

Übung zur Veranstaltung: Informationstechniken zur Wissens- integration in Engineering-Prozesse (IWEPE)

ProEngineer / Creo VB API-Programmierung am Beispiel einer Behälterberechnung

Prof. Dr.-Ing. Dipl.-Math. Peter Köhler
Thivakar Manoharan M. Sc.
Dipl.-Ing. (FH) Christoph Kesselmanns



Institut für Produkt Engineering

Visual Studio Einstellungen

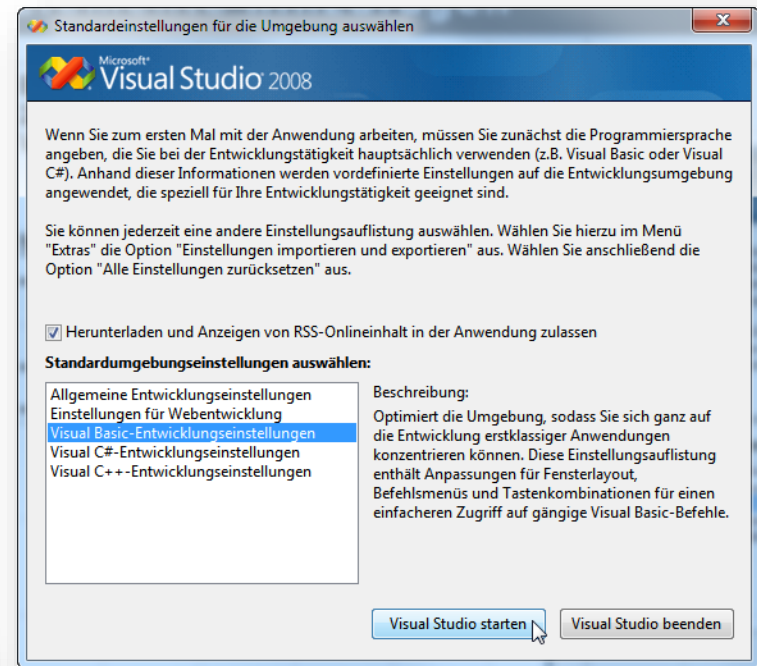
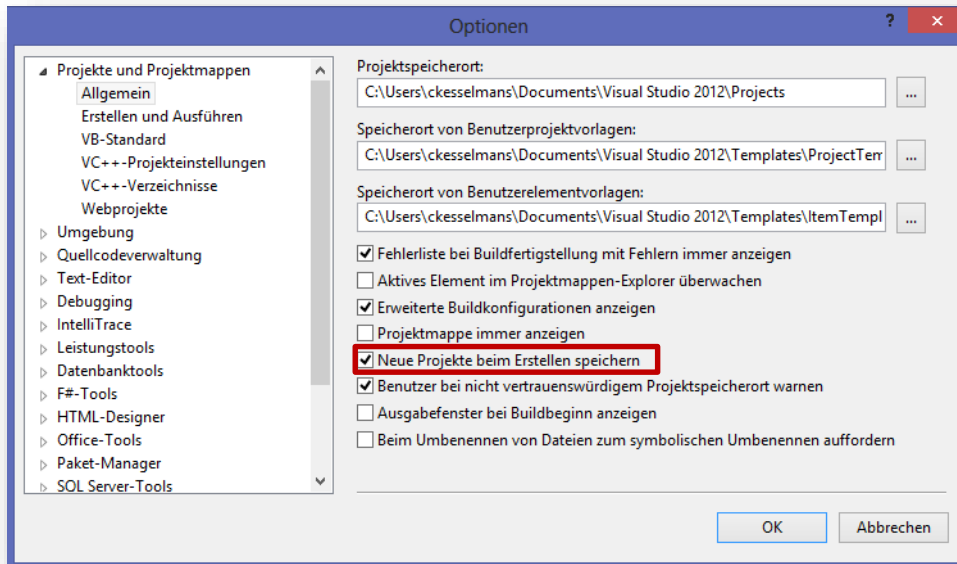
Beim erstmaligen Starten von Visual Studio 2012 ist die Entwicklungseinstellungen auf Visual Basic einzustellen.

Um den Speicherort beim Erstellen eines Projektes angeben zu können muss folgende Einstellung durchgeführt werden:

Tools ⇒ Optionen ⇒ Projekte und Projektmappen

⇒ Allgemein:

Neue Projekte beim Erstellen speichern



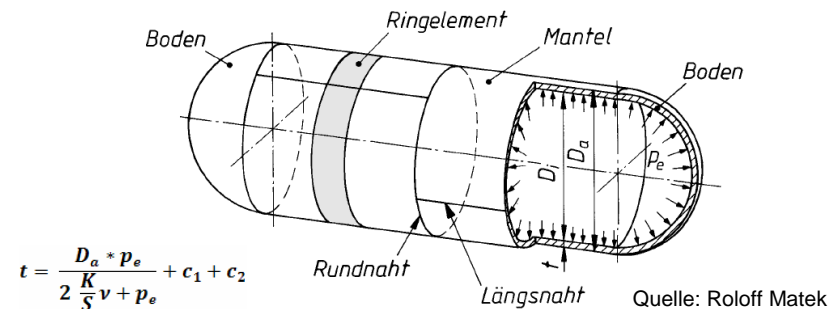
Druckbehälterberechnung

Aufgabe:

Es soll eine Windows Form - Anwendung entwickelt werden, um die Wanddicke von Druckbehältern nach dem AD 2000 - Regelwerk zu berechnen.

