Fallstudie: Simulation eines Geldautomaten

Diese Fallstudie ist in Bezug auf folgende Punkte interessant:

- Es wird darin gezeigt, wie ein selbst programmierter Zufallszahlengenerator in Simulationen eingesetzt werden kann.
- Die Anwendung ist objektorientiert programmiert, enthält aber nur eine kleine Anzahl von Objekten bzw. Klassen und ist deshalb auch gut für Anfänger in der OOP geeignet.
- Der Aufbau genügt einer Schichtenarchitektur. Diese ist gut überschaubar, denn es handelt sich nur um zwei Schichten.
- Es wird demonstriert, wie Dialoge mit Hilfe von Zustandsdiagrammen geplant werden können und wie diese Planung dann in eine zweckmäßige Programmerung umgesetzt werden kann.
- Das UI enthält sowohl Eingaben über Formular als auch Eingaben und Ausgaben auf Tabellenblättern.

Die Simulationsanwendung besteht aus zwei Teilsimulationen

- Simulation von Abhebungsvorgängen. Der Schwerpunkt liegt dabei auf dem Umgang des Kunden mit dem Automaten („look and feel“). Wir wollen diesen Teil als Simulation A bezeichnen.
- Simulation der Dauerbenutzung zur Optimierung der Füllung in Abhängigkeit von der Verteilung der Auszahlungsbeträge, bei gegebener Auszahlungsstrategie. Diesen Teil nennen wir Simulation B

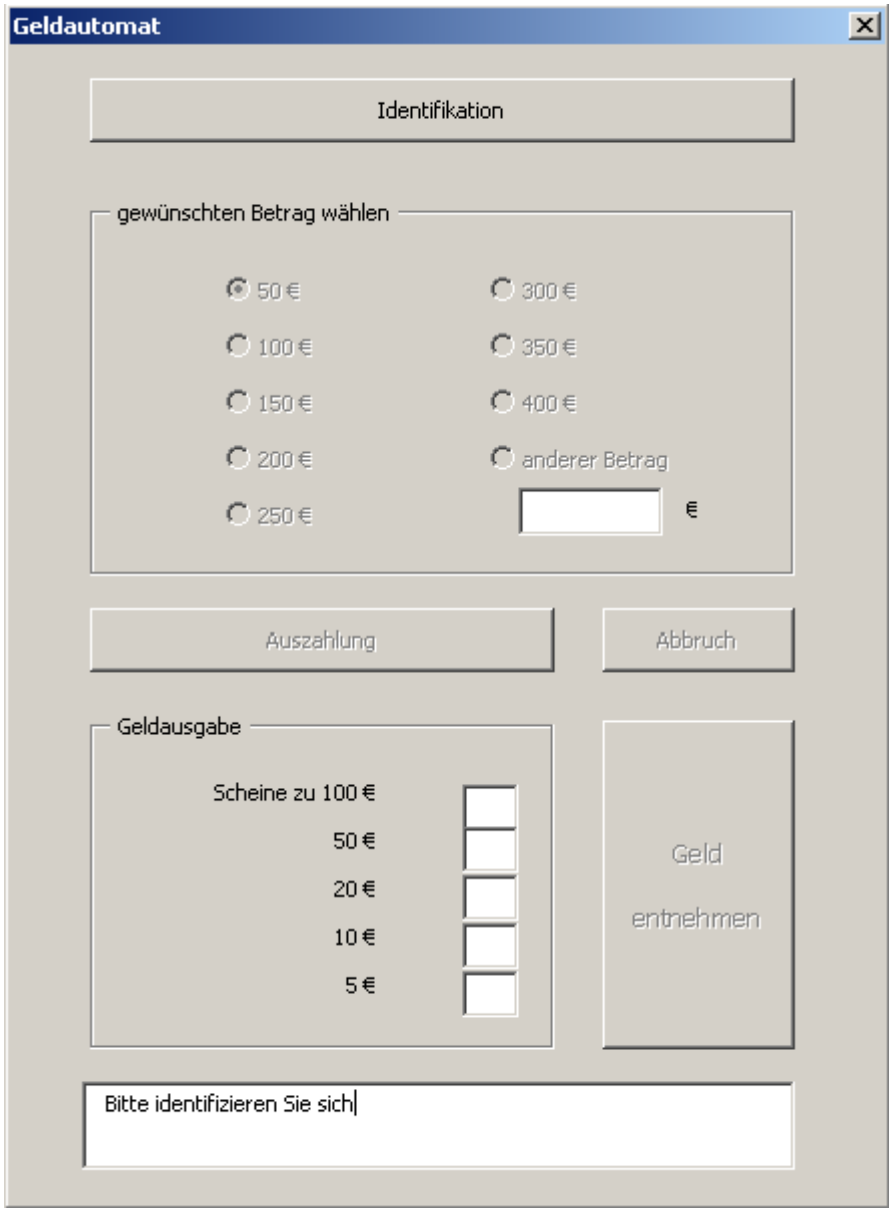
Die Auszahlungsstrategie des Automaten lautet „so groß wie möglich“. Es sollen also, abhängig vom gewünschten Betrag und dem aktuellen Bestand an Scheinen, jeweils so wenige Scheine wie möglich ausgegeben werden. Der Anfangsbestand an Scheinen wird in beiden Simulationen von einem Arbeitsblattbereich eingelesen, ist also beliebig veränderbar. Bei Simulation A geschieht dieses Einlesen über einen Vordialog, bei Simulation B ist das Einlesen im Hauptdialog enthalten. Der Unterschied macht Sinn, denn bei Simulation A ist der gedachte Benutzer der Bankkunde, bei Simulation B dagegen ein Angestellter der Bank.

