

```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace Calcolatricewindows
12 {
13     /// <summary>
14     /// Michele Sprocatti
15     ///Funzioni:
16     ///Calcolatrice funzionante
17     ///reciproco, negazione dell'operando, percentuale
18     ///Tasto C, tasto CE, tasto backspace funzionanti
19     ///input sia da tastiera che grafico( Numeri, 4 simboli operazioni(prodotta
20     ///con asterisco) e backspace)
21     ///premuti da tastiera invio e uguale effettuano il calcolo
22     /// </summary>
23     public partial class Form1 : Form
24     {
25         string operando=""; //variabile inizializzata per i controlli
26         double operando1;
27         double operando2;
28         double risultato;
29         public Form1()
30         {
31             InitializeComponent();
32         }
33
34         private void Elaborazione() //metodo che esegue i calcoli delle quattro
35         operazioni
36         {
37             switch (operando)
38             {
39                 //calcolo risultato in base a operando
40                 case "+":
41                     risultato = operando1 + operando2;
42                     break;
43                 case "*":
44                     risultato = operando1 * operando2;
45                     break;
46                 case "/":
47                     if (operando2 != 0) //controllo secondo operando se uguale
48                     a 0
49                         risultato = operando1 / operando2;
50                     else
51                         if (operando1 != 0)
52                             MessageBox.Show("Divisione per zero", "Errore",
53                             MessageBoxButtons.OK, MessageBoxIcon.Error);
54                     else
55                         MessageBox.Show("Risultato indefinito, 0/0",

```

-Manca una segnalazione d'errore sul calcolo della radice con valori negativi.

-Alcune funzionalità della calcolatrice win non funzionano correttamente es: 3+=.

Gestione del segno es:-3+6=9 sqrt non si può fare perchè lo considera un numero negativo.

Complessivamente corretto il funzionamento della calcolatrice.

