



# EXCEL

# VBA

Beispiel  
Überweisungen





---

<b>AUFGABE</b>	<b>4</b>
<b>REALISIERUNG</b>	<b>5</b>
<b>ANALYSE DER AUFGABE</b>	<b>5</b>
<b>STAMMDATEN</b>	<b>6</b>
ERSTELLUNG EINER DATENBANK	6
WAS SIE BEI DER ERSTELLUNG EINER DATENBANK BEACHTEN SOLLTEN:	6
<b>HILFSTABELLE</b>	<b>7</b>
<b>HISTORIE</b>	<b>7</b>
<b>FORMULAR - UND EINGABEBLATT</b>	<b>8</b>
<b>PROGRAMMINFORMATION</b>	<b>9</b>
<b>PROGRAMMIERUNG DER APPLIKATION</b>	<b>10</b>
ÖFFNEN DER ARBEITSMAPPE	10
EINGABE STAMMDATEN	11
AUSWAHL ÜBER DROPDOWN	12
DRUCKEN DER ÜBERWEISUNG	13
AUSGABE DER AKTUELLEN LISTE	15
LÖSCHEN VON EINTRÄGEN	16
DRUCKEN DER AKTUELLEN LISTE	17
ALLGEMEINE PROZEDUREN	19
<i>Blätter wechseln</i>	19
<i>Initialisierung</i>	20
<b>STEUERUNG ÜBER TASTEN</b>	<b>21</b>
<b>SOURCEN</b>	<b>23</b>
SUB AUTO_OPEN()	23
SUB AUTO_CLOSE()	23
SUB EMPFÄNGER_EINGEBEN()	24
SUB AUSWAHLNAME()	25
SUB LISTEZEIGEN()	26
SUB LISTE_EINTRAGLÖSCHEN()	28
SUB EINGABEFELDER_INIT()	29
SUB PROGRAMM_INIT()	30
SUB DIALOG_INFO()	31
SUB EINGABEMASKE()	31
SUB HISTORY_ZEIGEN()	31
SUB TB_WECHSELN()	31
SUB STANDARDDRUCKER_EINSTELLEN()	32
SUB FORMULARDRUCKER_EINSTELLEN()	32
SUB FORMULARDRUCKRANGEEINSTELLEN()	33
SUB FORMULAR_DRUCKEN()	34
SUB LISTEDRUCK()	36



## Aufgabe

Es soll eine Anwendung in Excel erstellt werden, die es ermöglicht Überweisungsformulare auszudrucken. Die Überweisungen sollen über Stammdaten ausgefüllt und anschließend über einen Drucker ausgegeben werden.

Zu jeder Überweisung soll eine Klassifizierung angegeben werden, die die Art der Überweisung ausweist.

Jede Überweisung, die gedruckt wird, soll in einer Historie gespeichert werden. In der Historie sollen das Datum, Uhrzeit, Klassifizierung und alle Daten, die in der Überweisung eingetragen wurden, vermerkt werden.

Des weiteren soll eine Tagesliste erstellt bzw. ausgedruckt werden, die alle aktuellen Überweisungen anzeigt.

Alle Ein - bzw. Ausgaben sollen entweder über Tasten oder Schaltflächen realisiert werden. Die Auswahl der Zahlungsempfänger und Klassifizierung sollen über Auswahlboxen ( DropDownListbox ) realisiert werden.

Beim Öffnen der Applikation soll automatisch das Eingabeformular ausgewählt werden.

Es stehen zwei Drucker für die Ausgabe zur Verfügung.

Die Drucker sollen automatisch vor dem Ausdrucken ausgewählt und eingestellt werden, so daß der Anwender nicht in die Seiteneinrichtung von Excel wechseln muß.



## Realisierung

### Analyse der Aufgabe

Für die Eingabe in ein Überweisungsformular benötigen wir verschiedene Arten von Arbeitsblättern.

Als erstes müssen die Stammdaten und Hilfsdaten festgelegt werden. Die Stammdaten müssen alle Daten des Empfängers enthalten. Die Hilfsdaten benötigen wir, um die Steuerung über Makros zu erleichtern.

Die Hilfsdaten enthalten zum einem die Klassifizierung, sowie einen Zähler, der die Anzahl der Überweisungen ausgibt. Dieser Zähler ist nötig, damit die Tagesliste ausgedruckt werden kann.

Die Klassifizierung soll bei der Eingabe über eine DropDownListbox ausgewählt werden .

Für die Stammdaten und Hilfsdaten benutzen wir jeweils zur besseren Übersicht ein Tabellenblatt.