Simulation A („look and feel“)

Ein Formular repräsentiert gegenüber dem Kunden das UI des Automaten. Das folgende Bild zeigt den Zustand des Formulars, bevor eine Abhebung beginnt. Zunächst ist der Automat im Bereitschaftszustand. Die Interaktion mit dem Bankkunden ist wie folgt:

- Die Identifikation des Benutzers, normalerweise mit Karte und PIN-Eingabe, wird in der Simulation der Einfachheit halber durch ein Anklicken der Schaltfläche <Identifikation> ersetzt.
- Nach dieser vereinfachten Identifizierung kann der Benutzer den gewünschten Betrag eingeben. Neben den explizit genannten Beträgen sind auch andere Beträge wählbar. Welcher Betrag maximal möglich ist kann innerhalb der Software vorgegeben werden. In jedem Fall muss ein vom Benutzer frei gewählter Betrag durch 5 ohne Rest teilbar sein, denn kleinere Scheine als 5€-Scheine hält der Automat nicht bereit.

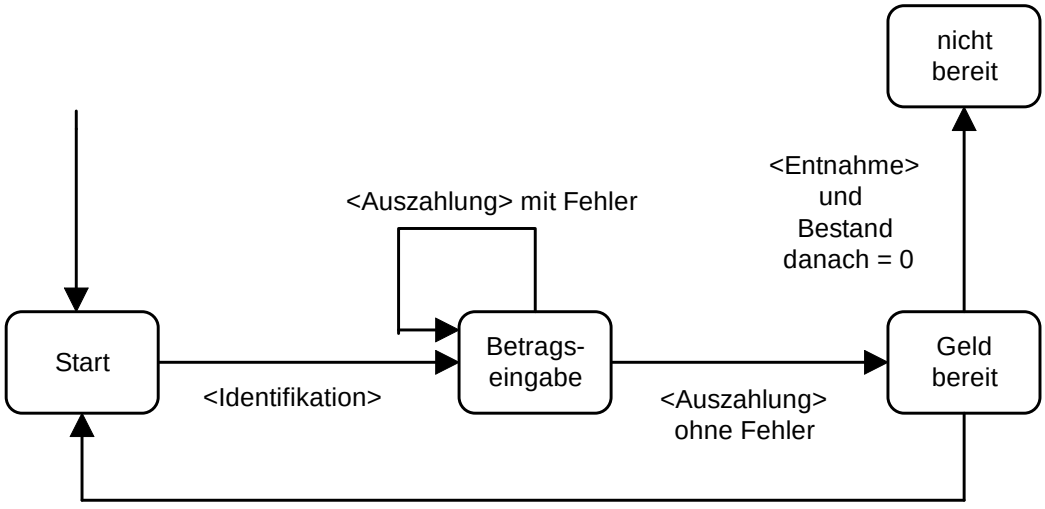
- So lange die Schaltfläche Auszahlung noch nicht gedrückt wurde, kann der Benutzer den ganzen Vorgang noch abbrechen. Drückt er die Schaltfläche <Abbruch>, so kehrt der Automat in den Bereitschaftszustand zurück. Weitere Benutzung des Automaten ist erst nach erneuter Identifikation möglich.
- Sobald der Kunde die Schaltfläche <Auszahlung> gedrückt hat, stellt der Automat die Scheine für den gewünschten Betrag zusammen. Anstatt tatsächlicher Scheine sieht der Kunde in der Simulation eine Aufzählung der Stückzahlen der verschiedenen Scheine.
- Der Kunde „entnimmt“ die Scheine, indem er die Schaltfläche <Geld entnehmen> drückt. Danach steht der Automat im Nomalfall für die nächste Abhebung bereit. Das Formular wird hierfür nicht neu gestartet.
- In Ausnahmefällen kann es dazu kommen, dass ein vom Benutzer gewünschter Betrag nicht ausgegeben werden kann, obwohl der Betrag den Maximalbetrag nicht übersteigt und im Automaten noch Geld ist. Dies ist der Fall, wenn der Geldbestand kleiner als der gewünschte Betrag ist, oder wenn der Betrag mit den vorhandenen Scheinen nicht realisiert werden kann. Beispiel: gewünschter Betrag 115€ bei einem Bestand von zweihundert-Euro-Scheinen.
- Falls nach einer Auszahlung der Bestand des Automaten den Wert 0 erreicht hat, nimmt der Automat den Zustand „nicht bereit“ an. Es ist dann auch keine Identifikation mehr möglich.



Planung des Dialogs und Vorbereitung der Dialogprogrammierung

Bei einem Dialog wie dem oben beschriebenen ist es wichtig, den Benutzer eng zu führen, damit er keine Fehler machen kann. Der Benutzer soll stets wissen, was von ihm erwartet wird, abweichende Aktionen sollen gar nicht möglich sein. Eine solche Dialogführung lässt sich gut mit Zustandsdiagrammen (endlichen Automaten) planen und vorbereiten (s. folgendes Bild).

Die Kästchen mit den abgerundeten Ecken repräsentieren die Dialogzustände. Der oben beschriebene Bereitschaftszustand ist im Diagramm mit „Start“ bezeichnet. Übergänge von einem Zustand zu einen anderen werden durch das Anklicken von Schaltflächen veranlasst; in manchen Fällen ist jedoch eine zusätzliche Bedingung für einen bestimmten Zustandsübergang notwendig. Wird z.B. im Zustand „Betragseingabe“ die Schaltfläche <Auszahlung> gedrückt, so ist der Folgezustand davon abhängig, ob bei der Verarbeitung des eingegebenen Betrags ein Fehler aufgetreten ist oder nicht.