Anforderungen an das Projekt:

- Abfrage der Berechnungsparameter vom Anwender
- Benötigte Werte anhand von eingegebenen Parameter bestimmen (siehe Tabellen)
- ggf. Anzeige einer Behälterskizze
- Berechnung der Wanddicke t
- Plausibilitätskontrolle eingegebener Werte
- Ausgabe des berechneten Wertes für t



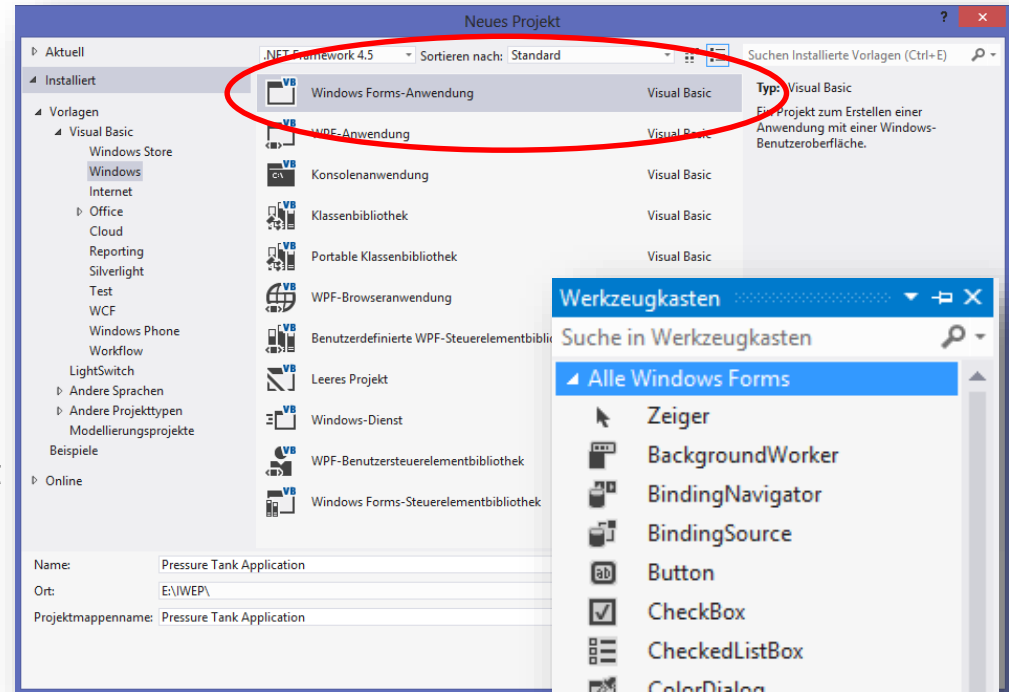
$$t = \frac{D_a \cdot p_e}{2 \frac{K}{S} \nu + p_e} + c_1 + c_2$$

Die Berechnung soll nur für ausgewählte Werkstoffe und nur für Temperaturen $T=20^\circ\text{C}$ erfolgen. Die benötigten Werte befinden sich in den entsprechenden Tabellen auf den folgenden Seiten. Die Druckbehälterformel ist ebenfalls gegeben.

Druckbehälterberechnung

Datei ⇒ Neues Projekt :

- Programmiersprache: Visual Basic (.NET Framework 4.5)
- Projekttyp: Windows-Anwendung
- Windows Forms-Anwendung (Name und Speicherort angeben!)
- Neu angelegte Steuerelemente (ComboBox, TextBox etc.) sollten stets mit sinnvollen Namen und einem Präfix versehen werden



Control-Typ	Präfix	Beispiel
Check box	chk	chkFase
Combo box	cbo	cboWerkstoff
Command button	cmd	cmdBeenden
Form	frm	frmStart
Label	lbl	lblDurchmesser
Picture	pic	picSkizze
Text box	txt	txtParameter1

Druckbehälterberechnung

The image shows a screenshot of Microsoft Visual Studio in Design mode for a Windows Form. The interface includes a menu bar, a toolbar, a central design area, and several toolboxes on the right. Annotations with arrows point to various parts of the interface:

- Debug-Modus**: Points to the 'Debug' dropdown in the toolbar.
- Werkzeugkasten**: Points to the 'Werkzeugkasten' (Toolbox) on the left side.
- aktuelle Windows Form**: Points to the 'Form1' design area.
- zeigt Quellcode**: Points to the 'Projektmappen-Explorer' (Solution Explorer) on the right.
- Projektmappe**: Points to the project structure in the Solution Explorer.
- Eigenschaften des Objektes Aktuell: Button btn_start**: Points to the 'Eigenschaften' (Properties) window at the bottom right, which shows the properties of the selected 'btn_start' button.

