

## Kapitel 9 Beispiel 1

```
1 void bt_Auswertung_Click(System::Object^ send, System::EventArgs^ e)
2 {
3     // Diese Methode wird aufgerufen, wenn der Button "Auswertung"
4     // geklickt wird
5     // Deklaration der lokalen Variablen
6     int xsa = 10; // x-Richtung, Screen, Anfang
7     int xse = 510; // x-Richtung, Screen, Ende
8     int ysa = 550; // y-Richtung, Screen, Anfang
9     int yse = 130; // y-Richtung, Screen, Ende
10    double xwa; // x-Richtung, Welt, Anfang aus Maske
11    double xwe; // x-Richtung, Welt, Ende aus Maske
12    double ywa = -1.; // y-Richtung, Welt, Anfang
13    double ywe = 1.; // y-Richtung, Welt, Ende
14    double sw = 0.001; // Schrittweite zur Zeichnungserstellung
15    int x1; // x-Koordinate des 1. Punktes einer Linie
16    int y1; // y-Koordinate des 1. Punktes einer Linie
17    int x2; // x-Koordinate des 2. Punktes einer Linie
18    int y2; // y-Koordinate des 2. Punktes einer Linie
19    double xlauf, ylauf, xtemp, ytemp; // Hilfsvariablen
20    double a, b, c, d; // Koeffizienten in
21    //  $f(x) = ax^3 + bx^2 + cx + d$ 
22    double af1s, bf1s, cf1s; // Koeffizienten der 1. Ableitung
23    //  $f'(x) = 3ax^2 + 2bx + c$ 
24    double af2s, bf2s; // Koeffizienten der 2. Ableitung
25    //  $f''(x) = 6ax + 2b$ 
26    double af3s; // Koeffizienten der 3. Ableitung
27    //  $f'''(x) = 6a$ 
28    int ssw; // Skalierungs-Schritt-Weite
29    // Create font and brush.
30    System::Drawing::Font^ dF1 = gcnew System::Drawing::
31    Font("Arial", 6);
32    System::Drawing::Font^ dF2 = gcnew System::Drawing::
33    Font("Arial", 10);
34    SolidBrush^ dB = gcnew SolidBrush( Color::Black );
35    // Erzeugung eines neuen Grafik-Objektes
36    Graphics ^g = this->CreateGraphics();
37    // Definition der Hintergrundfarbe durch R-G-B-Code
38    Color cl = Control::BackColor;
39    // Bereinigen des Grafik-Objektes für Folgeaufruf
40    g->Clear(cl);
41
42    // Übernahme der Einträge aus den TextBoxen mit try - catch
43    // fehlerhafte Eingabe ==> MessageBox
44    try
45    {
46        // Auslesen der Welt-Koordinaten, textBox ==> double
47        xwa = Convert::ToDouble(tB_xwa->Text);
48        xwe = Convert::ToDouble(tB_xwe->Text);
49        // Koeffizienten der Funktion f(x)
50        a = Convert::ToDouble(tB_a->Text);
51        b = Convert::ToDouble(tB_b->Text);
52        c = Convert::ToDouble(tB_c->Text);
53        d = Convert::ToDouble(tB_d->Text);
54        // Berechnung der Ableitungen
55        af1s = 3.*a;
56        bf1s = 2.*b;
57        cf1s = c;
58        af2s = 6.*a;
59        bf2s = bf1s;
60        af3s = af2s;
61        // Label für f'(x)
62        this->lb_f1s->ForeColor = Color::Blue;
63        this->lb_f1s->Text = "f'(x) = "+System::Convert::
64        ToString(af1s)+"x^2";
65
66        if (bf1s >= 0.0)
67        {
68            this->lb_f1s->Text += " + "+System::Convert::
69            ToString(bf1s)+"x";
70        }
71        else
72        {
73            this->lb_f1s->Text += " "+System::Convert::
74            ToString(bf1s)+"x";
75        }
76        if (cf1s >= 0.0)
77        {
78            this->lb_f1s->Text += " + "+System::Convert::
79            ToString(cf1s);
80        }
81        else
82        {
83        }