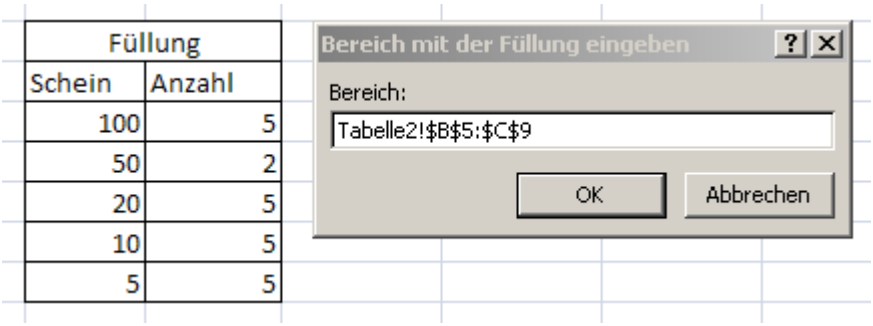
Die einzelnen Zustände werden, von einigen Feinheiten abgesehen, durch die folgende Tabelle definiert. Bei Eintrag A (für aktiviert) sind die betreffenden Felder eingabebereit, bei Eintrag N nicht.

Steuerelement	Start	Betrags-eingabe	Geld bereit	nicht bereit
<Identifikation>	A	N	N	N
Betragseingabe	N	A	N	N
<Auszahlung>	N	A	N	N
<Abbruch>	N	A	N	N
Geldausgabe	leer	leer	angezeigt	leer
<Geld entnehmen>	N	N	A	N

In der Tabelle nicht enthalten, aber ebenfalls relevant, ist der jeweilige Inhalt der Nachrichtenzeile am Fuß des Formulars. Außerdem sind einige Feinheiten ausgespart. So ist z.B. im Zustand „Betrags-eingabe“ das Feld für die Eingabe eines anderen Betrags normalerweise nicht aktiviert. Nur wenn der Benutzer den mit „anderer Betrag“ beschrifteten OptionButton markiert hat, ist das Feld eingabe-bereit.

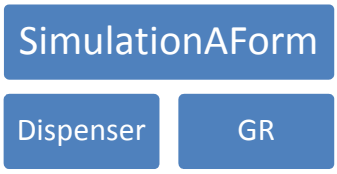
Hilfsdialog zur Eingabe der Füllung

Bevor die Simulationsmaske angezeigt wird, wird noch ein Hilfsdialog zur Eingabe der Füllung des Automaten mit Scheinen durchgeführt. Hierfür wird beim Initialisieren der Maske ein Standarddialog mit Hilfe einer Excel-InputBox gestartet (Bild). Die gewünschte, nach Scheingrößen spezifizierte Füllung muss vorher auf einem Tabellenblatt eingegeben worden sein:



Die Architektur von Simulation A

Die Architektur ist zweischichtig (s. Bild unten). Die obere Schicht beinhaltet das UI und besteht aus der Formularklasse SimulationAForm. In der darunter liegenden zweiten Schicht befindet sich die Klasse Dispenser, welche den Kern des Geldautomaten repräsentiert. Programmiertechnisch ist dieser Aufbau realisiert, indem in die Formularklasse eine Instanzenvariable der Klasse Dispenser aufgenommen wurde. Das ebenfalls in der zweiten Schicht enthaltene Modul GR enthält lediglich zwei Hilfsroutinen, von denen in SimulationAForm eine in Anspruch genommen wird: RangeToLong-Matrix verwandelt den im Hilfsdialog eingelesenen Bereich mit der Füllung in ein Array.



Die Klasse Dispenser

Diese Klasse, die den Kern des Automaten repräsentiert, wird sowohl in Simulation A als auch in Simulation B benötigt. Sie enthält deshalb mehr Methoden, als in Simulation A benutzt werden. Der folgende Kasten gibt zunächst einen Überblick über die Instanzenvariablen (oben) und die Methoden (unten). Properties wurden nicht benutzt. Die Werte der Instanzenvariablen müssen von außen mit Hilfe der Informationsmethoden Bestand (Gesamtbestand in €) und Scheinbestand (Anzahl der Scheinarten) abgefragt werden. Die Methode moeglicherBetrag vergleicht den Maximalbetrag mit dem Restbestand und liefert den kleineren Wert von beiden.

```
Scheine100
Scheine50
Scheine20
Scheine10
Scheine5
Maxbetrag
---------------------
Bestand
Scheinbestand
moeglicherBetrag
fuellen
Entnahme
```

Bis auf fuellen sind alle Methoden Funktionen. Die zentrale und mächtigste davon ist die Funktion Entnahme. Sie nimmt den gewünschten Betrag als Parameter entgegen und liefert ein Array mit den Scheinzahlen der Auszahlung. Außerdem schreibt sie die Scheinbestände fort. Im Fehlerfall (z.B. Betrag nicht mehr verfügbar) liefert die Methode das Array mit dem Wert -1 sowohl für die Indexuntergrenze als auch die Indexobergrenze, als Fehlersignal für die aufrufende Stelle. Hier ist der Code der Klasse:

```
'Bestände, in Anzahl der Scheine
Private Scheine100 As Long
Private Scheine50 As Long
Private Scheine20 As Long
Private Scheine10 As Long
Private Scheine5 As Long

Private MAXBETRAG As Integer  'höchstens auszahlbarer Betrag

'aktueller Geldbestand, in €
Public Function bestand() As Long
    bestand = Scheine100 * 100 + Scheine50 * 50 + Scheine20 * 20 + _
    Scheine10 * 10 + Scheine5 * 5
End Function


'aktueller Scheinbestand, in Stück
Public Function scheinbestand() As Long()
    Dim b(1 To 5) As Long
    b(1) = Scheine100
    b(2) = Scheine50
    b(3) = Scheine20
    b(4) = Scheine10
    b(5) = Scheine5
    scheinbestand = b
End Function


'liefert die Anzahl der auszugebenden Scheine und schreibt die
'Scheinbestände fort
Public Function entnahme(ByVal betrag As Integer) As Integer()
    Dim b() As Integer
    If Me.bestand < betrag Then                'Bestand reicht nicht
        ReDim b(-1 To -1)
    Else                                       'Bestand reicht
        ReDim b(1 To 5)
        b(1) = IIf(betrag \ 100 > Scheine100, Scheine100, betrag \ 100)
        betrag = betrag - b(1) * 100
        b(2) = IIf(betrag \ 50 > Scheine50, Scheine50, betrag \ 50)
        betrag = betrag - b(2) * 50
        b(3) = IIf(betrag \ 20 > Scheine20, Scheine20, betrag \ 20)
        betrag = betrag - b(3) * 20
        b(4) = IIf(betrag \ 10 > Scheine10, Scheine10, betrag \ 10)
        betrag = betrag - b(4) * 10
        b(5) = betrag \ 5
        If b(5) > Scheine5 Or betrag Mod 5 <> 0 Then  'nicht realisierbar
            ReDim b(-1 To -1)
        Else                    'Entnahme möglich, Beständeanpassung
            Scheine100 = Scheine100 - b(1)
            Scheine50 = Scheine50 - b(2)
            Scheine20 = Scheine20 - b(3)
            Scheine10 = Scheine10 - b(4)
```

```
            Scheine5 = Scheine5 - b(5)
        End If
    End If
    entnahme = b
End Function


'stattet den Automaten mit Geldscheinen aus, setzt Maximalbetrag
'-------------------------------------------------------------
Public Sub fuellen(ByVal s100 As Long, ByVal s50 As Long, _
                   ByVal s20 As Long, ByVal s10 As Long, _
                   ByVal s5 As Long, ByVal maxBetr As Integer)
   Scheine100 = s100
   Scheine50 = s50
   Scheine20 = s20
   Scheine10 = s10
   Scheine5 = s5
   MAXBETRAG = maxBetr
End Sub


'errechnet den aktuell möglichen höchsten Auszahlungsbetrag
(bestandsabhängig)
'-------------------------------------------------------------------------
--
Public Function moeglicherBetrag() As Long
    moeglicherBetrag = IIf(Me.bestand() > MAXBETRAG, MAXBETRAG, Me.bestand)
End Function
```

