

Algorithmenvergleich: VB - VC# - Java

Beispiel 1: Ablaufprotokoll "Trace"

VB 6.0

```
Private Sub cmdBerechnen_Click()
z = 0
While x <> 1
If (x Mod 2 = 1) Then x = x * 3 + 1 Else x = x / 2
-----
z = z + 1
Wend
End Sub
```

'setze den Zähler z auf 0
'solange x ungleich 1:
'wenn x ungerade: verdreifache x und erhöhe um 1,
'sonst halbiere x
'erhöhe z um 1 (Anzahl der Durchläufe)
'Schleifende

C# 2010 [Konsolenanwendung]

```
static void Main(string[] args)
{
    int z = 0;
    {
        Console.WriteLine("Bitte eine natürliche Zahl");
        string natZahl = Console.ReadLine();
        int x = int.Parse(natZahl);
        {
            while (x != 1)
            {
                if (x % 2 == 1)
                {
                    x = x * 3 + 1;
                }
                else
                {
                    x = x / 2;
                }
                z += 1;
            }
            Console.WriteLine("Die Anzahl der Durchläufe beträgt: " + z);
        }
    }
}
```

Java

```
{
int x, z;
x = IO.readInt("Bitte eine Zahl: ");
z = 0;
while (x != 1)
{
    if (x % 2 == 0)
        x = x / 2;
    else
        x = 3 * x + 1;
    z = z + 1;
}
IO.println("Anzahl der Abläufe: " + z);
}
```

// definiere 2 Variablen
// lies einen Wert ein
// setze z auf 0
// solange x ungleich 1:
// falls x gerade ist:
// halbiere x
// andernfalls:
// verdreifache x, addiere 1
// erhöhe außerdem z um 1
// gib z aus

Beispiel 2: Taschenrechner

VB 6.0 [Aufruf mit "Private Function"]

```
Private Sub cmdAddition_Click()                                     'Addition
Addition = Val(txtZahl1.Text) + Val(txtZahl2.Text)
MsgBox ("Der Wert beträgt " & (Addition) & "!")
End Sub
-----
Private Sub cmdSubtraktion_Click()                                 'Subtraktion
Subtraktion = Val(txtZahl1.Text) - Val(txtZahl2.Text)
MsgBox ("Der Wert beträgt " & (Subtraktion) & "!")
End Sub
-----
Private Sub cmdMultiplikation_Click()                             'Multiplikation
Multiplikation = Val(txtZahl1.Text) * Val(txtZahl2.Text)
MsgBox ("Der Wert beträgt " & (Multiplikation) & "!")
End Sub
-----
Private Sub cmdDivision_Click()                                   'Division durch "0"!
If Val(txtZahl2.Text) = "0" Then MsgBox ("Division durch 0 läuft nicht!") Else Division
End Sub
-----
Private Function Division()                                       'Division durch "<>0"
If Val(txtZahl2.Text) <> "0" Then Division = Val(txtZahl1.Text) / Val(txtZahl2.Text)
MsgBox ("Der Wert beträgt " & (Division) & "!")
End Function
```

C# 2010 [Konsolenanwendung: Programm und Module ("Function")]

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Taschenrechner.Module;

namespace Taschenrechner
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hallo Nutzer, bitte gib eine Zahl an");
            String Zahl1 = Console.ReadLine();

            Console.WriteLine("Bitte eine zweite Zahl");
            String Zahl2 = Console.ReadLine();

            Console.WriteLine("Bitte wähle die gewünschte Rechenoperation: * für
Multiplikation, / für Division, + für Addition und - für ");
            string Zeichen = Console.ReadLine();

            Rechner meinRechner;           //Deklaration "Speicherbereich schaffen"
            meinRechner = new Rechner();   //Instanzierung "Kopieren" der Klasse
                                           //und Einfüttern in den Speicherbereich

            if (Zeichen == "+")
            {
                Console.WriteLine("Die Summe aus " + Zahl1 + " und " + Zahl2 + "
lautet: " + meinRechner.Addiere(int.Parse(Zahl1), int.Parse(Zahl2)));
            }
        }
    }
}
```

