

```
1 using System;
2 using System.Text;
3
4 /*
5  * ## Dezimal zu Binärconverter ##
6  *
7  * Sebastian Welscher
8  *
9  * 16.04.2018
10 *
11 */
12
13 namespace BinaryConvert
14 {
15     public class Dec2Bin
16     {
17         private static int[] bin = new int[8];
18         StringBuilder sb;
19
20         public Dec2Bin()
21         {
22             sb = new StringBuilder();
23         }
24
25         public string Convert2Binary(string dec_number)
26         {
27             int calc, mod;
28             int i = 0;
29             if (dec_number.Equals(""))
30             {
31                 return "Bitte Nummer eingeben";
32             }
33             else
34             {
35                 calc = Convert.ToInt32(dec_number);
36             }
37
38             string ausgabe;
39             Array.Clear(bin, 0, bin.Length);
40             sb.Clear();
41
42             if (calc < 0 || calc > 255)
43             {
44                 return "Zahl zu groß";
45             }
46             else
47             {
48                 do
49                 {
50                     mod = calc % 2; // prüfe auf Rest
51                     calc = calc / 2; // teile durch 2
52                     bin[i] = mod; // schreibe Ergebnis ins Array
53
54                     i++; // Zähler erhöhen
55
56                 } while (calc != 0);
```

```
57
58         for (int j = bin.Length - 1; j >= 0; j--)
59         {
60             sb.Append(Convert.ToString(bin[j])); // Stringeinzelteile ↗
61             // an StringBuilder übergeben
62         }
63         i = 0;
64         ausgabe = sb.ToString(); // String aufbauen
65
66         return ausgabe;
67     }
68 }
69
70 /*
71 * ## Binär zu Dezimalconverter ##
72 *
73 *     Sebastian Welscher
74 *
75 *     12.05.2018
76 *
77 */
78
79 public class Bin2Dec
80 {
81     private char[] chardata = new char[8];
82     private int[] intdata = new int[8];
83
84     public Bin2Dec()
85     {
86         //Konstruktor
87     }
88
89     public string Convert2Dec(string input)
90     {
91         string output;
92         ClearArray(chardata); //Arrayinhalt löschen
93         ClearArray(intdata);
94         FillCharArray(input); //CharArray füllen
95         IntoIntArray(); //CharArrayinhalte in IntArrayinhalte ↗
96         // konvertieren
97         output = ParseArray(); //IntArrayinhalt auf 0 und 1 ↗
98         // abfragen
99         return output;
100     }
101
102     private void ClearArray(Array arr)
103     {
104         Array.Clear(arr, 0, arr.Length);
105     }
106
107     private void FillCharArray(string data)
108     {
109         chardata = data.ToCharArray(); // String in einzelne Chars ↗
110         // aufspalten und in Array speichern
111     }
112 }
```

```
109
110     private void IntoIntArray()
111     {
112         int dat;
113         for(int i = 0; i < chardata.Length; i++)
114         {
115             dat = (int)char.GetNumericValue(chardata[i]); // Numerischen ↗
116             // Inhalt aus Char abrufen und in int umwandeln
117             intdata[i] = dat; // Daten in ↗
118             // IntArray speichern
119         }
120     }
121
122     private string ParseArray()
123     {
124         // To-Do: Größere Zahlen abfangen
125
126         string dat;
127         int[] convnumbers = { 128, 64, 32, 16, 8, 4, 2, 1 }; // ↗
128         // Konvertierungsnummern
129         int sum = 0, calc = 0; // Hilfsvariablen
130         for(int i = 0; i < intdata.Length; i++)
131         {
132             if (intdata[i] == 1) // wenn IntArrayeintrag == 1, ↗
133                 // dann calc die jeweilige Konvertierungsnummer zuweisen
134             {
135                 calc = convnumbers[i];
136             }
137             else
138             {
139                 calc = 0; // ansonsten 0 setzten
140             }
141             sum = sum + calc; // Konvertierungsmummern zusammenrechnen
142         }
143
144         dat = sum.ToString(); // int zu string konvertieren
145         return dat;
146     }
147 }
148
149 /*
150  * ## Dezimal zu Hexadezimalconverter ##
151  *
152  * Sebastian Welscher
153  *
154  * 13.05.2018
155  *
156  */
157
158 public class Dez2Hexa
159 {
160     StringBuilder sb = new StringBuilder();
161     private int[] hexval;
162     public Dez2Hexa()
163     {
164         //Konstruktor
165     }
166 }
```

```
161         hexval = new int[2];
162     }
163
164     public string Convert2Hex(string input)
165     {
166         string output;
167
168         output = Calculate(input); // Funktion Calculate aufrufen
169
170         return output;
171     }
172
173     private void ClearArray(Array arr)
174     {
175         Array.Clear(arr, 0, arr.Length);
176     }
177
178     private string Calculate(string data)
179     {
180         string dat;
181         ClearArray(hexval);
182         int calc = 0, mod = 0; // Hilfsvariablen
183         // sb.Clear(); // StringBuilder leeren
184         if (data.Equals(""))
185         {
186             dat = "Bitte Nummer eingeben";
187         }
188         else
189         {
190             calc = Convert.ToInt32(data);
191         }
192         if (calc < 0 || calc > 255) // Testen ob Zahl kleiner 0 oder größer 255
193         {
194             dat = "Zahl zu groß";
195         }
196         else
197         {
198             mod = calc % 16; // Modulo Operation, prüfen auf Rest
199             calc = calc / 16;
200             hexval[0] = calc;
201             hexval[1] = mod;
202             dat = Convert.ToString(hexval[0] + hexval[1]);
203             //sb.Append(Convert.ToString(ParseNumber(calc))); // Teilberechnung zu StringBuilder hinzufügen
204             //sb.Append(Convert.ToString(ParseNumber(mod))); // Teilberechnung zu StringBuilder hinzufügen
205             //dat = sb.ToString(); // String aufbauen
206         }
207         return dat;
208     }
209     private string ParseNumber(int number) // Nummer überprüfen ob größer 9 und dann jeweils den passenden Buchstaben zuweisen
210     {
211         string output = "";
212         if (number < 10)
```

```
213     {
214         output = Convert.ToString(number);
215     }
216     else
217     {
218         switch (number)
219         {
220             case 10:
221             {
222                 output = "A";
223             }
224             break;
225             case 11:
226             {
227                 output = "B";
228             }
229             break;
230             case 12:
231             {
232                 output = "C";
233             }
234             break;
235             case 13:
236             {
237                 output = "D";
238             }
239             break;
240             case 14:
241             {
242                 output = "E";
243             }
244             break;
245             case 15:
246             {
247                 output = "F";
248             }
249             break;
250         }
251     }
252     return output;
253 }
254 }
255
256 }
257
```