(a) + i,
                        Panel1.Height)
213                 End If
214             Catch ex As Exception
215
216             End Try
217         Next
218         For i = incr2 To Panel1.Height Step incr2
219             Try
220                 If (c < 0) And (d > 0) Then
221                     gr.DrawLine(Pens.Fuchsia, 0, (cy(0) + i), Panel1.Width,
                        (cy(0) + i))
222                     gr.DrawLine(Pens.Fuchsia, 0, (cy(0) - i), Panel1.Width,
                        (cy(0) - i))
223                 Else
224                     gr.DrawLine(Pens.Fuchsia, 0, (cy(c) + i), Panel1.Width,
                        (cy(c) + i))
225                 End If
226             Catch ex As Exception
227
228             End Try
229         Next
230         lblQW.Text = "Larghezza Quadrato=" & Math.Round(incr, 3)
231         incrY = (Math.Abs(d - c) / (Panel1.Height / incr2))
232         lblQH.Text = "Altezza Quadrato=" & Math.Round(incrY, 3)
233     End Sub
234
235
236     Private Sub BtnRI_Click(ByVal sender As System.Object, ByVal e As
          System.EventArgs) Handles BtnRI.Click
237         Dim baseRett As Double
238         Dim i As Integer
239         Dim altRett1 As Double
240         Dim altRett2 As Double
241         Dim altRett As Double
242         Dim b1, b2 As Single
```

```
243         Dim baseP As Single
244         Dim altP As Single
245         Dim n As Integer
246         If FlagFunc = True And IsNumeric(TxtN.Text) = True And TxtN.Text <> "" Then
247             integ = New Integrale(TxtN.Text, txtMedia.Text, txtScarto.Text)
248             LblRI.Text = "Area Rett Inf=" & Math.Round(integ.rettangoliInf(a, b), 3)
249             n = TxtN.Text
250             baseRett = (b - a) / n
251             For i = 1 To n
252                 Try
253                     b1 = CSng(a + (baseRett * (i - 1)))
254                     b2 = CSng(a + (baseRett * (i)))
255                     altRett1 = f(a + (baseRett * (i - 1)))
256                     altRett2 = f(a + (baseRett * i))
257                     baseP = cx(baseRett) - cx(0)
258                     If altRett1 < altRett2 Then
259                         altRett = altRett1
260                     Else
261                         altRett = altRett2
262                     End If
263                     altP = Math.Abs(cy(altRett) - cy(0))
264                     If ((f(altRett)) < 0) Then
265                         gr.DrawRectangle(Pens.Blue, cx(b1), cy(0), baseP, altP)
266                     Else
267                         gr.DrawRectangle(Pens.Blue, cx(b1), cy(altRett), baseP, altP)
268                     End If
269                 Catch ex As Exception
270
271                 End Try
272             Next
273         End If
274     End Sub
275
276     Private Sub BtnRS_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnRS.Click
277         Dim baseRett As Double
278         Dim i As Integer
279         Dim altRett1 As Double
280         Dim altRett2 As Double
281         Dim altRett As Double
282         Dim b1, b2 As Single
283         Dim baseP As Single
284         Dim altP As Single
285         Dim n As Integer
286         If FlagFunc = True And IsNumeric(TxtN.Text) = True And TxtN.Text <> "" Then
287             integ = New Integrale(TxtN.Text, txtMedia.Text, txtScarto.Text)
288             LblRS.Text = "Area Rett Sup=" & Math.Round(integ.rettangoliSup(a, b), 3)
289             n = TxtN.Text
290             baseRett = (b - a) / n
291             For i = 1 To n
```