Die Formularklasse SimulationAForm

Der Code dieser Klasse hat eine beträchtliche Länge. Dies liegt vor allem an den Zustandsprozeduren am Ende. Jede dieser Prozeduren realisiert einen der Zustände aus dem Zustandsübergangsdiagramm (s. oben).

```
Private disp As dispenser    'Objekt, das den Automaten repräsentiert

'Aktionen nach Anklicken des Buttons Abbruch
Private Sub AbbruchBtn_Click()
    ZustandStart
End Sub


'vertritt die Identifikation durch Karte und PIN
'----------------------------------------------
Private Sub IdentBtn_Click()
    ZustandBetragseingabe ""
End Sub


'Aktionen nach Anklicken des Buttons Auszahlung
'---------------------------------------------
Private Sub AuszahlungBtn_Click()
    Dim b As Long                   'für den Betrag
    Dim a() As Integer              'für die Scheinezahlen
    Dim mStr As String              'Zusatzmitteilung
    mStr = " "
    If Me.Betr50OBtn.Value = True Then
        b = 50
```

```
    ElseIf Me.Betr100OBtn.Value = True Then
        b = 100
    ElseIf Me.Betr150OBtn.Value = True Then
        b = 150
    ElseIf Me.Betr200OBtn.Value = True Then
        b = 200
    ElseIf Me.Betr250OBtn.Value = True Then
        b = 250
    ElseIf Me.Betr300OBtn.Value = True Then
        b = 300
    ElseIf Me.Betr350OBtn.Value = True Then
        b = 350
    ElseIf Me.Betr400OBtn.Value = True Then
        b = 400
    Else       'Überprüfung des Betrags und Verwandlung in Ganzzahl
      If Not BetragOK(Me.andBetrTBx.Value) Then
         mStr = mStr & "Auszahlung nicht möglich. Wählen Sie einen " & _
        "durch 5 " & "teilbaren Betrag bis " & disp.moeglicherBetrag
        ZustandBetragseingabe mStr
        Exit Sub
      End If
        b = CLng(Me.andBetrTBx.Value)
    End If
    If b > disp.moeglicherBetrag Then
        mStr = mStr & "Der eingegebene Betrag ist zu hoch"
        ZustandBetragseingabe mStr
        Exit Sub
    End If

    a = disp.entnahme(b)    'Stückelung abrufen
    If LBound(a) = -1 Then  'falls Fehler aufgetreten
        mStr = mStr & "Auszahlung dieses Betrags nicht möglich. " & _
                      "Wählen Sie einen gerundeten Betrag"
        ZustandBetragseingabe mStr
        Exit Sub
    End If

    'Kein Fehler. Die "Auszahlung" wird vorgenommen
    Me.anz100TBx.Value = CStr(a(1))
    Me.anz50TBx.Value = CStr(a(2))
    Me.anz20TBx.Value = CStr(a(3))
    Me.anz10TBx.Value = CStr(a(4))
    Me.anz5TBx.Value = CStr(a(5))

    ZustandGeldBereit
End Sub


'Aktionen nach Klicken des Entnahme-Buttons (=Geldentnahme)
'---------------------------------------------------------
Private Sub EntnahmeBtn_Click()
    If disp.bestand > 0 Then ZustandStart Else ZustandNichtBereit
End Sub


'regelt die Eingabe eines "anderen" Betrags
'-----------------------------------------
Private Sub AndBetrOBtn_Change()
    If Me.AndBetrOBtn.Value = True Then
        Me.andBetrTBx.Enabled = True
        Me.andBetrTBx.SetFocus
```