The 'Eigenschaften' window displays the following properties for the 'btn_start' button:

Property	Value
(Name)	btn_start
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
AllowDrop	False
Anchor	Top, Left
AutoEllipsis	False
AutoSize	False
AutoSizeMode	GrowOnly
BackColor	Control

Druckbehälterberechnung

Die GUI (engl. „Graphical User Interface“) des „Frm1.vb“ soll entsprechend dem Designentwurf mit Hilfe des Werkzeugkastens (Ansicht \Rightarrow Werkzeugkasten) erstellt werden.

Entwurf Pressure Tank Application
Dienstag, 8. Mai 2012

Pressure Tank Application

Eingabewerte:

Äußerer Durchmesser [Da] mm

Betriebsdruck [pe] bar

Werkstoff ▼

Ausnutzungsfaktor Schweißnaht [v] ▼

Zuschlag Wanddickenunter. [c₁] kein Zuschlag (0 mm) mm

Wandstärke [t] mm

Berechnen

Beenden

Label

TextBox

ComboBox

CheckBox

TextBox

Label

Buttons

Druckbehälterberechnung

Entwurf Pressure Tank Application

Dienstag, 8. Mai 2012

ComboBox Werkstoffe

- Angezeigt werden die Kurznamen (S235JRG1 etc.) der Werkstoffe
- Bei Auswahl werden den Variablen k, S, und C2 entsprechende Werte zugewiesen (siehe Tabelle) (Select Case Anweisung)

CheckBox und TextBox Wanddickenunterschreitung

- Standardeinstellung CheckBox checked -> TextBox ist deaktiviert (enabled=false)
- Einstellung CheckBox unchecked -> TextBox wird aktiviert (enabled=true)

Werkstoff	Kurzname	Kennwert	k bei 20°C	Werkstoff und Ausführung	S	C2	Dichte in kg/dm ³
unlegierter Stahl	S235JRG1	R _{P0,2}	235	Walz-Schmiedestahl	1,5	1	7,85
nichtrostender austenitischer Stahl	X5CrNi18-10	R _{P0,1}	210	Walz-Schmiedestahl	1,5	0	7,9
Stahlguss	GS-45	R _{P0,2}	230	Stahlguss	2,0	1	7,8
Aluminium	AlMg3	R _{P0,2}	80	Aluminiumknetlegierung	1,5	0	2,7

Quelle Roloff Matek

Druckbehälterberechnung

Entwurf Pressure Tank Application

Dienstag, 8. Mai 2012

Pressure Tank Application

Eingabewerte:

Äußerer Durchmesser [Da] mm

Betriebsdruck [p_g] bar

Werkstoff ▼

Ausnutzungsfaktor Schweißnaht [v] ▼

Zuschlag Wanddickenunter. [c₁] kein Zuschlag (0 mm) mm

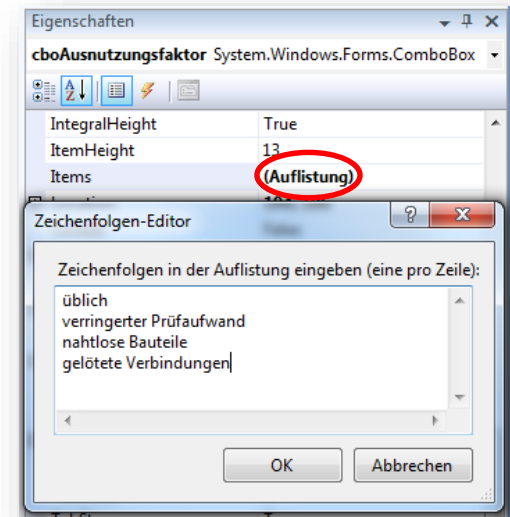
Wandstärke [t] mm

Berechnen Beenden

Combobox Ausnutzungsfaktor Schweißnaht v

- Angezeigt wird das Kriterium (üblich, verringerter Prüfaufwand ...)
- Bei Auswahl wird der Variablen v ein entsprechender Wert zugewiesen

Kriterium	v
üblich	1,0
verringertes Prüfaufwand	0,85
nahtlose Bauteile	1,0
gelötete Verbindungen	0,8



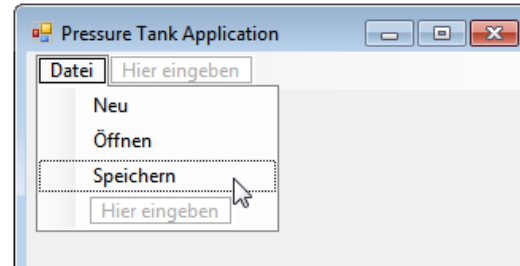
TextBox Wandstärke

- Eigenschaft: ReadOnly

Erweiterung 1:

Neue Anforderungen an das Projekt:

- Ergänzung des Formulars um Bearbeiter und Projekt eingeben zu können
- Menüleiste Datei -> Neu, Öffnen, Speichern: Dateiformate .txt und .xml nach vorgegebener Struktur