```
292                 Try
293                     b1 = CSng(a + (baseRett * (i - 1)))
294                     b2 = CSng(a + (baseRett * (i)))
295                     altRett1 = f(a + (baseRett * (i - 1)))
296                     altRett2 = f(a + (baseRett * i))
297                     baseP = cx(baseRett) - cx(0)
298                     If altRett1 > altRett2 Then
299                         altRett = altRett1
300                     Else
301                         altRett = altRett2
302                     End If
303                     altP = Math.Abs(cy(altRett) - cy(0))
304                     If ((f(altRett)) < 0) Then
305                         gr.DrawRectangle(Pens.Green, cx(b1), cy(0), baseP,
                        altP)
306                     Else
307                         gr.DrawRectangle(Pens.Green, cx(b1), cy(altRett),
                        baseP, altP)
308                     End If
309                 Catch ex As Exception
310
311                 End Try
312             Next
313         End If
314     End Sub
315
316     Private Sub BtnT_Click(ByVal sender As System.Object, ByVal e As
          System.EventArgs) Handles BtnT.Click
317         Dim baseRett As Double
318         Dim i As Integer
319         Dim altRett1 As Double
320         Dim altRett2 As Double
321         Dim b1, b2 As Single
322         Dim baseP As Single
323         Dim altP1 As Single
324         Dim altP2 As Single
325         Dim n As Integer
326         Dim punto1 As Point
327         Dim punto2 As Point
328         Dim punto3 As Point
329         Dim punto4 As Point
330         If FlagFunc = True And IsNumeric(TxtN.Text) = True And TxtN.Text <> ""
            Then
331             integ = New Integrale(TxtN.Text, txtMedia.Text, txtScarto.Text)
332             LblT.Text = "Area Trap=" & Math.Round(integ.trapezi(a, b), 3)
333             n = TxtN.Text
334             baseRett = (b - a) / n
335             For i = 1 To n
336                 Try
337                     b1 = CSng(a + (baseRett * (i - 1)))
338                     b2 = CSng(a + (baseRett * (i)))
339                     altRett1 = f(a + (baseRett * (i - 1)))
340                     altRett2 = f(a + (baseRett * i))
341                     baseP = cx(baseRett) - cx(0)
342                     altP1 = Math.Abs(cy(altRett1) - cy(0))
343                     altP2 = Math.Abs(cy(altRett2) - cy(0))
```

```
344                     punto1.X = cx(b1)
345                     punto1.Y = cy(0)
346                     punto2.X = cx(b2)
347                     punto2.Y = cy(0)
348                     punto3.X = cx(b2)
349                     punto3.Y = cy(altRett2)
350                     punto4.X = cx(b1)
351                     punto4.Y = cy(altRett1)
352
353                     gr.DrawLine(Pens.Lavender, punto2.X, punto2.Y, punto3.X,
                        punto3.Y)
354                     gr.DrawLine(Pens.IndianRed, punto3.X, punto3.Y, punto4.X,
                        punto4.Y)
355                     gr.DrawLine(Pens.IndianRed, punto4.X, punto4.Y, punto1.X,
                        punto1.Y)
356                 Catch ex As Exception
357
358                 End Try
359             Next
360         End If
361     End Sub
362
363     Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
          System.EventArgs) Handles Button1.Click
364         Dim media As Double
365         Dim scarto As Double
366         If txtMedia.Text <> "" And txtScarto.Text <> "" Then
367             media = txtMedia.Text
368             scarto = txtScarto.Text
369             If RadioButton1.Checked = True Then
370                 GaussianaStd.a = (a - media) / scarto
371                 GaussianaStd.b = (b - media) / scarto
372             Else
373                 GaussianaStd.a = (a - CDbl(lblM.Text)) / CDbl(lblS.Text)
374                 GaussianaStd.b = (b - CDbl(lblM.Text)) / CDbl(lblS.Text)
375             End If
376             GaussianaStd.ShowDialog()
377         Else
378             media = 0
379             scarto = 0
380             MsgBox("Inserisci il valore di media e scarto prima!!")
381         End If
382     End Sub
383
384     Private Sub Tmr_Tick(ByVal sender As System.Object, ByVal e As
          System.EventArgs)
385         If a < b Then
386             assi() 'disegna assi
387             funzione() 'disegna funzione
388         End If
389     End Sub
390
391
392
393     Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
          System.EventArgs) Handles MyBase.Load
```