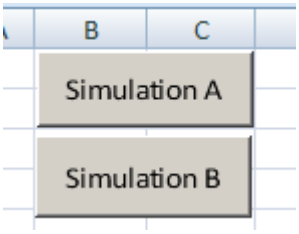
```
    Else
        Me.andBetrTBx.Value = ""
        Me.andBetrTBx.Enabled = False
    End If
End Sub


'Aktionen beim Öffnen des Formulars
'----------------------------------
Private Sub UserForm_Initialize()
    Me.EntnahmeBtn.Caption = "Geld" & vbLf & vbLf & "entnehmen"
    Me.EntnahmeBtn.Font.Size = 10
    Set disp = New dispenser
    On Error GoTo Fehlerbehandlung 'falls Fehler bei Füllungseingabe
    Dim frng As Range              'für die Füllung
    Dim r() As Long                'für die Füllung
    Set frng = Application.InputBox( _
            prompt:="Bereich: ", _
            Title:="Bereich mit der Füllung eingeben", _
            Type:=8)
    r = GR.RangeToLongMatrix(frng)
    disp.fuellen r(1, 2), r(2, 2), r(3, 2), r(4, 2), r(5, 2), 600
    ZustandStart
    Exit Sub

Fehlerbehandlung:
    ZustandNichtBereit
End Sub


'prüft einen eingegebenen "anderen" Betrag
'-----------------------------------------
Private Function BetragOK(ByVal eingabe As String) As Boolean
    If Not IsNumeric(eingabe) Then
        BetragOK = False
    Else
        Dim ez As Double
        ez = CDbl(eingabe)
        If ez > 0 And ez = Int(ez) And Int(ez) Mod 5 = 0 Then
            BetragOK = True
        Else
            BetragOK = False
        End If
    End If
End Function


'Zustandsprozeduren
'------------------------------------------------------------------------------

Private Sub ZustandStart()
    Me.IdentBtn.Enabled = True

    Me.anz100TBx.Enabled = False
    Me.anz10TBx.Enabled = False
    Me.anz10TBx.Enabled = False
    Me.anz20TBx.Enabled = False
    Me.anz50TBx.Enabled = False
    Me.anz5TBx.Enabled = False

    Me.AuszahlungBtn.Enabled = False
```

```
    Me.EntnahmeBtn.Enabled = False
    Me.MssgTBx.Enabled = True
    Me.AbbruchBtn.Enabled = False

    Me.Betr50OBtn.Value = True
    Me.Betr100OBtn.Value = False
    Me.Betr150OBtn.Value = False
    Me.Betr200OBtn.Value = False
    Me.Betr250OBtn.Value = False
    Me.Betr300OBtn.Value = False
    Me.Betr350OBtn.Value = False
    Me.Betr400OBtn.Value = False
    Me.AndBetrOBtn.Value = False
    Me.andBetrTBx.Value = ""

    Me.anz100TBx.Value = ""
    Me.anz10TBx.Value = ""
    Me.anz20TBx.Value = ""
    Me.anz50TBx.Value = ""
    Me.anz5TBx.Value = ""

    Me.Betr50OBtn.Enabled = False
    Me.Betr100OBtn.Enabled = False
    Me.Betr150OBtn.Enabled = False
    Me.Betr200OBtn.Enabled = False
    Me.Betr250OBtn.Enabled = False
    Me.Betr300OBtn.Enabled = False
    Me.Betr350OBtn.Enabled = False
    Me.Betr400OBtn.Enabled = False
    Me.AndBetrOBtn.Enabled = False
    Me.andBetrTBx.Enabled = False

    Me.MssgTBx.Text = "Bitte identifizieren Sie sich"
End Sub


Private Sub ZustandBetragseingabe(ByVal msg As String)
    Me.IdentBtn.Enabled = False

    Me.anz100TBx.Enabled = False
    Me.anz10TBx.Enabled = False
    Me.anz10TBx.Enabled = False
    Me.anz20TBx.Enabled = False
    Me.anz50TBx.Enabled = False
    Me.anz5TBx.Enabled = False

    Me.AuszahlungBtn.Enabled = True
    Me.EntnahmeBtn.Enabled = False
    Me.MssgTBx.Enabled = True
    Me.AbbruchBtn.Enabled = True

    Me.Betr50OBtn.Enabled = True
    Me.Betr100OBtn.Enabled = True
    Me.Betr150OBtn.Enabled = True
    Me.Betr200OBtn.Enabled = True
    Me.Betr250OBtn.Enabled = True
    Me.Betr300OBtn.Enabled = True
    Me.Betr350OBtn.Enabled = True
    Me.Betr400OBtn.Enabled = True
    Me.AndBetrOBtn.Enabled = True
    Me.andBetrTBx.Enabled = False
```

```
    Me.Betr50OBtn.Value = True

    Me.MssgTBx.Text = "aktueller Höchstbetrag: " & disp.moeglicherBetrag() _
                      & " €. " & msg
    End Sub


Private Sub ZustandGeldBereit()
    Me.IdentBtn.Enabled = False

    Me.anz100TBx.Enabled = False
    Me.anz10TBx.Enabled = False
    Me.anz10TBx.Enabled = False
    Me.anz20TBx.Enabled = False
    Me.anz50TBx.Enabled = False
    Me.anz5TBx.Enabled = False

    Me.AuszahlungBtn.Enabled = False
    Me.EntnahmeBtn.Enabled = True
    Me.MssgTBx.Enabled = True
    Me.AbbruchBtn.Enabled = False

    Me.Betr50OBtn.Enabled = False
    Me.Betr100OBtn.Enabled = False
    Me.Betr150OBtn.Enabled = False
    Me.Betr200OBtn.Enabled = False
    Me.Betr250OBtn.Enabled = False
    Me.Betr300OBtn.Enabled = False
    Me.Betr350OBtn.Enabled = False
    Me.Betr400OBtn.Enabled = False
    Me.AndBetrOBtn.Enabled = False
    Me.andBetrTBx.Enabled = False
    Me.Betr50OBtn.Value = False

    Me.MssgTBx.Text = "Bitte Geld entnehmen"

End Sub


Private Sub ZustandNichtBereit()
    Me.IdentBtn.Enabled = False

    Me.anz100TBx.Value = ""
    Me.anz10TBx.Value = ""
    Me.anz20TBx.Value = ""
    Me.anz50TBx.Value = ""
    Me.anz5TBx.Value = ""

    Me.anz100TBx.Enabled = False
    Me.anz10TBx.Enabled = False
    Me.anz10TBx.Enabled = False
    Me.anz20TBx.Enabled = False
    Me.anz50TBx.Enabled = False
    Me.anz5TBx.Enabled = False

    Me.AuszahlungBtn.Enabled = False
    Me.EntnahmeBtn.Enabled = False
    Me.MssgTBx.Enabled = True
    Me.AbbruchBtn.Enabled = False

    Me.Betr50OBtn.Value = True
```

```
    Me.Betr100OBtn.Value = False
    Me.Betr150OBtn.Value = False
    Me.Betr200OBtn.Value = False
    Me.Betr250OBtn.Value = False
    Me.Betr300OBtn.Value = False
    Me.Betr350OBtn.Value = False
    Me.Betr400OBtn.Value = False
    Me.AndBetrOBtn.Value = False
    Me.andBetrTBx.Value = ""

    Me.Betr50OBtn.Enabled = False
    Me.Betr100OBtn.Enabled = False
    Me.Betr150OBtn.Enabled = False
    Me.Betr200OBtn.Enabled = False
    Me.Betr250OBtn.Enabled = False
    Me.Betr300OBtn.Enabled = False
    Me.Betr350OBtn.Enabled = False
    Me.Betr400OBtn.Enabled = False
    Me.AndBetrOBtn.Enabled = False
    Me.andBetrTBx.Enabled = False
    Me.Betr50OBtn.Value = False

    Me.MssgTBx.Text = "Automat nicht bereit." & vbLf & _
                      "Bitte benutzen Sie den Automat einer anderen Filiale"
End Sub
```