```

```

71         this->lb_f1s->Text += " "+System::Convert::
                                   ToString(cfls);
72     }
73     // Label für f''(x)
74     this->lb_f2s->ForeColor = Color::Brown;
75     this->lb_f2s->Text = "f''(x) = "+System::Convert::
                                   ToString(af2s)+"x";
76     if (bf2s >= 0.0)
77     {
78         this->lb_f2s->Text += " + "+System::Convert::
                                   ToString(bf2s);
79     }
80     else
81     {
82         this->lb_f2s->Text += " "+System::Convert::ToString(bf2s);
83     }
84     // Label für f'''(x)
85     this->lb_f3s->ForeColor = Color::Green;
86     this->lb_f3s->Text = "f'''(x) = "+System::Convert::
                                   ToString(af3s);
87     // Errechnung der max. und min. Werte für y aus f(x), f'(x),
88     // f''(x) und f'''(x)
89     for (xlauf = xwa; xlauf <= xwe; xlauf += sw)
90     {
91         ytemp = a*pow(xlauf,3) + b*pow(xlauf,2) + c*xlauf + d;
92         if (ytemp < ywa) ywa = ytemp;
93         if (ytemp > ywe) ywe = ytemp;
94         ytemp = afls*pow(xlauf,2) + bfls*xlauf + cfls;
95         if (ytemp < ywa) ywa = ytemp;
96         if (ytemp > ywe) ywe = ytemp;
97         ytemp = af2s*xlauf + bf2s;
98         if (ytemp < ywa) ywa = ytemp;
99         if (ytemp > ywe) ywe = ytemp;
100        ytemp = af3s;
101        if (ytemp < ywa) ywa = ytemp;
102        if (ytemp > ywe) ywe = ytemp;
103    }
104    // "Glätten" der Werte
105    ywa = (float)((int)(ywa - 0.99));
106    ywe = (float)((int)(ywe + 0.99));
107    // Kontrollrahmen für die Zeichnungsfläche
108    g->DrawRectangle(gcnew Pen(Color::Blue), xsa, yse, xse-xsa,
109                                                            ysa-yse);
110
111    // Konstruktion des Koordinatensystems
112    // x-Achse: Strich von (xwa,0) nach (xwe,0)
113    x1 = trans(xsa,xse,xwa,xwe,xwa);
114    y1 = trans(ysa,yse,ywa,ywe,0.);
115    x2 = trans(xsa,xse,xwa,xwe,xwe);
116    g->DrawLine(gcnew Pen(Color::Black), x1, y1, x2, y1);
117    // x-Achse: Pfeilspitze
118    x1 = trans(xsa,xse,xwa,xwe,xwe);
119    x2 = trans(xsa,xse,xwa,xwe,xwe)-5;
120    y2 = trans(ysa,yse,ywa,ywe,0.)+5;
121    g->DrawLine(gcnew Pen(Color::Black), x1, y1, x2, y2);
122    y2 = trans(ysa,yse,ywa,ywe,0.)-5;
123    g->DrawLine(gcnew Pen(Color::Black), x1, y1, x2, y2);
124    // Beschriftung
125    g->DrawString("x",dF2,dB,x1-10,y1-20);
126    // y-Achse: Strich von (0,ywa) nach (0,ywe)
127    x1 = trans(xsa,xse,xwa,xwe,0.);
128    y1 = trans(ysa,yse,ywa,ywe,ywa);
129    y2 = trans(ysa,yse,ywa,ywe,ywe);
130    g->DrawLine(gcnew Pen(Color::Black), x1, y1, x1, y2);
131    // y-Achse: Pfeilspitze
132    y1 = trans(ysa,yse,ywa,ywe,ywe);
133    x2 = trans(xsa,xse,xwa,xwe,0.)-5;
134    y2 = trans(ysa,yse,ywa,ywe,ywe)+5;
135    g->DrawLine(gcnew Pen(Color::Black), x1, y1, x2, y2);
136    x2 = trans(xsa,xse,xwa,xwe,0.)+5;
137    g->DrawLine(gcnew Pen(Color::Black), x1, y1, x2, y2);
138    // Beschriftung
139    g->DrawString("y",dF2,dB,x1+10,y1);
140    // Skalierung der Achsen
141    // x-Achse
142    ssw = 1;
143    if (xwe-xwa > 20.) ssw = 2;
144    if (xwe-xwa > 50.) ssw = 5;
145    for (xlauf=(int)(xwa+0.99);xlauf<=(int)(xwe-0.99);
146         xlauf+=ssw)
147    {
