

Überlegungen zum Übungsblatt „eBanking“

1 Erste Projektphase

Eine empfehlenswerte Möglichkeit, die Bearbeitung eines Sw-Projektes zu konkretisieren, ist die **Betrachtung aus der Sicht der Nutzenden**:

1.1 Die OK-Schaltfläche:

Haupteffekt der Nutzung ist die Berücksichtigung der Ein- und Auszahlungsbeträge und die Berechnung des neuen Kontostandes:

```
KtoStd = KtoStd + txtEinEu.Text - txtAusEu.Text
```

(alle Beträge sollten sicherheitshalber mit `numChk (... .Text)` eingetragen sein.)

Voraussetzung für die o.a. Summenbildung ist, daß das Konto nicht überzogen wird:

```
If txtAusEu.Text <= KtoStd Then (...) End If
```

(Man beachte: Diese Bedingung ist auch bei Einzahlungen erfüllt!)

Mit dem neuen Kontostand sollten auch die Anzahl verbleibender Kontobewegungen ermittelt, die Textfelder gelöscht und die Meldungen aktualisiert werden:

```
KtoBew = KtoBew - 1  
txtAusEu.Text = ""  
txtEinEu.Text = ""  
message
```

Weniger wichtig, aber korrekt: Noch vor Überprüfung der Überziehungsbedingung sollte sichergestellt werden, ob die Eingabefelder ungleich Null sind (damit die Kontobewegungen nicht durch versehentliches OK-Drücken „verwirkt“ werden):

```
If txtEinEu.Text <> 0 Or txtAusEu.Text <> 0 Then (...)
```

Allen bisherigen Überlegungen vorangestellt ist die Frage, ob die gewährten 5 Kontobewegungen inzwischen aufgebraucht worden sind – d.h.:

```
If KtoBew > 0 Then (...)
```

(Der formal-logisch naheliegende Gedanke, alle vorangegangenen Abfragen zusammenzufassen – etwa zu:

```
If KtoBew > 0 And (txtEinEu.Text <> 0 Or txtAusEu.Text <> 0)  
And txtAusEu.Text <= KtoStd) Then (...)
```

wird hier nur deswegen nicht weiterverfolgt, weil in der 2. Phase der Einbau spezieller Meldungen vorgesehen ist.)

1.2 Die Textfelder `txtEinEu` und `txtAusEu`

Wie im Übungsblatt bereits hingewiesen, können (und sollten) „Click-Ereignisse“ als „Indikatoren“ für die Art der Kontobewegung dienen. Daraus ergibt sich für die beiden Textfelder in der 1. Projektphase unmittelbar, daß ihre Click-Ereignisse das jeweils andere löschen sollen - konkret:

Sub `txtEinEu_Click()` soll auf alle Fälle die Zeile:

```
txtAusEu.Text = ""
```

enthalten; umgekehrt benötigt Sub `txtAusEu_Click()` die Zeile:

```
txtEinEu.Text = ""
```

Kurze Überlegung / Überprüfung führt zu dem Schluß, daß diese eine Zeile für die 1. Phase ausreicht.

1.3 Das Bezeichnungsfeld `lblEinDM`

Der letzte noch nicht besprochene Bestandteil der Benutzungsoberfläche in dieser 1. Phase ist die DM-Ausgabe bei der Einzahlung. Der Auslöser und die Daten hierfür kommen (gemäß Beschreibung im Übungsblatt) vom Textfeld `txtEinEu.Text`, konkret: vom Change-Ereignis dieses Textfeldes. Bei jeder Änderung seines Inhaltes soll der Label-Wert neu berechnet werden zu (hier wieder der Übersichtlichkeit zuliebe ohne die im Übungsblatt empfohlenen Funktionen `Format$()` und `numChk()`):

```
lblEinDM.Caption = txtEinEu.Text * Eu2DM
```

2 Zweite Projektphase

2.1 Die Funktion `numChk()`

Wie im Übungsblatt erläutert, dient die eingesetzte Funktion `numChk()` dazu, im Falle programmtechnisch unpassender Eingaben (die aber mathematisch-logisch evtl. durchaus erklärlich wären – z.B.: 10,- €) nachvollziehbare Ergebnisse zu zeitigen und vor allem „Programmabstürze“ (fehlerbedingte, vorzeitige Beendigungen) zu vermeiden. Die vorgegebene Minimal-Version:

```
numChk = Abs(Val(txt))
```

interpretiert den Inhalt des Textfeldes `txt` als Zahl und nimmt deren Absolutwert.

Für Projektphase 2 soll nun der Inhalt des Textfeldes `txtAusEu` auch dann zur Berechnung des aktuellen Kontostands herangezogen werden, wenn er aus dem DM-Betrag berechnet wurde und daher (nach VB-Konvention) mit einem Dezimal-