Es bleibt noch zu klären, wie das Formular für Simulation A gestartet wird. Die einfachste Lösung ist, hierfür eine Schaltfläche in ein Arbeitsblatt aufzunehmen (s. Bild). Klickt der Benutzer auf diese Schaltfläche, so wird eine Prozedur im Modul des Tabellenblatts gestartet, welche ein Formular erzeugt und öffnet.




```
Private Sub CommandButton1_Click()
    Dim fa As SimulationAForm
    Set fa = New SimulationAForm
    fa.Show
End Sub
```

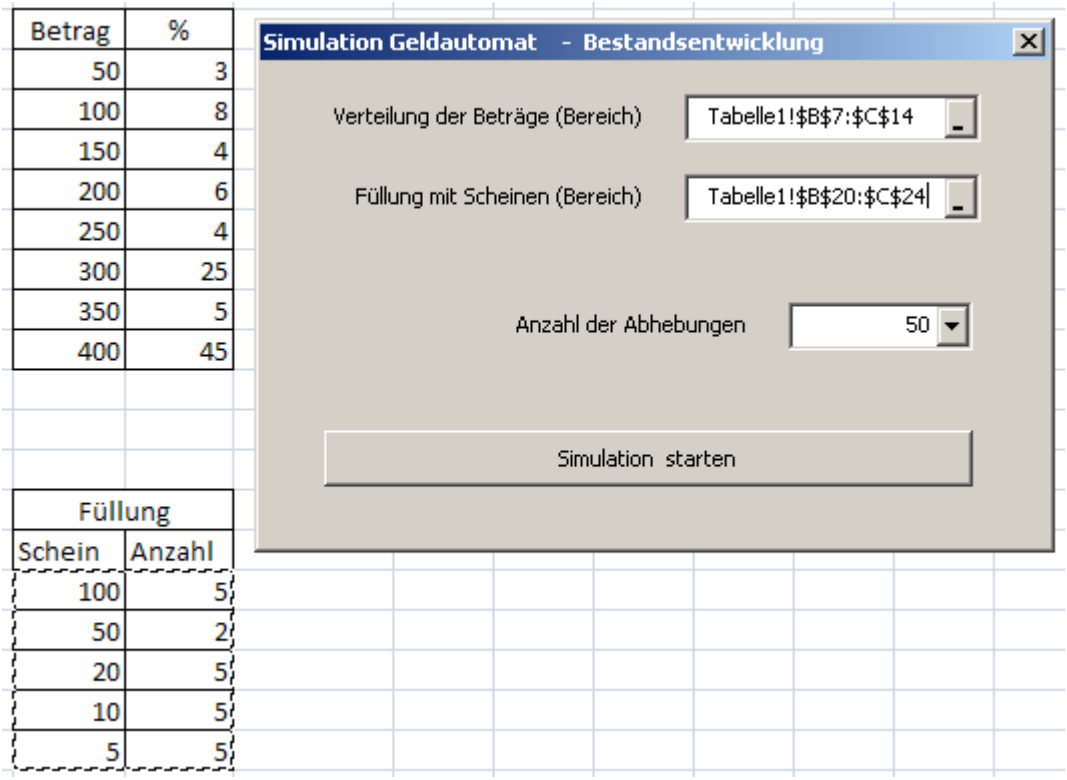
Eine entsprechende Lösung wurde für Simulation B realisiert, die im Folgenden beschrieben wird.

