

Kapitel 8 Beispiel 5

```
1  #pragma endregion
2  private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e)
3  {
4      // Diese Methode wird beim Load der Maske aufgerufen
5      // Versorgung der Koordinatenfelder mit Anfangswerten (Defaults)
6      // Group "Weltkoordinaten"
7      this->tB_xwa->Text = "-5,0";
8      this->tB_xwe->Text = "5,0";
9      this->tB_schrittweite->Text = "0,01";
10     // Group "Funktion"
11     this->tB_a->Text = "1";
12     this->tB_b->Text = "2";
13     this->tB_c->Text = "3";
14 }
15 private: System::Void bt_zeichne_Click(System::Object^ sender,
16                                         System::EventArgs^ e)
17 {
18     // Methode wird aufgerufen, wenn Button geklickt wird
19     // Deklaration der lokalen Variablen
20     int xsa = 50; // x-Richtung, Screen, Anfang
21     int xse = 610; // x-Richtung, Screen, Ende
22     int ysa = 550; // y-Richtung, Screen, Anfang
23     int yse = 150; // y-Richtung, Screen, Ende
24     double xwa; // x-Richtung, Welt, Anfang aus Maske
25     double xwe; // x-Richtung, Welt, Ende aus Maske
26     double ywa = -1.; // y-Richtung, Welt, Anfang
27     double ywe = 1.; // y-Richtung, Welt, Ende
28     double schrittweite; // Schrittweite Zeichnungserstellung
29     int x1; // x-Koordinate erster Punkt Linie
30     int y1; // y-Koordinate erster Punkt Linie
31     int x2; // x-Koordinate zweiter Punkt Linie
32     int y2; // y-Koordinate zweiter Punkt Linie
33     double xlauf, ylauf, xtemp, ytemp;
34     double a, b, c; // Koeffizienten in  $ax^2 + bx + c$ 
35     int ssw; // Skalierungs-Schritt-Weite
36     // Create font and brush.
37     System::Drawing::Font^ dF1 = gcnew System::Drawing::Font("Arial",6);
38     System::Drawing::Font^ dF2 = gcnew System::Drawing::Font("Arial",10);
39     SolidBrush^ dB = gcnew SolidBrush( Color::Black );
40     // Erzeugung eines neuen Grafik-Objektes
41     Graphics ^g = this->CreateGraphics();
42     // Definition der Hintergrundfarbe durch R-G-B-Code
43     Color cl = Control::BackColor;
44     // Bereinigen des Grafik-Objektes für Folgeaufruf
45     g->Clear(cl);
46     // Übernahme der Einträge aus den TextBoxen mit try - catch
47     // fehlerhafte Eingabe ==> MessageBox
48     try
49     {
50         // Auslesen der Welt-Koordinaten, textBox ==> double
51         xwa = Convert::ToDouble(tB_xwa->Text);
52         xwe = Convert::ToDouble(tB_xwe->Text);
53         // Auslesen der Schrittweite, textBox ==> double
54         schrittweite = Convert::ToDouble(tB_schrittweite->Text);
55         // Koeffizienten der Funktion f(x)
56         a = Convert::ToDouble(tB_a->Text);
57         b = Convert::ToDouble(tB_b->Text);
58         c = Convert::ToDouble(tB_c->Text);
59         // Errechnung der max. und min. Werte für y aus f(x)
60         for (xlauf = xwa; xlauf <= xwe; xlauf += schrittweite)
61         {
62             ytemp = a * pow(xlauf,2) + b*xlauf +c;
63             if (ytemp < ywa) ywa = ytemp;
64             if (ytemp > ywe) ywe = ytemp;
65         }
66         // "Glätten" der Werte
67         ywa = (float)((int)(ywa - 0.99));
68         ywe = (float)((int)(ywe + 0.99));
69         // Kontrollrahmen für die Zeichnungsfläche