```
394         DGV.AllowUserToAddRows = False
395     End Sub
396
397     Private Sub leggiMdb(ByVal s As String)
398         Dim i As Integer
399         Dim conn As New OleDb.OleDbConnection()
400         Dim nomi As New nomiTab
401         Dim med, var As Double
402         i = nomi.RitornaNomi(DGV, CMB, s)
403         If (i <= 1) Then
404             Dim cmd As OleDbDataAdapter
405             Dim DS As New DataSet
406             conn = New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;" &
                  "Data Source=" & s)
407             cmd = New OleDbDataAdapter("SELECT * FROM [" & CMB.Items(0) & "]",
                    conn)
408             cmd.Fill(DS, "tabella")
409             DGV.DataSource = DS.Tables(0)
410             conn.Close()
411         Else
412             CMB.Visible = True
413             DGV.Visible = False
414         End If
415         DGV.AllowUserToAddRows = False
416         mediaVarMdb(DGV, med, var)
417         var = Math.Sqrt(var)
418         lblM.Text = Math.Round(med, 3)
419         lblS.Text = Math.Round(var, 3)
420     End Sub
421
422     Public Sub leggiCsv(ByVal s As String)
423         Dim strV As String()
424         Dim y, i, colCount As Integer
425         Dim Flag As Boolean = False
426         Dim med, var As Double
427         colCount = 1
428         Dim a As String = ""
429         Try
430             miofile = File.OpenText(s)
431             While miofile.Peek <> -1
432                 a = miofile.ReadLine
433                 strV = Split(a, ",")
434                 For i = 0 To strV.Length - 1
435                     If y = 0 Then
436                         colCount += 1
437                         DGV.Columns.Add(strV(i), strV(i))
438                     Else
439                         If y <> 0 And colCount <= strV.Length Then
440                             MsgBox("Troppi elementi nella riga" & y)
441                             Flag = True
442                             Exit For
443                         End If
444                         If y <> 0 And i = 0 Then
445                             DGV.Rows.Add()
446                         End If
447                         If y <> 0 And colCount > strV.Length Then
```

```
448                             DGV.Item(i, (y - 1)).Value = strV(i)
449                         End If
450                     End If
451                 Next
452                 If Flag = True Then
453                     Exit While
454                 End If
455                 y += 1
456             End While
457             DGV.AllowUserToAddRows = False
458             miofile.Close()
459             mediaVar(DGV, med, var)
460             var = Math.Sqrt(var)
461             lblM.Text = Math.Round(med, 3)
462             lblS.Text = Math.Round(var, 3)
463         Catch ex As Exception
464             MsgBox(ex.Message)
465         End Try
466     End Sub
467
468     Public Sub leggiXls(ByVal s As String)
469         Dim DS As New DataSet
470         Dim i As Integer
471         Dim nomi As New nomiTab
472         Dim med, var As Double
473         Dim cmd As OleDb.OleDbDataAdapter
474         i = nomi.RitornaNomi(DGV, CMB, s)
475         If (i <= 1) Then
476             Dim conn As New OleDbConnection                                 ↵
                  ("Provider=Microsoft.Jet.OLEDB.4.0;" & "Data Source=" & s & ";"  ↵
                  & "Extended Properties='Excel 8.0;HDR=YES;'")
477             cmd = New OleDbDataAdapter("SELECT * FROM [" & CMB.Items(0) & "]", ↵
                   conn)
478             cmd.Fill(DS, "tabella")
479             DGV.DataSource = DS.Tables(0)
480             conn.Close()
481         Else
482             CMB.Visible = True
483             DGV.Visible = False
484         End If
485         DGV.AllowUserToAddRows = False
486         mediaVar(DGV, med, var)
487         var = Math.Sqrt(var)
488         lblM.Text = Math.Round(med, 3)
489         lblS.Text = Math.Round(var, 3)
490     End Sub
491
492     Private Sub CMB_SelectedIndexChanged(ByVal sender As System.Object, ByVal  ↵
          e As System.EventArgs) Handles CMB.SelectedIndexChanged
493         Dim cmd As OleDbDataAdapter
494         Dim DS As New DataSet
495         If Strings.Right(Ofd.FileName, 3) = "mdb" Then
496             Dim conn As New OleDbConnection                                 ↵
                  ("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & Ofd.FileName)
497             cmd = New OleDbDataAdapter("SELECT * FROM [" & CMB.Text & "]",  ↵
                  conn)
```