Voto:8- Melon Federico.

```
        "Attenzione", MessageBoxButtons.OK,
        MessageBoxIcon.Warning);
51         break;
52     case "-":
53         risultato = operando1 - operando2;
54         break;
55     }
56
57 }
58 private void btncclear_Click(object sender, EventArgs e)//tasto C
59 { //cancellazione completa
60     operando1 = 0;
61     operando2 = 0;
62     risultato = 0;
63     operando = "";
64     txtinput.Text="0";
65     txtoutput.Clear();
66 }
67
68 private void btn8_Click(object sender, EventArgs e)//bottoni numeri
69 {
70     if (txtinput.Text == "0")
71         txtinput.Clear();//cancellazione dello zero iniziale dalla
72         textbox
73     txtinput.Text += ((Button)sender).Text;
74
75 private void btnprodotto_Click(object sender, EventArgs e)//bottoni
76     operandi
77 {
78     if (operando != "-")//controllo per negare secondo operando
79         if (((Button)sender).Text == "-")
80             txtinput.Text += ((Button)sender).Text;
81     else
82         if (txtinput.Text != "")//controllo mancanza primo operando
83         {
84             txtoutput.Clear();
85             bool operandoinserito = txtinput.Text[txtinput.Text.Length -
86             1] == '+' || txtinput.Text[txtinput.Text.Length - 1] == '*'
87             || txtinput.Text[txtinput.Text.Length - 1] == '-' ||
88             txtinput.Text[txtinput.Text.Length - 1] == '/';
89             if (!operandoinserito)//controllo operando già inserito
90             {
91                 if (operando != "")
92                     btncalcola.PerformClick();
93                 operando1 = Convert.ToDouble(txtinput.Text);
94                 operando = ((Button)sender).Text;
95                 if (operando != "/")
96                     txtinput.Text += operando;
97                 else
98                     txtinput.Text += "÷";//scrittura del carattere ÷ a
99                     posto del carattere /
100                 txtoutput.Text += txtinput.Text;
101                 txtinput.Clear();
102             }
103         }
104     else
```

```
99         MessageBox.Show("Operando già inserito", "Già inserito",
100             MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
101     }
102     else
103         MessageBox.Show("Manca primo operando", "Errore",
104             MessageBoxButtons.OK, MessageBoxIcon.Error);
105 }
106 private void btncalcola_Click(object sender, EventArgs e)//bottone
107     uguale
108 {
109     if(operando!="")//controllo operando inserito
110     {
111         txtoutput.Text += txtinput.Text;
112         if(txtinput.Text!="")//controllo 2 operando inserito
113         {
114             operando2 = Convert.ToDouble(txtinput.Text);
115             txtinput.Clear();
116             Elabora();
117             if (operando != "/" || operando2 != 0)
118                 txtinput.Text += risultato;
119             operando = "";
120             txtoutput.Clear();
121         }
122         else
123             MessageBox.Show("Secondo operando mancante", "Mancanza",
124                 MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
125     }
126     else
127         if(operando1!=0)//controllo mancanza anche primo operando
128             MessageBox.Show("Operando mancante", "Mancanza",
129                 MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
130         else
131             MessageBox.Show("primo operando, operando e secondo
132                 operando mancanti", "Mancanza", MessageBoxButtons.OK,
133                 MessageBoxIcon.Exclamation);
134 }
135 private void btnquadrato_Click(object sender, EventArgs e)//bottone
136     per elevare al quadrato
137 {
138     double num;
139     if (txtinput.Text != "")//controllo se numero presente o no
140     {
141         num = Convert.ToDouble(txtinput.Text);
142         txtoutput.Text += "(" + num + ")^2";
143         num *= num;
144         txtinput.Clear();
145         txtinput.Text += num;
146     }
147     else
148         MessageBox.Show("Elemento di cui fare il quadrato mancante",
149             "Mancanza", MessageBoxButtons.OK,
150             MessageBoxIcon.Exclamation);
151 }
```

```
145
146     private void btnpercentuale_Click(object sender, EventArgs e)//bottone ➤
           percentuale
147     {
148         double num;
149         if (txtinput.Text != "")//controllo se numero presente o no
150         {
151             num = Convert.ToDouble(txtinput.Text);
152             num = (num/100)*operando1;
153             txtinput.Clear();
154             txtinput.Text += num;
155         }
156         else
157             MessageBox.Show("Elemento di cui calcolare la percentuale ➤
           mancante", "Mancanza", MessageBoxButtons.OK, ➤
           MessageBoxIcon.Exclamation);
158     }
159
160     private void btnradice_Click(object sender, EventArgs e)//bottone ➤
           radice
161     {
162         double num;
163         if (txtinput.Text != "")//controllo se numero presente o no
164         {
165             num = Convert.ToDouble(txtinput.Text);
166             if (operando1 >= 0)//controllo numero maggiore di zero
167             {
168                 txtoutput.Text += ((Button)sender).Text + num;
169                 num = Math.Sqrt(num);
170                 txtinput.Clear();
171                 txtinput.Text += num;
172             }
173             else
174                 MessageBox.Show("Radice di un numero negativo", "Errore", ➤
           MessageBoxButtons.OK, MessageBoxIcon.Error);
175         }
176         else
177             MessageBox.Show("Elemento di cui calcolare la radice ➤
           mancante", "Mancanza", MessageBoxButtons.OK, ➤
           MessageBoxIcon.Exclamation);
178     }
179
180
181     private void btnreciproco_Click(object sender, EventArgs e)//bottone ➤
           reciproco
182     {
183         double num;
184         if (txtinput.Text != "")//controllo se numero presente o no
185         {
186             num = Convert.ToDouble(txtinput.Text);
187             num = 1 / num;
188             txtoutput.Text += 1 + "÷" + num;
189             txtinput.Clear();
190             txtinput.Text += num;
191         }
192         else
```

```

...olatrice\Michele Sprocatti\Calcolatricewindows\Form1.cs 5
193     MessageBox.Show("Elemento di cui calcolare fare il reciproco  ↗
        mancante", "Mancanza", MessageBoxButtons.OK,  ↗
        MessageBoxIcon.Exclamation);
194 }
195
196 private void btnbackspace_Click(object sender, EventArgs e)//bottone  ↗
        backspace
197 {
198     int dim = 0;
199     string txt;
200     if (txtinput.Text != "")//controllo se bisogna cancellare  ↗
        l'operando oppure un numero
201     {
202         dim = txtinput.Text.Length - 1;
203         txt = txtinput.Text;
204         txtinput.Clear();
205     }
206     else
207     {
208         dim = txtoutput.Text.Length - 1;
209         txt = txtoutput.Text;
210         txtoutput.Clear();
211     }
212     char[] input = new char[dim];
213     for (int i = 0; i < txt.Length - 1; i++)
214     {
215         input[i] = txt[i];
216     }
217     //controllo ultimo carattere se è l'operando
218     if (txt[txt.Length - 1] == '+' || txt[txt.Length - 1] == '*' ||  ↗
        txt[txt.Length - 1] == '-' || txt[txt.Length - 1] == '÷')
219         operando = "";
220     foreach (var item in input)
221     {
222         txtinput.Text += item;
223     }
224 }
225
226 private void btnvirgola_Click(object sender, EventArgs e)//bottone  ↗
        virgola
227 {
228     string s = txtinput.Text;
229     int pos=s.IndexOf(",");//ricerca dell'elemento
230     if(pos==-1)//controllo se presente o no
231         txtinput.Text += ",";
232     else
233         if(s=="")//distinzione errore
234             MessageBox.Show("Impossibile effettuare inserimento",  ↗
                "Errore", MessageBoxButtons.OK, MessageBoxIcon.Error);
235         else
236             MessageBox.Show("Virgola già presente", "Errore",  ↗
                MessageBoxButtons.OK, MessageBoxIcon.Error);
237
238 }
239
240 private void txtinput_KeyPress(object sender, KeyPressEventArgs e)//  ↗