- Datei→Neu: Alle Eingabefelder werden geleert bzw. zurückgesetzt
- Datei→Öffnen: Der Standard „*Windows Datei öffnen*“ Dialog wird angezeigt
- Datei→Speichern : Der Standard „*Windows Datei speichern*“ Dialog wird angezeigt
- Zum Öffnen eines Projektes sind zwei Prozeduren anzulegen (ReadTxt und ReadXml) als Argument wird *openFileDialog* übergeben
- Zum Speichern eines Projektes sind ebenfalls zwei Prozeduren anzulegen (WriteTxt und WriteXml) als Argument wird *saveFileDialog* übergeben
- Diese Prozeduren sollen ausgelöst werden, wenn eine entsprechende Auswahl im Datei Öffnen/Speichern Dialog getroffen wurde
- Im Header müssen `Imports System.IO` und `Imports System.Xml` angegeben werden

Erweiterung 2:

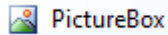
Neue Anforderungen an das Projekt:

- Problem: Sinnvolle Werteingaben werden nur bei Berechnung überprüft, nicht beim Speichern oder Öffnen -> Auslagerung der Plausibilitätskontrollen in separate Prozedur
- Erweiterung des Formulars
- Automatisierte Generierung des CAD-Modells nach aktueller Konfiguration
 - Übertragung der Werte Außendurchmesser, Wandstärke und Material, Bearbeiter an das parametrisierte CAD-Modell (Parameter editieren)
 - Löschen von Einzelteilen (Behälterenden) in der Baugruppe und Einbau von neuen, der Konfiguration entsprechenden Teilen
 - Möglichkeit Stutzenausschnitte in Behältermantel einzubringen (UDF)
- Rückgabe des Volumeninhaltes und der Masse des Behälters an die Anwendung Behälterberechnung
- Weitere sinnvolle Plausibilitätskontrollen (z.B. Button „Trennen“ kann nur geklickt werden, wenn Button „Verbindung“ vorher betätigt wurde)
- Zusätzlichen Informationen (Bearbeiter, Auftragsnummer, Mantellänge, Innenvolumen und Masse) in .txt und .xml Datei speichern
- ...

Druckbehälterberechnung

Ergänzungen am Eingabeformular

Bilder können über eine „Picture Box“ eingebunden werden



Groupboxen zur Abgrenzung. GroupBox

RadioButtons die in einer Groupbox zusammengefasst sind aktivieren und deaktivieren sich automatisch!

Default Werte über die Eigenschaft:

Text 2500

Nur Lesezugriff:

ReadOnly True

vorerst deaktiviert:

Enabled True

Druckbehälterberechnung

Ergänzungen am Eingabeformular

btnverbinden

btnBaugruppeAkt

btntrennen

btnUDF

btnEigenschBaugr

Druckbehälterberechnung

Projektstruktur

```
1 ' Behälterberechnung Übung IWEF VB.NET 2008
2 ' aktueller Bearbeitungsstand: 30.5.12
3
4 Imports pfcls
5 Imports System.Math
6 Imports System.IO
7 Imports System.Xml
8
9 Public Class PressureTankApplication
10
11     '*** globale Deklaration der Variablen, damit sie in allen Prozeduren/Funktionen bekannt sind
12     Dim Da, pe, K, S, v, C1, C2, t, Vol, m, L As Double
13     Dim Seite_1, Seite_2 As String
14     Dim asyncConnection As IpfcAsyncConnection = Nothing
15     Dim cAC As CCpfAsyncConnection
16     Dim session As IpfcBaseSession
17
18     '*** Buttons zur Berechnung
19
20     Private Sub btnBerechnen_Click ...
34
35     Private Sub ChkzuschlWanddicke_CheckedChanged ...
45
46     Private Sub cboWerkstoff_SelectedIndexChanged ...
69
70     Private Sub btnBeenden_Click ...
73
74     Private Sub cboAusnutzungsfaktor_SelectedIndexChanged ...
88
89     '*** ausgelagerte Funktionen ***
90
91     Function InputCheck ...
154
155     Private Sub berechnen ...
164
165     '*** Menüstreifen ***
166
167     Private Sub NeuToolStripMenuItem1_Click ...
180
181     Private Sub ÖffnenToolStripMenuItem1_Click ...
199
200     Private Sub SpeichernToolStripMenuItem1_Click ...
225
226     '*** Speichern und Schreiben als Text oder XML ***
227
228     Private Sub ReadTxt ...
259
260     Private Sub ReadXml ...
282
283     Private Sub WriteTxt ...
310
311     Private Sub WriteXML ...
```