148        x1 = trans(xsa,xse,xwa,xwe,xlauf);
149        y1 = trans(ysa,yse,ywa,ywe,0.);
150        g->DrawLine(gcnew Pen(Color::Black), x1, y1-2, x1, y1+2);
151        g->DrawString(""+xlauf,dF1,dB,x1,y1);
152    }

```

```

148     }
149     // y-Achse
150     ssw = 1;
151     if (ywe-ywa > 20.) ssw = 2;
152     if (ywe-ywa > 50.) ssw = 5;
153     if (ywe-ywa > 100.) ssw = 10;
154     // positive y-Achse (von 0 ausgehend)
155     for (ylauf = 0; ylauf <= (int)(ywe-0.99); ylauf += ssw)
156     {
157         x1 = trans(xsa,xse,xwa,xwe,0.);
158         y1 = trans(ysa,yse,ywa,ywe,ylauf);
159         g->DrawLine(gcnew Pen(Color::Black), x1-2, y1, x1+2, y1);
160         if (ylauf != 0.) g->DrawString(""+ylauf,dF1,dB,x1,y1);
161     }
162     // negative y-Achse (von 0 ausgehend)
163     for (ylauf = 0; ylauf >= (int)(ywa+0.99); ylauf -= ssw)
164     {
165         x1 = trans(xsa,xse,xwa,xwe,0.);
166         y1 = trans(ysa,yse,ywa,ywe,ylauf);
167         g->DrawLine(gcnew Pen(Color::Black), x1-2, y1, x1+2, y1);
168         if (ylauf != 0.) g->DrawString(""+ylauf,dF1,dB,x1,y1);
169     }
170     // Konstruktion der Graphen
171     // 1. f(x)
172     for (xlauf = xwa + sw; xlauf <= xwe; xlauf = xlauf + sw)
173     {
174         x1 = trans(xsa,xse,xwa,xwe,xlauf-sw);
175         xtemp = xlauf-sw;
176         ytemp = a*pow(xtemp,3) + b*pow(xtemp,2) + c*xtemp + d;
177         y1 = trans(ysa,yse,ywa,ywe,ytemp);
178         x2 = trans(xsa,xse,xwa,xwe,xlauf);
179         ytemp = a*pow(xlauf,3) + b*pow(xlauf,2) + c*xlauf + d;
180         y2 = trans(ysa,yse,ywa,ywe,ytemp);
181         g->DrawLine(gcnew Pen(Color::Red), x1, y1, x2, y2);
182     }
183     // 2. f'(x)
184     for (xlauf = xwa + sw; xlauf <= xwe; xlauf = xlauf + sw)
185     {
186         x1 = trans(xsa,xse,xwa,xwe,xlauf-sw);
187         xtemp = xlauf-sw;
188         ytemp = af1s*pow(xtemp,2) + bf1s*xtemp + cf1s;
189         y1 = trans(ysa,yse,ywa,ywe,ytemp);
190         x2 = trans(xsa,xse,xwa,xwe,xlauf);
191         ytemp = af1s*pow(xlauf,2) + bf1s*xlauf + cf1s;
192         y2 = trans(ysa,yse,ywa,ywe,ytemp);
193         g->DrawLine(gcnew Pen(Color::Blue), x1, y1, x2, y2);
194     }
195     // 3. f''(x)
196     for (xlauf = xwa + sw; xlauf <= xwe; xlauf = xlauf + sw)
197     {
198         x1 = trans(xsa,xse,xwa,xwe,xlauf-sw);
199         xtemp = xlauf-sw;
200         ytemp = af2s*xtemp + bf2s;
201         y1 = trans(ysa,yse,ywa,ywe,ytemp);
202         x2 = trans(xsa,xse,xwa,xwe,xlauf);
203         ytemp = af2s*xlauf + bf2s;
204         y2 = trans(ysa,yse,ywa,ywe,ytemp);
205         g->DrawLine(gcnew Pen(Color::Brown), x1, y1, x2, y2);
206     }
207     // 4. f'''(x)
208     for (xlauf = xwa + sw; xlauf <= xwe; xlauf = xlauf + sw)
209     {
210         x1 = trans(xsa,xse,xwa,xwe,xlauf-sw);
211         xtemp = xlauf-sw;
212         ytemp = af3s;
213         y1 = trans(ysa,yse,ywa,ywe,ytemp);
214         x2 = trans(xsa,xse,xwa,xwe,xlauf);
215         ytemp = af3s;
216         y2 = trans(ysa,yse,ywa,ywe,ytemp);
217         g->DrawLine(gcnew Pen(Color::Green), x1, y1, x2, y2);
218     }
219 }
220 catch (Exception ^e)
221 {
222     MessageBox::Show("Eingabe nicht numerisch - bitte
223                       korrigieren!", "Eingabefehler");
224 }

```