70         g->DrawRectangle(gcnew Pen(Color::Blue),xsa,yse,xse-xsa,ya-yse);
71         // Konstruktion des Koordinatensystems
72         // x-Achse: Strich von (xwa,0) nach (xwe,0)
73         x1 = trans(xsa,xse,xwa,xwe,xwa);
74         y1 = trans(ysa,yse,ywa,ywe,0.);
75         x2 = trans(xsa,xse,xwa,xwe,xwe);
76         g->DrawLine(gcnew Pen(Color::Black), x1, y1, x2, y1);
```

```

76         // x-Achse: Pfeilspitze
77         x1 = trans(xsa,xse,xwa,xwe,xwe);
78         x2 = trans(xsa,xse,xwa,xwe,xwe)-5;
79         y2 = trans(ysa,yse,ywa,ywe,0.)+5;
80         g->DrawLine(gcnew Pen(Color::Black), x1, y1, x2, y2);
81         y2 = trans(ysa,yse,ywa,ywe,0.)-5;
82         g->DrawLine(gcnew Pen(Color::Black), x1, y1, x2, y2);
83         // Beschriftung
84         g->DrawString("x",dF2,dB,x1-10,y1-20);
85         // y-Achse: Strich von (0,ywa) nach (0,ywe)
86         x1 = trans(xsa,xse,xwa,xwe,0.);
87         y1 = trans(ysa,yse,ywa,ywe,ywa);
88         y2 = trans(ysa,yse,ywa,ywe,ywe);
89         g->DrawLine(gcnew Pen(Color::Black), x1, y1, x1, y2);
90         // y-Achse: Pfeilspitze
91         y1 = trans(ysa,yse,ywa,ywe,ywe),
92         x2 = trans(xsa,xse,xwa,xwe,0.)-5;
93         y2 = trans(ysa,yse,ywa,ywe,ywe)+5;
94         g->DrawLine(gcnew Pen(Color::Black), x1, y1, x2, y2);
95         x2 = trans(xsa,xse,xwa,xwe,0.)+5;
96         g->DrawLine(gcnew Pen(Color::Black), x1, y1, x2, y2);
97         // Beschriftung
98         g->DrawString("y",dF2,dB,x1+10,y1);
99         // Skalierung der Achsen
100        // x-Achse
101        ssw = 1;
102        if (xwe-xwa > 20.) ssw = 2;
103        if (xwe-xwa > 50.) ssw = 5;
104        for (xlauf = (int)(xwa+0.99); xlauf <= (int)(xwe-0.99); xlauf += ssw)
105        {
106            x1 = trans(xsa,xse,xwa,xwe,xlauf);
107            y1 = trans(ysa,yse,ywa,ywe,0.);
108            g->DrawLine(gcnew Pen(Color::Black), x1, y1-2, x1, y1+2);
109            g->DrawString(""+xlauf,dF1,dB,x1,y1);
110        }
111        // y-Achse
112        ssw = 1;
113        if (ywe-ywa > 20.) ssw = 2;
114        if (ywe-ywa > 50.) ssw = 5;
115        for (ylauf = (int)(ywa+0.99); ylauf <= (int)(ywe-0.99); ylauf += ssw)
116        {
117            x1 = trans(xsa,xse,xwa,xwe,0.);
118            y1 = trans(ysa,yse,ywa,ywe,ylauf);
119            g->DrawLine(gcnew Pen(Color::Black), x1-2, y1, x1+2, y1);
120            if (ylauf != 0.) g->DrawString(""+ylauf,dF1,dB,x1,y1);
121        }
122        // Konstruktion des Graphen
123        for (xlauf = xwa + schrittweite; xlauf <= xwe; xlauf += schrittweite)
124        {
125            x1 = trans(xsa,xse,xwa,xwe,xlauf-schrittweite);
126            xtemp = xlauf-schrittweite;
127            ytemp = a * pow(xtemp,2) + b*xtemp +c;
128            y1 = trans(ysa,yse,ywa,ywe,ytemp);
129            x2 = trans(xsa,xse,xwa,xwe,xlauf);
130            ytemp = a * pow(xlauf,2) + b*xlauf +c;
131            y2 = trans(ysa,yse,ywa,ywe,ytemp);
132            g->DrawLine(gcnew Pen(Color::Red), x1, y1, x2, y2);
133        }
134    }
135    catch (Exception ^e)
136    {
137        MessageBox::Show("Eingabe nicht numerisch - bitte
                                korrigieren!","Eingabefehler");
138    }
139 }
140 };
141 }

```