Druckbehälterberechnung

Projektstruktur- Fortsetzung

```
384
385 '*** Auswahl der Endstücke ***
386
387 Private Sub rdioKugel1_CheckedChanged ...
400
401 Private Sub rdioKugel2_CheckedChanged ...
414
415
416 '*** CAD Kopplung Vorbedingung Creogestartet; Baugruppe Behaelter.asm aus dem Arbeitsverzeichnis geladen sein
417
418 #Region "Buttons CAD"
419
420 '*** Erstellt die Verbindung und schaltet weitere CAD-Buttons frei
421 Private Sub btnverbinden_Click ...
434
435 '*** Trennt die Verbindung zu Creo und deaktiviert die CAD-Buttons wieder
436 Private Sub btntrennen_Click ...
447
448 '*** Aktualisiert ALLE Creo-Teile mit den eingestellten Werten, Baut die aktuelle Konfiguration (checkboxbuttons) zusammen (Einbau der Komponenten)
449 '*** Baugruppe wird regeneriert, Fenster wird aktualisiert, Textboxen Masse und Volumen werden auf "" gesetzt, Button btnEigenschBaugr aktiviert
450 Private Sub btnBaugruppeAkt_Click ...
528
529 '*** Button "durchsucht die Baugruppe auf eingebaute Teile, liest aus den verbauten Teilen das Innenvolumen (Parameter V) aus, addiert diese auf
530 '*** schreibt das Ergebnis an "m" bzw. an txtMasse
531 Private Sub btnEigenschBaugr_Click ...
564
565 '*** Button löst zunächst einen Inputcheck aus (um Durchmesser = 0) zu vermeiden, ruft dann die Prozedur zum Einbau des UDF auf,
566 '*** regeneriert die Baugruppe und aktualisiert das aktuelle Fenster
567 Private Sub btnUDF_Click ...
588
589 #End Region
590
591 #Region "in den Buttons verwendet Unterfunktionen und Prozeduren"
592
593 '*** ändert einen Creo Parameter vom Typ String
594 Sub ParameterAendernString ...
607
608 '*** ändert einen Creo Parameter vom Typ Double
609 Sub ParameterAendernDouble ...
622
623 '*** liest den Wert eines Creo Double Parameters aus
624 Function ParamAusCreo ...
641
642 '*** gibt das "model" eines im Arbeitsverzeichnis abgelegten Teil oder Baugruppe zurück
643 '*** das Teil bzw. Baugruppe muss nicht in Creo (im Hintergrund) geöffnet sein!
644 Function Modell_Aus_Arbeitsverz ...
655
656 '*** Baut eine Komponente in die Baugruppe ein
657 Sub AssembleComponent ...
702
703 '*** Löscht ALLE Teile aus der Baugruppe bis auf Behaeltermantel.prt
704 Sub EraseComponents ...
733
734 '*** Baut das UDF in das Einzelteil Behaeltermantel.prt ein
735 Function createUDFInPart ...
800
801 #End Region
802
803 End Class
```

btnverbinden

```
Private Sub btnverbinden_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnverbinden.Click

    cAC = New CCpfcAsyncConnection()
    asyncConnection = cAC.Connect(Nothing, Nothing, Nothing, Nothing)
    session = asyncConnection(...)

    btnverbinden.Enabled = False
    btntrennen.Enabled = True
    (...)
    btnBeenden.Enabled = False

End Sub
```

btntrennen

```
Private Sub btntrennen_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btntrennen.Click

    asyncConnection(...)

    btnverbinden.Enabled = True

    (...)
    btnBeenden.Enabled = True

End Sub
```

Druckbehälterberechnung

btnBaugruppeAkt

```
Private Sub btnBaugruppeAkt_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnBaugruppeAkt.Click
```

```
    Dim model As IpfcModel
    Dim Assembly As IpfcAssembly
    Dim Part As IpfcPart
    Dim window As IpfcWindow
    Dim status As Boolean

    status = InputCheck() ' damit das UDF sinnvolle Werte erhält
    If status = 0 Then
        Exit Sub
    End If

    berechnen()

    model = Modell_Aus_Arbeitsverz("kugelboden_1.prt")
    ParameterAendernDouble(model, "t", t)
    ParameterAendernDouble(model, "da", Da)
    ParameterAendernString(model, "PTC_Material_Name", cboWerkstoff.SelectedItem)

    (...)

    Assembly = Modell_Aus_Arbeitsverz("behaelter.asm")
    EraseComponents(Assembly)

    Part = Modell_Aus_Arbeitsverz(Seite_1)
    AssembleComponent(...), (...), "ASM_KOS_SEITE_1", "PRT_KOS_SEITE_1")

    (...)
    → Fortsetzung nächste Seite
```

Unterfunktion bzw.
-Prozeduren stehen in
Kästchen

Unterfunktion Modell_Aus_Arbeitsverz

```
Function Modell_Aus_Arbeitsverz(ByVal Name As String) As IpfcModel
    Dim ImodelDscr As IpfcModelDescriptor
    Dim CModelDscr As New CCpfcModelDescriptor
    Dim model As IpfcModel
    ImodelDscr = CModelDscr.CreateFromFileName(Name)
    model = session.RetrieveModel(...)
    Return (...)
End Function
```