Simulation B (Simulation der Dauerbenutzung zur Optimierung der Füllung)

Diese Simulation führt eine Serie von Abhebungen durch und protokolliert dabei auf einem Tabellenblatt für jede Abhebung den Betrag, die ausgegebenen Scheine und die Restbestände an Scheinen

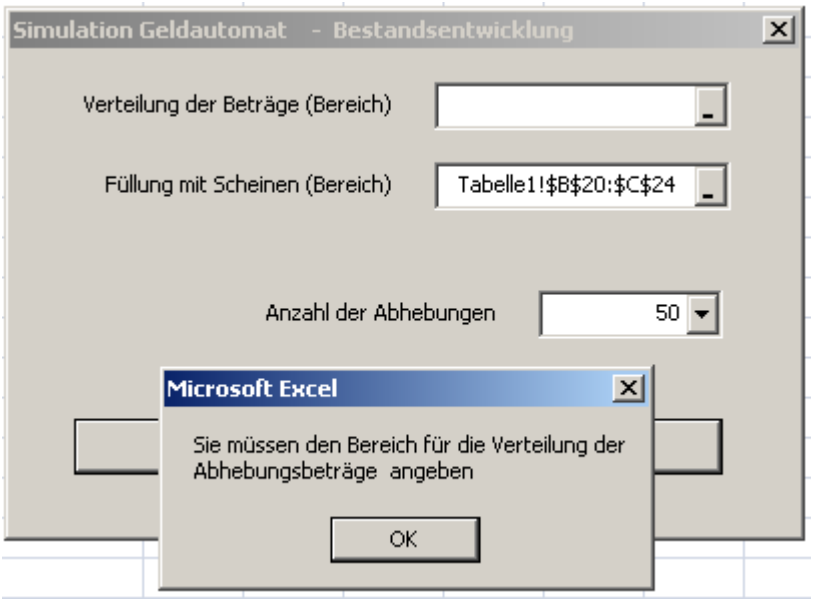
nach der Ausgabe. Das UI aus Simulation A wird hierfür nicht benötigt, wohl aber die Klasse Dispenser, die den Automaten repräsentiert.

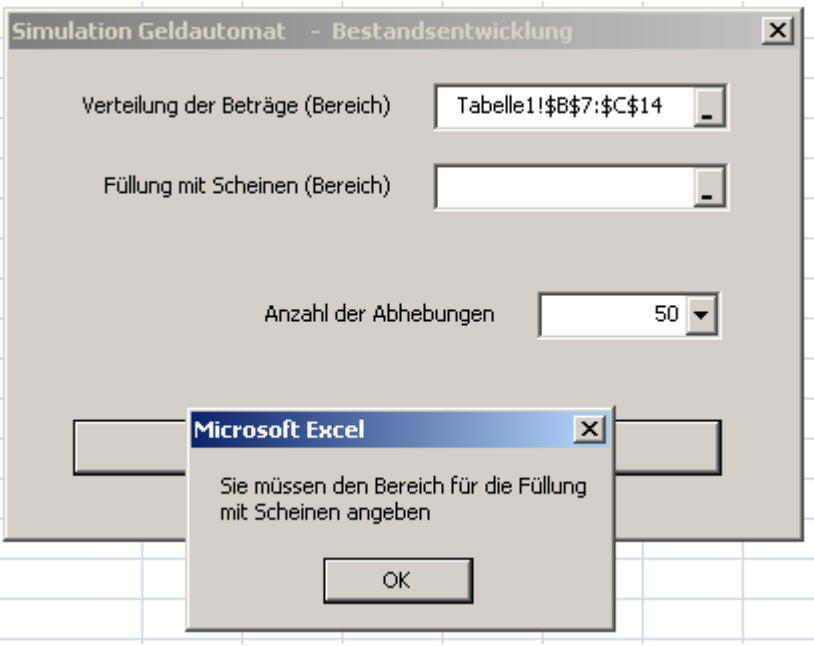
Die Simulation wird mit Hilfe eines Formulars angestoßen:



Die Häufigkeitsverteilung der möglichen Beträge muss bereits vorher auf dem Tabellenblatt eingegeben worden sein, desgleichen die Anfangsausstattung des Automaten mit Scheinen. In das Formular werden diese Daten mit Hilfe des Steuerelements RefEdit übernommen. Die Anzahl der Abhebungen wird mit Hilfe einer ComboBox eingelesen, welche die Werte 10, 50, 100, 200, 300, 500 und 1000 erlaubt.

Fehler bei der Eingabe der Verteilung und der Scheinausstattung werden abgefangen. Im Fehlerfall wird eine Meldung ausgegeben. Nachdem diese Meldung weggeklickt ist, befindet sich der Cursor in dem Eingabefeld, wo die Eingabe fehlt (s. Bilder unten).





Nach dem Starten der Simulation werden nacheinander die Abhebungen in der gewünschten Anzahl durchgeführt und auf einem Tabellenblatt protokolliert. Die Abhebungsbeträge werden dabei mit Hilfe eines Zufallszahlengenerators gefunden, der diese Beträge gemäß der eingegebenen Häufigkeitsverteilung generiert.

E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
Betrag		A100	A50	A20	A10	A5		R100	R50	R20	R10	R5		Bestand
250		2	1	0	0	0		48	49	100	200	400		13250
400		4	0	0	0	0		44	49	100	200	400		12850
400		4	0	0	0	0		40	49	100	200	400		12450
300		3	0	0	0	0		37	49	100	200	400		12150
400		4	0	0	0	0		33	49	100	200	400		11750
400		4	0	0	0	0		29	49	100	200	400		11350
100		1	0	0	0	0		28	49	100	200	400		11250
400		4	0	0	0	0		24	49	100	200	400		10850
300		3	0	0	0	0		21	49	100	200	400		10550
200		2	0	0	0	0		19	49	100	200	400		10350

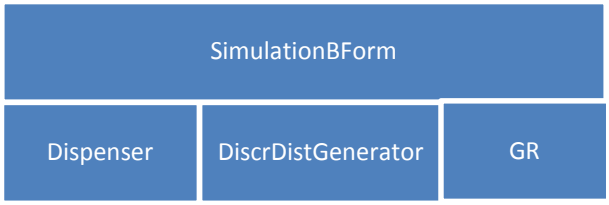
Die erste Spalte der Ausgabe zeigt den vom Kunden angeforderten Betrag. Alle hier aufgelisteten Beträge sind grundsätzlich zulässige Beträge. Es kann allerdings vorkommen, dass auch ein zulässiger Betrag nicht ausgezahlt werden kann, entweder weil der Bestand des Geldautomaten nicht ausreicht oder weil nicht die Scheine vorhanden sind, um den Betrag zusammen zu setzen.

