

Kapitel 8 Beispiel 6

```
1 private: System::Void bt_zeichne_Click(System::Object^ sender,
                                     System::EventArgs^ e)
2 {
3     // Methode wird aufgerufen, wenn der Button "Zeichne" geklickt wird
4     // Deklaration der lokalen Variablen
5     int    xsa = 50;           // x-Richtung, Screen, Anfang
6     int    xse = 610;         // x-Richtung, Screen, Ende
7     int    ysa = 550;         // y-Richtung, Screen, Anfang
8     int    yse = 100;         // y-Richtung, Screen, Ende
9     double xwa = -0.05;        // x-Richtung, Welt, Anfang
10    double xwe = 1.0;          // x-Richtung, Welt, Ende
11    double ywa = -0.05;        // y-Richtung, Welt, Anfang
12    double ywe = 1.0;          // y-Richtung, Welt, Ende
13    double schrittw;           // Schrittweite Zeichnungserstellung
14    int     x1;                // x-Koordinate erster Punkt einer Linie
15    int     y1;                // y-Koordinate erster Punkt einer Linie
16    int     x2;                // x-Koordinate zweiter Punkt einer Linie
17    int     y2;                // y-Koordinate zweiter Punkt einer Linie
18    double xlauf, ylauf, xtemp, ytemp;
19    double alpha;              // Koeffizient in f(x)
20    // Create font and brush.
21    System::Drawing::Font^ dF = gcnew System::Drawing::Font("Arial",10);
22    SolidBrush^ dB = gcnew SolidBrush( Color::Black );
23    // Erzeugung eines neuen Grafik-Objektes
24    Graphics ^g = this->CreateGraphics();
25    // Definition der Hintergrundfarbe durch R-G-B-Code
26    Color cl = Control::BackColor;
27    // Bereinigen des Grafik-Objektes für Folgeaufruf
28    g->Clear(cl);
29
30    // Übernahme der Einträge aus den TextBoxen mit try - catch
31    // fehlerhafte Eingabe ==> MessageBox
32    try
33    {
34        // Auslesen der Schrittweite, textBox ==> double
35        schrittw = Convert::ToDouble(tB_schrittweite->Text);
36        // Koeffizient der Funktion f(x)
37        alpha = Convert::ToDouble(tB_alpha->Text);
38        // Überprüfung alpha >= 0.0
39        if (alpha < 0.0)
40        {
41            MessageBox::Show("alpha muss positiv sein! - bitte
                                korrigieren!", "Eingabefehler");
42        }
43    }
44    else
45    {
46        // Kontrollrahmen für die Zeichnungsfläche
47        g->DrawRectangle(gcnew Pen(Color::Blue), xsa, yse,
48                        xse-xsa, ysa-yse);
49        // Konstruktion des Koordinatensystems
50        // x-Achse: Strich von (xwa,0) nach (xwe,0)
51        x1 = trans(xsa, xse, xwa, xwe, xwa);
52        y1 = trans(ysa, yse, ywa, ywe, 0.);
53        x2 = trans(xsa, xse, xwa, xwe, xwe);
54        g->DrawLine(gcnew Pen(Color::Black), x1, y1, x2, y1);
55        // x-Achse: Pfeilspitze
56        x1 = trans(xsa, xse, xwa, xwe, xwe);
57        x2 = trans(xsa, xse, xwa, xwe, xwe)-5;
58        y2 = trans(ysa, yse, ywa, ywe, 0.)+5;
59        g->DrawLine(gcnew Pen(Color::Black), x1, y1, x2, y2);
60        y2 = trans(ysa, yse, ywa, ywe, 0.)-5;
61        g->DrawLine(gcnew Pen(Color::Black), x1, y1, x2, y2);
62        // Beschriftung
63        g->DrawString("x", dF, dB, x1-10, y1-20);
64        // y-Achse: Strich von (0,ywa) nach (0,ywe)
65        x1 = trans(xsa, xse, xwa, xwe, 0.);
66        y1 = trans(ysa, yse, ywa, ywe, ywa);
67        y2 = trans(ysa, yse, ywa, ywe, ywe);
68        g->DrawLine(gcnew Pen(Color::Black), x1, y1, x1, y2);
69        // y-Achse: Pfeilspitze
70        y1 = trans(ysa, yse, ywa, ywe, ywe);
71        x2 = trans(xsa, xse, xwa, xwe, 0.)-5;
72        y2 = trans(ysa, yse, ywa, ywe, ywe)+5;
73        g->DrawLine(gcnew Pen(Color::Black), x1, y1, x2, y2);
74        x2 = trans(xsa, xse, xwa, xwe, 0.)+5;
75        g->DrawLine(gcnew Pen(Color::Black), x1, y1, x2, y2);
76    }
77 }
```

```

74         // Beschriftung
75         g->DrawString("y",dF,dB,x1+10,y1);
76         // Skalierung der Achsen
77         for (xlauf = 0.0; xlauf <= 0.9; xlauf += 0.1)
78         {
79             x1 = trans(xsa,xse,xwa,xwe,xlauf);
80             y1 = trans(ysa,yse,ywa,ywe,0.);
81             g->DrawLine(gcnew Pen(Color::Black), x1, y1-2, x1, y1+2);
82             g->DrawString(""+xlauf,dF,dB,x1,y1);
83         }
84         // y-Achse
85         for (ylauf = 0.0; ylauf <= 0.9; ylauf += 0.1)
86         {
87             x1 = trans(xsa,xse,xwa,xwe,0.);
88             y1 = trans(ysa,yse,ywa,ywe,ylauf);
89             g->DrawLine(gcnew Pen(Color::Black), x1-2, y1, x1+2, y1);
90             if (ylauf != 0.) g->DrawString(""+ylauf,dF,dB,x1,y1);
91         }
92         // Konstruktion des Graphen
93         for (xlauf = schrittweite; xlauf <= xwe; xlauf += schrittw)
94         {
95             x1 = trans(xsa,xse,xwa,xwe,xlauf-schrittw);
96             xtemp = xlauf-schrittw;
97             ytemp = xtemp / ( 1 + alpha*xtemp - alpha*pow(xtemp,2));
98             y1 = trans(ysa,yse,ywa,ywe,ytemp);
99             x2 = trans(xsa,xse,xwa,xwe,xlauf);
100            ytemp = xlauf / ( 1 + alpha*xlauf - alpha*pow(xlauf,2));
101            y2 = trans(ysa,yse,ywa,ywe,ytemp);
102            g->DrawLine(gcnew Pen(Color::Red), x1, y1, x2, y2);
103        }
104    }
105 }
106 catch (Exception ^e)
107 {
108     MessageBox::Show("Eingabe nicht numerisch - bitte
                        korrigieren!","Eingabefehler");
109 }
110 }
111 private: System::Void Form1_Load(System::Object^ sender,
                                System::EventArgs^ e)
112 {
113     this->tB_alpha->Text = "2,5";
114     this->tB_schrittweite->Text = "0,001";
115 }

```