Unterprozedur ParameterAendernDouble

```
Sub ParameterAendernDouble(ByVal model As IpfcModel, ByVal Parametername As String, ByVal wert As Double)

    Dim ParaOwner As IpfcParameterOwner
    Dim para As IpfcParameter = Nothing
    Dim CValue As New (...)
    Dim Value As (...)
    Value = CValue.CreateDoubleParamValue(wert)
    ParaOwner = model
    para = ParaOwner.GetParam(...)
    para.SetScaledValue(...)

End Sub
```


Druckbehälterberechnung

btnBaugruppeAkt - Fortsetzung

```
'durch die Analyse des Innenvolumens ist zweimaliges regenerieren notwendig, die Funktion  
'regenerate liefert eine Ausnahme bzw. Fehler, wenn nach einmaligem regenerieren nicht alle KE's aktuell sind.  
'Diese Ausnahme wird abgefangen und löst ein weiteres Regenerieren aus.
```

```
model = Modell_Aus_Arbeitsverz("behaelter.asm")
```

```
regen:
```

```
Try  
    model.Regenerate(Nothing) 'regenerieren  
Catch ex As Exception  
    If ex.Message = "pfcExceptions::XToolkitRegenerateAgain" Then  
        GoTo regen  
    End If  
End Try
```

```
window = session.CurrentWindow() 'aktuelles Fenster holen und Bildneuaufbau  
window.(...)  
txtVolumen.Text = ""  
txtMasse.Text = ""  
btnEigenschBaugr.Enabled = True
```

```
End Sub
```

Unterprozedur EraseComponents

```
Sub EraseComponents(ByRef Assembly As IpfcAssembly)  
  
    Dim solid As IpfcSolid  
    Dim Features As IpfcFeatures  
    Dim Feature As IpfcFeature  
    Dim component As IpfcComponentFeat  
    Dim deleteop As IpfcDeleteOperation  
    Dim operations As New CpfcFeatureOperations  
    Dim modeldescriptor As IpfcModelDescriptor  
    solid = Assembly  
    Features = solid.ListFeaturesByType(False, (...))  
    For i As Integer = 0 To (... ) - 1  
        component = (...)  
        Feature = (...)  
        modeldescriptor = component.ModelDescr()  
        deleteop = Feature.CreateDeleteOp()  
        'MsgBox(modeldescriptor.GetFileName())  
        If modeldescriptor.GetFileName <> (...) Then  
            operations.Append(deleteop)  
        End If  
    Next  
    solid.ExecuteFeatureOps(..., Nothing)  
  
End Sub
```

```
Sub AssembleComponent(ByRef Baugruppe As IpfcAssembly, ByRef Teil As IpfcPart, _
    ByVal KOS_Baugr As String, ByVal KOS_Teil As String)

    Dim Feature As IpfcComponentFeat
    Dim ASMKOS_OW As IpfcModelItemOwner
    Dim PRTKOS_OW As IpfcModelItemOwner
    Dim ASMKOS As IpfcModelItem
    Dim PRTKOS As IpfcModelItem
    Dim constraint_cls As CCpfcComponentConstraint
    Dim constraint As IpfcComponentConstraint
    Dim constraints As IpfcComponentConstraints
    Dim ids As Cintseq
    Dim path As IpfcComponentPath
    Dim path_cls As CmpfcAssembly
    Dim asmSelect As IpfcSelection
    Dim asmSelect_cls As CmpfcSelect
    Dim prtSelect As IpfcSelection
    Dim prtSelect_cls As CmpfcSelect
    Dim window As IpfcWindow

    ASMKOS_OW = Baugruppe
    ASMKOS = ASMKOS_OW.GetItemByName(...)
    PRTKOS_OW = Teil
    PRTKOS = PRTKOS_OW.GetItemByName(...)
    constraint_cls = New CCpfcComponentConstraint
    constraint = constraint_cls.Create(EpfcComponentConstraintType.EpfcASM_CONSTRAINT_CSYS) 'Einbauoption Koordinatensystem wird definiert
    constraints = New CpfcComponentConstraints
    constraints.Insert(0, constraint)
    ids = New Cintseq
    path_cls = New CmpfcAssembly
    path = path_cls.CreateComponentPath(Baugruppe, ids)
    asmSelect_cls = New CmpfcSelect
    asmSelect = asmSelect_cls.CreateModelItemSelection(ASMKOS, path) 'Selektiert das Koordinatensystem in der Baugruppe
    constraint.AssemblyReference = (...)
    prtSelect_cls = New CmpfcSelect
    prtSelect = prtSelect_cls.CreateModelItemSelection(...)
    constraint.ComponentReference = (...)
    Feature = Baugruppe.AssembleComponent(Teil, Nothing)
    Feature.SetConstraints(..., Nothing)
    Feature.Regenerate()
    window = session.CurrentWindow()
    window.Refresh()

    'durch diese Zuweisung erhält ASMKOS_OW alle Modelitem (Feature, Bezüge etc.) der Baugruppe
    'Aus ASMKOS_OW wird das Modelitem vom Typ KOS über den Namen gefiltert
    'vgl. Baugruppe
    'vgl. Baugruppe
    'erstellt ein Objekt, dass Eigenschaften Methoden für constraints bereitstellt
    'Sammel Array für Constraints
    'Array wird eine Referenzbedingung übergeben -> Koordinatensystem reicht eine Einbaureferenz
    'Identifiziert eine Komponente in Baugruppe
    'Klasse stellt Funktionen zur Erstellung eines Komponentenpfades zur Verfügung
    'Soll auf Unterbaugruppen oder vorhandene Teile referenziert werden muss der Komponentenpfad noch eingestellt werden
    'Klasse stellt Funktionen zur allgemeinen Selektion bereit
    'Als Baugruppenreferenz wird das Koordinatensystem der Baugruppe übergeben
    'vgl. Baugruppe
    'vgl. Baugruppe
    'vgl. Baugruppe
    'eine eingebaute Komponente stellt ein Baugruppenfeature dar
    'Einbaubedingungen werden gesetzt
    'regenerieren
    'aktuelles Fenster holen und Bildneuaufbau

End Sub
```