Die mit A100 bis A5 überschriebenen Spalten enthalten die Anzahl der ausgegebenen Scheine. Die Spalten mit den Überschriften R100 bis R5 zeigen den Restbestand an Scheinen nach der Auszahlung bzw. der Ablehnung des Auszahlungswunsches. Die Spalte Bestand zeigt den gesamten Restbestand in €.

Das Protokoll zeigt normalerweise so viele Benutzungsfälle, wie im Formular angefordert wurden. Falls aber bereits vorher der Bestand 0 erreicht ist, werden Simulation und Protokoll beendet.

Die Architektur von Simulation B

Auch hier handelt es sich um eine zweischichtige Architektur. Neben der Klasse Dispenser und dem Modul GR, die bereits bei Simulation A zum Einsatz kamen, ist hier in der zweiten Schicht noch die Klasse DiscrDistGenerator enthalten. Hierbei handelt es sich um den Zufallszahlengenerator, der die Beträge für die Abhebungen erzeugen soll.



Da die Klasse Dispenser bereits aus Simulation A bekannt ist und die Klasse DiscrDistGenerator an anderer Stelle ausführlich beschrieben ist, betrachten wir hier nur noch die Formularklasse **SimulationBForm** im Detail.

Beachten Sie im folgenden Code die Validierung der Eingaben in die beiden RefEdit-Steuerelemente in der Methode GenerierenBtn_Click. Als Eingaben werden hier Zeichenfolgen erwartet, welche Bereiche ausdrücken sollen. Es wird auf eine inhaltliche Analyse der Eingaben verzichtet, weil dies sehr aufwändig wäre. Stattdessen wird die klassische Art der VBA-Fehlerbehandlung mit On Error GoTo angewandt. Im Fehlerfall, wenn eine der Eingaben nicht verwertbar ist, wird zu der zuständigen

Fehlerbehandlung am Ende der Methode gesprungen. Dem Benutzer wird eine Fehlermeldung gezeigt. Anschließend hat er die Möglichkeit, seine Eingaben zu korrigieren.

```
Private gen As DiscrDistGenerator   'Zufallszahlengenerator
Private disp As dispenser           'der Geldautomat

'Aktionen beim Öffnen des Formulars
Private Sub UserForm_Initialize()
    Set gen = New DiscrDistGenerator
    Set disp = New dispenser
    Me.AnzahlCBx.AddItem ("10")
    Me.AnzahlCBx.AddItem ("50")
    Me.AnzahlCBx.AddItem ("100")
    Me.AnzahlCBx.AddItem ("200")
    Me.AnzahlCBx.AddItem ("300")
    Me.AnzahlCBx.AddItem ("400")
    Me.AnzahlCBx.AddItem ("500")
    Me.AnzahlCBx.AddItem ("1000")
    Me.AnzahlCBx.Value = "50"
    Me.DistrREd.Text = ActiveWindow.RangeSelection.Address
End Sub


'Aktionen nach Anklicken des Buttons <Generieren>
Private Sub GenerierenBtn_Click()
    Dim vrng As Range              'für die Verteilung
    Dim frng As Range              'für die Füllung
    Dim arng As Range              'für die Ausgabe
    Dim v() As Double              'für die Verteilung
    Dim betrag As Integer
    Dim a() As Integer             'für die auszugebenden Scheine
    Dim r() As Long                'für den Restbestand an Scheinen
    Dim i As Integer, j As Integer, n As Integer, h As Integer

    'dem Generator die Verteilung übergeben
    On Error GoTo Fehlerbehandlung1
    Set vrng = Range(Me.DistrREd.Text)
    v = GR.RangeToDoubleMatrix(vrng)
    gen.setDist v

    'dem Automaten Füllung und Maxbetrag bekanntgeben
    On Error GoTo Fehlerbehandlung2
    Set frng = Range(Me.fillREd.Text)
    r = GR.RangeToLongMatrix(frng)
    disp.fuellen r(1, 2), r(2, 2), r(3, 2), r(4, 2), r(5, 2), 600

    'die Protokolltabelle ausgeben
    n = CInt(Me.AnzahlCBx.Text)   'Anzahl der Abhebungen
    Set arng = Worksheets("Tabelle1").Range("E1:S" & (n + 1))
    Worksheets("Tabelle1").Range("E1:S1001").Cells.Clear
    arng.ColumnWidth = 6.45
    arng(1, 1).Value = "Betrag"     'Überschriften
    arng(1, 3).Value = "A100"
    arng(1, 4).Value = "A50"
    arng(1, 5).Value = "A20"
    arng(1, 6).Value = "A10"
    arng(1, 7).Value = "A5"
    arng(1, 9).Value = "R100"
    arng(1, 10).Value = "R50"
    arng(1, 11).Value = "R20"
```

```
    arng(1, 12).Value = "R10"
    arng(1, 13).Value = "R5"
    arng(1, 15).Value = "Bestand"
    For i = 1 To n                  'die einzelnen Abhebungen
        betrag = gen.rDist(Rnd)
        arng(i + 1, 1) = betrag
        a = disp.entnahme(betrag)
        If UBound(a) <> -1 Then     'wenn Auszahlung möglich
            For j = 1 To 5
                arng(i + 1, j + 2) = a(j)
            Next j
        End If
        r = disp.scheinbestand
        For j = 1 To 5
            arng(i + 1, j + 8) = r(j)
        Next j
        arng(i + 1, 15) = disp.bestand
        If disp.bestand = 0 Then Exit For
    Next i
    Exit Sub

Fehlerbehandlung1:
    MsgBox "Sie müssen den Bereich für die Verteilung der " & vbLf & _
           "Abhebungsbeträge  angeben"
    Me.DistrREd.SetFocus
    Exit Sub

Fehlerbehandlung2:
    MsgBox "Sie müssen den Bereich für die Füllung " & vbLf & _
           "mit Scheinen angeben"
    Me.fillREd.SetFocus
    Exit Sub
End Sub
```