

```

1  using System;
2  using System.Windows.Forms;
3  using SldWorks;
4  using System.Runtime.InteropServices;
5
6  // Ausgangspunkt: SolidWorks gestartet, keine Datei geladen
7  // MathCAD geöffnet, Datei Kurve.xcmd geladen
8
9  namespace SW_Macro_MathCAD
10 {
11     public partial class Form1 : Form
12     {
13         SldWorks.SldWorks swApp;
14
15         public Form1()
16         {
17             InitializeComponent();
18
19             try
20             {
21                 swApp = (SldWorks.SldWorks)Marshal.GetActiveObject      ➤
22                     ("SldWorks.Application");
23             }
24             catch (Exception)
25             {
26                 swApp = new SldWorks.SldWorks();
27                 swApp.Visible = true;
28             }
29         }
30
31         //
32         private void b_create_Click(object sender, EventArgs e)
33         {
34             // Deklarieren der benötigten Variablen
35             double a0x, a0y, lh, rg, PMax, PsiMax;
36
37             // Auslesen der eingegebenen Werte im Formular
38             if (double.TryParse(label_1.Text, out a0x) == false ||
39                 double.TryParse(label_2.Text, out a0y) == false ||
40                 double.TryParse(label_3.Text, out lh) == false ||
41                 double.TryParse(label_4.Text, out rg) == false ||
42                 double.TryParse(tb_pmax.Text, out PMax) == false ||
43                 double.TryParse(label_6.Text, out PsiMax) == false)
44             {
45                 MessageBox.Show("Fehlerhafte Eingabe!");
46                 return;
47             }
48
49             // Erzeugen eines neuen Bauteils
50             ModelDoc2 swModel = (ModelDoc2)swApp.NewPart();
51
52             // Herstellen zwischen MatCAD und aktives Arbeitsblatt auslesen

```

```

53      Matcad.Application mcApp = (Mathcad.Application)
           Marshal.GetActiveObject("Mathcad.Applicaiton");
54      Mathcad.Worksheet mcWks = mcApp.ActiveWorksheet;
55
56      // neuer Sketch in SolidWorks
57      swModel.Extension.SelectByID2("Ebene vorne", "PLANE", 0, 0, 0,
           false, 0, null, 0);
58      swModel.InsertSketch2(true);
59
60      // Setzen der Parameter im MathCAD Arbeitsblatt
61      mcWks.SetValue("a0x", a0x);
62      mcWks.SetValue("a0y", a0y);
63      mcWks.SetValue("lh", lh);
64      mcWks.SetValue("rg", rg);
65      mcWks.SetValue("pmax", PMax);
66      mcWks.SetValue("psimax", PsiMax);
67
68      double xa = 0.0, ya = 0.0;
69      double xk, yk;
70      Mathcad.NumericValue mcVal;
71
72      // Schleife zur Berechnung der Stützpunkte
73      for (int i = 0; i < PMax; i++)
74      {
75          // Setzen des Parameters für aktuellen Stützpunkt
76          mcWks.SetValue("pact", i);
77
78          // Auslesen der berechneten Koordinaten aus MathCAD
79          mcVal = (Mathcad.NumericValue)mcWks.GetValue("X");
80          xk = mcVal.Real;
81
82          mcVal = (Mathcad.NumericValue)mcWks.GetValue("Y");
83          yk = mcVal.Real;
84
85          if (i > 0)
86          {
87              swModel.CreateLine2(xa / 1000.0, ya / 1000.0, 0, xk /
88                  1000.0, yk / 1000.0, 0);
89          }
90          xa = xk;
91          ya = yk;
92      }
93
94      // Kreis zeichnen und extrudieren
95      swModel.CreateCircleByRadius2(a0x / 1000.0, a0y / 1000.0, 0.0,
           3.0 / 1000.0);
96
97      swModel.FeatureManager.FeatureExtrusion2(true, false, false, 0,
           0,
98          0.01, 0, false, false, false, false, 0, 0, false, false,
           false,
99          false, false, false, false, 0, 0, false);
100  }

```

```

101
102 //
103 private void b_createWithSpline_Click(object sender, EventArgs e)
104 {
105     double a0x, a0y, lh, rg, PMax, PsiMax;
106
107     // Auslesen der eingegebenen Werte im Formular
108     if (double.TryParse(tb_a0x.Text, out a0x) == false ||
109         double.TryParse(tb_a0y.Text, out a0y) == false ||
110         double.TryParse(tb_lh.Text, out lh) == false ||
111         double.TryParse(tb_rg.Text, out rg) == false ||
112         double.TryParse(tb_pmax.Text, out PMax) == false ||
113         double.TryParse(tb_psimax.Text, out PsiMax) == false)
114     {
115         MessageBox.Show("Fehlerhafte Eingabe!");
116         return;
117     }
118
119     // Erzeugen eines neuen Bauteils
120     ModelDoc2 swModel = (ModelDoc2)swApp.NewPart();
121
122     // Herstellen zwischen MatCAD und aktives Arbeitsblatt auslesen
123     Mathcad.Application mcApp = (Mathcad.Application)
124         Marshal.GetActiveObject("Mathcad.Applicaiton");
125     Mathcad.Worksheet mcWks = mcApp.ActiveWorksheet;
126
127     // neuer Sketch in SolidWorks
128     swModel.Extension.SelectByID2("Ebene vorne", "PLANE", 0, 0, 0,
129         false, 0, null, 0);
130     swModel.InsertSketch2(true);
131
132     // Setzen der Parameter im MathCAD Arbeitsblatt
133     mcWks.SetValue("a0x", a0x);
134     mcWks.SetValue("a0y", a0y);
135     mcWks.SetValue("lh", lh);
136     mcWks.SetValue("rg", rg);
137     mcWks.SetValue("psimax", PsiMax);
138     mcWks.SetValue("pcnt", PMax);
139
140     double[] PointData = new double[(int)(PMax * 3 + 3)];
141     int pIndex = 0;
142     double xk, yk;
143     Mathcad.NumericValue mcVal;
144
145     // Schleife zur Berechnung der Stützpunkte
146     for (int i = 0; i < PMax; i++)
147     {
148         // Setzen des Parameters für aktuellen Stützpunkt
149         mcWks.SetValue("pact", i);
150
151         // Auslesen der berechneten Koordinaten aus MathCAD
152         mcVal = (Mathcad.NumericValue)mcWks.GetValue("X");

```

```

152         xk = mcVal.Real;
153
154         mcVal = (Mathcad.NumericValue)mcWks.GetValue( "Y");
155         yk = mcVal.Real;
156
157         PointData[pIndex] = xk / 1000.0;
158         PointData[pIndex + 1] = yk / 1000.0;
159         PointData[pIndex + 2] = 0;
160         pIndex = pIndex + 3;
161     }
162
163     swModel.CreateSpline(PointData);
164     swModel.CreateCircleByRadius2(a0x / 1000.0, a0y / 1000.0, 0.0, 3.0 / 1000.0);
165
166     swModel.FeatureManager.FeatureExtrusion2(true, false, false, 0,
        0,
167         0.01, 0, false, false, false, false, 0, 0, false, false,
            false,
168         false, false, false, false, 0, 0, false);
169     }
170 }
171 }
172

```