```

        else if (Zeichen == "-")
        {
            Console.WriteLine("Die Differenz aus " + Zahl1 + " und " + Zahl2 + "
lautet: " + meinRechner.Subtrahiere(int.Parse(Zahl1), int.Parse(Zahl2)));
        }

        else if (Zeichen == "*")
        {
            Console.WriteLine("Die Multiplikation von " + Zahl1 + " mit " + Zahl2
+ " lautet: " + meinRechner.Multipliziere(int.Parse(Zahl1), int.Parse(Zahl2)));
        }

        else if (Zeichen == "/")
        {
            Console.WriteLine("Der Quotient aus " + Zahl1 + " und " + Zahl2 + "
lautet: " + meinRechner.Dividiere(int.Parse(Zahl1), int.Parse(Zahl2)));
        }
    }
}

//-----

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace Taschenrechner.Module
{
    class Rechner //Klasse, die alle nötigen Rechenoperationen mitführt
    {
        public Rechner //Addiert die beiden übergebenen Rechner
        ( )
        { }

        public int Addiere(int eineZahl, int andereZahl)
        {
            return eineZahl + andereZahl;
        }

        public int Subtrahiere(int eineZahl, int andereZahl)
        {
            return eineZahl - andereZahl;
        }

        public int Multipliziere(int eineZahl, int andereZahl)
        {
            return eineZahl * andereZahl;
        }

        public float Dividiere(float eineZahl, float andereZahl)
        {
            return eineZahl / andereZahl;
        }
    }
}

```

Beispiel 3: "ggT"

VB 6.0

```
Private Function GGT(N1 As Double, N2 As Double) As Double      'Deklaration aller Parameter
  If N2 = 0 Then                                              'Bedingung
    GGT = N1
  Else                                                        'Sonst
    GGT = GGT(N2, N1 Mod N2)
  End If                                                    'Bedingungsende
End Function

-----

Private Sub cmdGGT_Click()
  Dim GCD As Double                                          'Deklaration
  txtErgebnis = " "
  GCD = GGT(Val(txtNum1), Val(txtNum2))                    'Kopie der Funktion
  txtErgebnis = Str(GCD)
End Sub
```

C# 2010 mit Ausschluss G < 1

```
using System;
using System.Collections.Generic;
using System.Text;
namespace ggT
{
    class Program
    {

//-----

        static void Main(string[] args)
        {
            double iZahl1 = 0, iZahl2 = 0, iggT = 0;          //Definitionen
            ueberschrift();                                  //Methoden und
            iZahl1 = EingabeZahlGroesserAlsNull(1);          //Funktionen
            iZahl2 = EingabeZahlGroesserAlsNull(2);
            iggT = ggTBerechnen(iZahl1, iZahl2);
            AusgabeErgebnis(iZahl1, iZahl2, iggT);
        }

//-----

        static void ueberschrift()
        {
            Console.WriteLine("Berechnung des ggT");          //Methode Titel
            Console.WriteLine("");                             //Leerzeile
        }

//-----

//Methode Ergebnisausgabe
        static void AusgabeErgebnis(double iZahl1, double iZahl2, double iggT)
        {
            Console.WriteLine("Der ggT von {0} und {1}: {2} ", iZahl1, iZahl2, iggT);
        }

//-----
```

```

//Methode Eingabe Zahl: > 0
static double EingabeZahlGroesserAlsNull(double izaahl)
{
    string sZahl1 = ""; //Deklaration
    double iZahl1 = 0;

    do //solange iZahl1 < 1
    {
        if (izaahl == 1) //Bedingung
        {
            Console.WriteLine("Bitte die erste Zahl eingeben: ");
        }
        else
        {
            Console.WriteLine("Und nun bitte die zweite Zahl eingeben: ");
        }

        sZahl1 = Console.ReadLine();
        iZahl1 = Convert.ToInt32(sZahl1);

        if (iZahl1 < 1) //Bedingung Ausgabe
        {
            Console.WriteLine("Die Zahl muss größer als Null sein");
        }
    } while (iZahl1 < 1);
    return iZahl1; //Rückgabe Zahl1
}

//-----

static double ggTBerechnen(double iZahl1, double iZahl2) //Methode ggT-
//Berechnung
{
    double iR = 0; //deklariert

    do //Schleife
    {
        iR = iZahl1 % iZahl2; //Modulo-Restoperator

        if (iR != 0)
        {
            iZahl1 = iZahl2; //Wertzuweisung
            iZahl2 = iR;
        }
    } while (iR != 0);

    return iZahl2; //Rückgabe
}
}
}

```