Druckbehälterberechnung

btnUDF

Private Sub btnUDF_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnUDF.Click

```
Dim model As IpfcModel
Dim window As IpfcWindow
Dim status As Boolean
status = (...) ' damit das UDF sinnvolle Werte erhält

If status = 0 Then
    Exit Sub
End If

model = Modell_Aus_Arbeitsverz(...)
createUDFInPart(...)

model.Regenerate(Nothing) 'regenerieren
model = Modell_Aus_Arbeitsverz(...)
model.Regenerate(Nothing) 'regenerieren
window = session. (...) 'aktuelles Fenster holen und Bildneuaufbau
window.Refresh()

End Sub
```

The image shows a CAD software interface with a 3D model of a cylinder. The model is oriented with FRONT, RIGHT, and TOP views. A red arrow points to the center of the cylinder, labeled 'ACHSE'. A dimension line is shown on the top surface, labeled 'DTM1'. Another dimension line is shown on the side of the cylinder, labeled 'A_29'. Two dialog boxes are open. The top dialog box, titled 'UDF-Platzierung', has tabs for 'Platzierung', 'Variablen', 'Optionen', 'Anpassungen', and 'Eigenschaften'. It shows 'Referenzen der ursprünglichen KEs' (1. AXIS, 2. SURFACE, 3. SURFACE) and 'Referenzen der UDFs' (A_1(Achse);F6(KÖRPER)). The bottom dialog box, also titled 'UDF-Platzierung', has a table with columns 'Besitzer', 'Quelle', 'Name', and 'Wert'. The table contains three rows: 'Bezugseb...' with 'Dimension' 'd4' and 'Wert' '30.0', 'Profil 1' with 'Dimension' 'd6' and 'Wert' '300.00', and 'Profil 1' with 'Dimension' 'd7' and 'Wert' '1500.00'. Below the table, there is a text area with 'd4 - Winkel', 'd6 - Durchmesser', and 'd7 - Abstand'. At the bottom of the dialog box, there is a checkbox for 'Autom Regenerierung' and a status bar showing 'KE-Nr: 2 von 2' and 'PROFIL 1 War erfolgreich'.

Besitzer	Quelle	Name	Wert
Bezugseb...	Dimension	d4	30.0
Profil 1	Dimension	d6	300.00
Profil 1	Dimension	d7	1500.00

d4 - Winkel
d6 - Durchmesser
d7 - Abstand

```
Function createUDFinPart(ByVal placementModel As IpfcSolid) As  
IpfcFeatureGroup
```

```
Dim csys As IpfcCoordSystem = Nothing  
Dim udfInstructions As IpfcUDFCustomCreateInstructions  
Dim group As IpfcFeatureGroup  
Dim references As CpfcUDFReferences  
Dim reference1 As IpfcUDFReference  
Dim reference2 As IpfcUDFReference  
Dim reference3 As IpfcUDFReference  
Dim ModelItemOW As IpfcModelItemOwner  
Dim ModelItem As IpfcModelItem  
Dim prtSelect As IpfcSelection  
Dim prtSelect_cls As CmpfcSelect  
Dim VariantDim As IpfcUDFVariantDimension  
Dim variantVals As IpfcUDFVariantValues
```

```
udfInstructions = (New CCpfcUDFCustomCreateInstructions).Create(...)  
'*** Einbaureferenzen setzen ***  
ModelItemOW = placementModel  
ModelItem = ModelItemOW.GetItemByName(...)  
prtSelect_cls = New CmpfcSelect  
prtSelect = prtSelect_cls.CreateModelItemSelection(ModelItem, Nothing)  
references = New CpfcUDFReferences()  
prtSelect.Highlight(0)  
reference1 = (New CCpfcUDFReference).Create(...)  
reference1.IsExternal = False  
references.Set(0, reference1)  
ModelItem = ModelItemOW.GetItemByName(...)  
prtSelect = prtSelect_cls.CreateModelItemSelection(ModelItem, Nothing)  
reference2 = (New CCpfcUDFReference).Create(...)  
reference2.IsExternal = False  
references.Set(...)  
ModelItem = ModelItemOW.GetItemByName(...)  
prtSelect = prtSelect_cls.CreateModelItemSelection(ModelItem, Nothing)  
reference3 = (New CCpfcUDFReference).Create(...)  
reference3.IsExternal = False  
references.Set(...)  
udfInstructions.References = references
```

```
'*** Bemaßungen anpassen ***  
VariantDim = (New CCpfcUDFVariantDimension).Create("d4", 30)  
'Winkel  
variantVals = New CpfcUDFVariantValues  
variantVals.Set(0, VariantDim)  
VariantDim = (New CCpfcUDFVariantDimension).Create("d6", Da / 10)  
'Durchmesser  
variantVals.Set(...)  
VariantDim = (New CCpfcUDFVariantDimension).Create("d7", L / 2)  
'Abstand  
variantVals.Set(...)  
udfInstructions.VariantValues = (...)  
group = placementModel.CreateUDFGroup(...)  
Return group  
End Function
```