```

```
key press
{
241     string btn="";
242     char car = e.KeyChar;
243     e.Handled = true;//non scrive nulla da tastiera
244     if (e.KeyChar == '+' || e.KeyChar == '*' || e.KeyChar == '-' ||
245         e.KeyChar == '/')
246     {
247         switch (e.KeyChar)//distinzione operando premuto da tastiera
248         {
249             case '+':
250                 btnsomma.PerformClick();//sollevo evento click in base
251                 all'operando premuto
252                 break;
253             case '-':
254                 btndifferenza.PerformClick();
255                 break;
256             case '/':
257                 btndivisione.PerformClick();
258                 break;
259             case '*':
260                 btnprodotto.PerformClick();
261                 break;
262         }
263     }
264     if (char.IsDigit(e.KeyChar))//controllo se viene premuto un numero
265     {
266         btn = "btn" + e.KeyChar;
267         //controls è una collezione che contiene tutti gli elementi
268         presenti nel form
269         (Controls.Find(btn, true).FirstOrDefault() as
270         Button).PerformClick();//sollevo l'evento click del bottone
271         che ha come nome la stringa btn
272     }
273     if (e.KeyChar == '=' || e.KeyChar == 13)
274         btncalcola.PerformClick();//sollevo evento click se viene
275         premuto uguale o invio da tastiera
276     if (e.KeyChar == '%')
277         btnpercentuale.PerformClick();//sollevo evento click se viene
278         premuto il tasto percentuale da tastiera
279     if (e.KeyChar == ',')
280         btnvirgola.PerformClick();//sollevo evento click se viene
281         premuto il tasto virgola
282     if (e.KeyChar == 8)
283         btnbackspace.PerformClick();//sollevo evento click se viene
284         premuto il tasto backspace
285 }
286
private void btnnegate_Click(object sender, EventArgs e)
{
287     //nego operando
288     double num;
289     if (txtinput.Text != "")
290         num = Convert.ToDouble(txtinput.Text);
291     else
292         num = 0;
293     num = num - (num * 2);
}
```

```
287         txtinput.Clear();
288         txtinput.Text += num;
289     }
290
291     private void Form1_Load(object sender, EventArgs e)
292     { //scrivo nella textbox 0 e la seleziono
293         txtinput.Text += "0";
294         txtinput.Select();
295     }
296
297     private void btncanc_Click(object sender, EventArgs e)
298     { //tasto CE
299         int dim = 0;
300         string txt;
301         if (txtinput.Text != "") //controllo se è stato inserito solo il primo/secondo operando
302         {
303             txtinput.Clear();
304             txtinput.Text += 0;
305         }
306         else
307         {
308             dim = txtoutput.Text.Length - 1;
309             txt = txtoutput.Text;
310             txtoutput.Clear();
311             char[] input = new char[dim];
312             for (int i = 0; i < txt.Length - 1; i++)
313             {
314                 input[i] = txt[i];
315             }
316             if (txt[txt.Length - 1] == '+' || txt[txt.Length - 1] == '*' || txt[txt.Length - 1] == '-' || txt[txt.Length - 1] == '/')
317                 operando = "";
318             foreach (var item in input)
319             {
320                 txtinput.Text += item;
321             }
322         }
323     }
324 }
325 }
326 }
327 }
```