```
498             cmd.Fill(DS, "tabella").ToString()
499             DGV.DataSource = DS.Tables(0)
500             conn.Close()
501         End If
502         If Strings.Right(Ofd.FileName, 3) = "xls" Then
503             Dim conn As New OleDbConnection
                  ("Provider=Microsoft.Jet.OLEDB.4.0;" & "Data Source=" &
                  Ofd.FileName & ";" & "Extended Properties='Excel 8.0;HDR=YES;'")
504             cmd = New OleDbDataAdapter("SELECT * FROM [" & CMB.Text & "]",
                  conn)
505             cmd.Fill(DS, "tabella").ToString()
506             DGV.DataSource = DS.Tables(0)
507             conn.Close()
508         End If
509         CMB.Visible = False
510         DGV.Visible = False
511         BtnDisegna.Visible = True
512     End Sub
513
514     Private Sub importa()
515         DGV.DataSource = Nothing
516         DGV.Columns.Clear()
517         Ofd.InitialDirectory = "C:\Users\Seyriu\Desktop\ApriFileInDGV
              \ApriFileInDGV\bin\Debug"
518         Ofd.Title = "Apri"
519         Ofd.Filter = "Access Files|*.mdb|Excel Files|*.xls|Text Files|*.csv"
520         Ofd.ShowDialog()
521         If Strings.Right(Ofd.FileName, 3) = "mdb" Then
522             Try
523                 leggiMdb(Ofd.FileName)
524             Catch ex As Exception
525                 MsgBox(ex.Message)
526             End Try
527         End If
528         If Strings.Right(Ofd.FileName, 3) = "csv" Then
529             Try
530                 leggiCsv(Ofd.FileName)
531             Catch ex As Exception
532                 MsgBox(ex.Message)
533             End Try
534         End If
535         If Strings.Right(Ofd.FileName, 3) = "xls" Then
536             Try
537                 leggiXls(Ofd.FileName)
538             Catch ex As Exception
539                 MsgBox(ex.Message)
540             End Try
541         End If
542         If Strings.Right(Ofd.FileName, 3) = "csv" Then
543             BtnDisegna.Visible = True
544         End If
545     End Sub
546
547     Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object,
          ByVal e As System.EventArgs) Handles RadioButton1.CheckedChanged
548         BtnDisegna.Visible = True
```

```
549         'Label6.Visible = True
550         'Label7.Visible = True
551         txtMedia.Visible = True
552         txtScarto.Visible = True
553         lblM.Visible = False
554         lblS.Visible = False
555         BtnApri.Visible = False
556         CMB.Visible = False
557         Comb.Visible = False
558     End Sub
559
560     Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object,
          ByVal e As System.EventArgs) Handles RadioButton2.CheckedChanged
561         BtnDisegna.Visible = False
562         'Label6.Visible = False
563         'Label7.Visible = False
564         txtMedia.Visible = False
565         txtScarto.Visible = False
566         lblM.Visible = True
567         lblS.Visible = True
568         BtnApri.Visible = True
569         'CMB.Visible = True
570     End Sub
571
572     Private Sub BtnApri_Click(ByVal sender As System.Object, ByVal e As
          System.EventArgs) Handles BtnApri.Click
573         importa()
574         If Not Strings.Right(Ofd.FileName, 3) = "csv" Then
575             CMB.Visible = True
576             BtnDisegna.Visible = True
577             Comb.Visible = True
578         End If
579     End Sub
580
581     Private Sub mediaVarMdb(ByVal dgv As DataGridView, ByRef media As Double,
          ByRef varianza As Double)
582         Dim i As Integer
583         Dim somma, mediaS As Double
584
585         For i = 0 To dgv.RowCount - 1
586             somma += dgv.Item(2, i).Value
587         Next
588         media = somma / dgv.RowCount
589
590         somma = 0
591         For i = 0 To dgv.RowCount - 1
592             somma += (dgv.Item(2, i).Value) ^ 2
593         Next
594         mediaS = somma / dgv.RowCount
595         varianza = mediaS - (media ^ 2)
596     End Sub
597
598     Private Sub mediaVar(ByVal dgv As DataGridView, ByRef media As Double,
          ByRef varianza As Double)
599         Dim i As Integer
600         Dim somma, mediaS As Double
```

```
601         For i = 0 To dgv.RowCount - 1
602             somma += dgv.Item(1, i).Value
603         Next
604         media = somma / dgv.RowCount
605 
606         somma = 0
607         For i = 0 To dgv.RowCount - 1
608             somma += (dgv.Item(1, i).Value) ^ 2
609         Next
610         mediaS = somma / dgv.RowCount
611         varianza = mediaS - (media ^ 2)
612     End Sub
613 
614     Private Sub Comb_Click(ByVal sender As System.Object, ByVal e As
          System.EventArgs) Handles Comb.Click
615         Combinazioni.Show()
616     End Sub
617 End Class
618 
```