Druckbehälterberechnung

btnEigenschBaugr

```
Private Sub btnEigenschBaugr_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEigenschBaugr.Click
```

```
    Dim solid As IpfcSolid
    Dim Features As IpfcFeatures
    Dim component As IpfcComponentFeat
    Dim modeldescriptor As IpfcModelDescriptor
    Dim model As IpfcModel
    Dim MassProp As IpfcMassProperty
    solid = Modell_Aus_Arbeitsverz("Behaelter.asm")
    Features = solid.ListFeaturesByType(False, (...))
    Vol = 0
    For i As Integer = 0 To Features.Count - 1
        component = Features(i)
        modeldescriptor = component.ModelDescr()
        model = session.GetModelFromDescr(modeldescriptor)
        Vol = Vol + (...)
```

```
Next
```

```
txtVolumen.Text = Round(Vol / 10 ^ 9, 2)
solid.Regenerate(Nothing) 'regenerieren
MassProp = solid(...)
m = MassProp.Mass * 10 ^ 3
txtMasse.Text = Round(m, 2)
```

```
End Sub
```

Unterfunktion ParamAusCreo

```
Function ParamAusCreo(ByVal model As IpfcModel, ByVal Parametername As String) As Double
```

```
    Dim wert As Double
    Dim ParaOwner As IpfcParameterOwner
    Dim para As IpfcParameter = Nothing
    Dim CValue As New CMpfcModelItem
    Dim Value As IpfcParamValue
    Value = CValue.CreateDoubleParamValue(wert)
    ParaOwner = model
    para = ParaOwner.GetParam(...)
    Value = (...)
    wert = (...)
    Return wert
```

```
End Function
```

Druckbehälterberechnung

Angepasstes Speichern und Öffnen

```
Private Sub WriteXML(ByVal savefiledialog1 As SaveFileDialog)
```

```
    Dim WriterXMLSet As New XmlWriterSettings()
    Dim WriterXML As XmlWriter
    Dim Wanddickenunterschr As Integer
    Dim BodenSeite1 As String = ""
    Dim BodenSeite2 As String = ""

    If ChkzuschlWanddicke.CheckState = CheckState.Checked Then
        Wanddickenunterschr = 0
    Else
        If txtWanddickenunterschr.Text = "" Then
            Wanddickenunterschr = 0
        Else
            Wanddickenunterschr = txtWanddickenunterschr.Text
        End If
    End If

    If rdioKugel1.Checked = True Then
        BodenSeite1 = "Kugel"
    ElseIf rdioKloeppe1.Checked = True Then
        BodenSeite1 = "Kloeppe"
    End If

    If rdioKugel2.Checked = True Then
        BodenSeite2 = "Kugel"
    ElseIf rdioKloeppe2.Checked = True Then
        BodenSeite2 = "Kloeppe"
    End If

    WriterXMLSet.Indent = True
    WriterXMLSet.IndentChars = "  "
    ...
end Sub
```

The screenshot displays the XML Notepad application window. The main window title is 'XML Notepad - C:\Users\sckesselma...\Desktop\test.xml'. The interface includes a menu bar (File, Edit, View, Insert, Window, Help) and a toolbar. The 'Tree View' on the left shows a hierarchical structure of XML elements:

- xml
 - Druckbehälter
 - Auftragsdaten
 - Bearbeiter
 - Auftragsnummer
 - Berechnungsdaten
 - AussenDurchmesser
 - Betriebsdruck
 - Werkstoff
 - Ausnutzungsfaktor
 - Wanddickenunterschr
 - Mantellänge
 - CADModell
 - Seite1
 - Seite2
 - Ergebnis
 - Wanddicke
 - Masse
 - Volumen

The 'XSL Output' pane on the right shows the XML data being processed:

```
version="1.0" encoding="utf-8"
Kesselma
k-007-2012
1500
10
S235JRG1
üblich
0
2500
Kugel
Kugel
5,77
849,27
6,08
```

A small dialog box titled 'Datei Bearbeiten' is overlaid on the bottom right, showing a list of data items:

```
Datei Bearbeiten
Format Ansicht ?
Kesselma
k-007-2012
1500
10
S235JRG1
üblich
0
2500
Kugel
Kugel
5,77
849,27
6,08
```