

```
1  /*=====
2  | Konsolenanwendung Version: 01
3  | Programmname: Widerstandsberechnung
4  | Windows 10
5  | Autor: Raphael Patzelt
6  | Beschreibung:
7  |     Eingabe Anzahl von Widerstände
8  |     Einagbe Seriell/Paralell
9  |     Ausgabe des Zusammengesetzten Widerstandes
10 =====*/
11 using System;
12 using System.Collections.Generic;
13 using System.Linq;
14 using System.Text;
15 using System.Threading.Tasks;
16 using static System.Console;
17
18 namespace Widerstandsberechnung
19 {
20     class Program
21     {
22         static void Main()
23         {
24             int iResistorCount;
25             bool bFlag;
26             string sAbbruch = "N";
27
28             while (sAbbruch == "N")
29             {
30                 bFlag = true;
31                 //Einlesen der Anzahl der Widerstände
32                 WriteLine("Geben Sie die Anzahl der Widerstände ein:");
33                 while (!false)
34                 {
35                     if (int.TryParse(ReadLine(), out iResistorCount))
36                     {
37                         break;
38                     }
39                     else
40                     {
41                         WriteLine("Eingabe ungültig, bitte versuchen Sie es  
erneut:");
42                     }
43                 }
44
45                 //Erstellen des Arrays mit der zuvor eingegebenen  
Zeilenanzahl
46                 double[] daResistorValues = new double[iResistorCount];
47
48                 //Einlesen der Widerstandswerte für jeden Widerstand
49                 for (int i = 0; i < daResistorValues.Length; i++)
50                 {
51                     WriteLine($"Geben Sie den Wert für den Widerstand {i + 1
```

```
1} ein:");
52 while (!false)
53 {
54     if (double.TryParse(ReadLine(), out daResistorValues
55 [i]))
56     {
57         break;
58     }
59     else
60     {
61         WriteLine("Eingabe ungültig, bitte versuchen Sie
62 erneut");
63     }
64 }
65 //Abfrage ob parallel oder in Serie berechnet werden soll,
66 und Ausgabe dieser Berechnung
67 WriteLine("Geben Sie ein ob Sie die Widerstände in Serie
68 oder Parallel berechnen wollen (s/p):");
69 while (bFlag)
70 {
71     switch (ReadLine().ToLower())
72     {
73     case "s":
74     case "serie":
75         WriteLine($"Alle Widerstände in Serie geschaltet
76 ergeben einen Gesamtwiderstand von {Berechnung.SerienWid
77 (daResistorValues)} Ohm");
78         bFlag = false;
79         break;
80
81     case "p":
82     case "parallel":
83         WriteLine($"Alle Widerstände in Serie geschaltet
84 ergeben einen Gesamtwiderstand von
85 {Berechnung.ParalleleWid(daResistorValues)} Ohm");
86         bFlag = false;
87         break;
88
89     default:
90         WriteLine("Eingabe ungültig, bitte versuche Sie
91 es erneut");
92         break;
93     }
94 }
95 while (!false)
96 {
97     WriteLine("Wollen Sie das Programm beenden? (j/n)");
98     sAbbruch = ReadLine().ToUpper();
99
100     if (sAbbruch == "N" | sAbbruch == "J")
```

```
95         {
96             break;
97         }
98         WriteLine("Ungültige Eingabe! Nur \"N\" oder \"J\" erlaubt!");
99     }
100 }
101 }
102 }
103
104
105 //=====|Statische Klasse|=====
106 //static - Es muss kein Objekt erstellt werden, um auf die Klasse zuzugreifen
107 static class Berechnung
108 {
109     //Berechnen der Widerstände wenn in Parallel geschalten
110     public static double ParalleleWid(double[] daResistorValues)
111     {
112         double iSum = 0;
113         for (int i = 0; i < daResistorValues.Length; i++)
114         {
115             if (daResistorValues[i] == 0)
116             {
117                 return 0;
118             }
119             iSum += 1 / daResistorValues[i];
120         }
121         return 1 / iSum;
122     }
123
124
125     //Berechnen der Widerstände wenn in Serie geschalten
126     public static double SerienWid(double[] daResistorValues)
127     {
128         double iSum = 0;
129         foreach (var item in daResistorValues)
130         {
131             iSum += item;
132         }
133
134         return iSum;
135     }
136 }
137 }
138
```