Für die Stammdaten benutzen wir Excel eigene Datenbank Funktionalitäten.

Jedes Programm oder Programmsteuerung sollte einen Copyright Vermerk haben. Dazu benutzen wir eine Dialogbox, in der der Name, die Version und der Programmierer vermerkt werden.

Für das Drucken der Überweisung benutzen wir ein Tabellenblatt, das die eingegeben Daten aufnimmt. Dieses Blatt muß genau auf das Formular, das später bedruckt wird, eingestellt werden. Das Einstellen des Bereiches für einen Formulardruck ist sehr zeitaufwendig und verbraucht viel Papier und Geduld. Über die Windows Druckereinstellungen müssen wir ein benutzerdefiniertes Papierformat, das der Größe des zu druckenden Formulars entspricht, einstellen.

Anschließend müssen durch Testen und Drucken die Zeilenhöhen und Spaltenbreiten eingestellt werden. Daraus ergeben sich auch die Zellbezüge, die wir später zum Ausfüllen des Formulars benötigen.

Für die Eingabe in das Formular benutzen wir ebenfalls ein Tabellenblatt. Dazu kopieren wir das vorher eingestellte Formular und verknüpfen die Zellen der Eingabe mit denen des Formulars. Alle Eingaben im Eingabeformular werden dann automatisch im Druckformular eingestellt.

Für die Historie und Tagesliste werden Tabellenblätter benutzt.

Als letzte Arbeitsblätter benutzen wir VBA Modulblätter, die die Steuerung der kleinen Applikation aufnehmen sollen. Wir erstellen ein VBA Modul für die Eingabesteuerung und ein Modulblatt für die Ausgabesteuerung.



## Stammdaten

### Erstellung einer Datenbank

Es gibt zwei Möglichkeiten, eine Datenbank zu erstellen:

- Jede Excel Tabelle, die bestimmte Kriterien erfüllt (Feldüberschriften in der ersten Zeile, gleichartige Daten in allen Datenfeldern einer Spalte), wird von Excel automatisch als Datenbank erkannt, sobald eine beliebige Zelle der Tabelle aktiviert wird. Mit anderen Worten: Sie können die Daten wie gewohnt ins Tabellenblatt eingeben. Das Tabellenblatt darf neben der Datenbank - Tabelle weitere Daten wie Texte, Berechnungen usw. enthalten, wenn diese durch mindestens eine Leerzeile bzw. Leerspalte von der Datenbank - Tabelle getrennt sind
- Als Arbeitserleichterung bei der Erstellung von Datenbanken stellt Excel eine Eingabemaske zu Verfügung :

Wenn Sie mit dieser Maske arbeiten möchten, geben Sie im Tabellenblatt lediglich die erste Zeile der Datenbank - also die Feldnamen - ein und aktivieren die Maske im Menü Daten / Maske. Excel stellt in der Maske zu jedem Feldnamen ein Eingabefeld zur Verfügung, in das die Daten eingegeben werden können. Sind alle Datenfelder der Maske gefüllt, ist der erste Datensatz komplett und wird in die Datenbank eingetragen. Excel zeigt wieder eine leere Maske an, in die der nächste Datensatz eingegeben werden kann

Sie müssen sich zwischen beiden Eingabearten nicht ein für alle Mal entscheiden, sondern können die Datenbank auch nachträglich, wahlweise mit Hilfe der Eingabemaske oder direkt im Tabellenblatt, ergänzen.

### Was Sie bei der Erstellung einer Datenbank beachten sollten:

- Jeder Feldname sollte eindeutig sein, d.h. nur einmal in der Datenbank vergeben werden.
- Grundsätzlich sollten alle Daten jeweils in separate Datenfelder eingegeben werden. Z.B. ist es nicht sinnvoll, Postleitzahl und Ort in das gleiche Datenfeld einzugeben, da Excel dadurch die Möglichkeit genommen wird, getrennt nach Postleitzahlen und Orten zu suchen. Es wäre beispielsweise nicht möglich, alle Postleitzahlen anzuzeigen, die größer 10000 sind, da der Vergleichsoperator  $>$  nicht auf eine Zelle angewandt werden kann, die auch Text enthält.

Berechnet sich ein Datenfeld aus anderen Datenfeldern der Datenbank, so sollten die entsprechenden Formeln oder Funktionen vor Aufruf der Eingabemaske in die Tabelle eingetragen werden. Excel zeigt dann in der Maske für die berechneten Felder, anstelle eines Eingabefeldes, das Ergebnis der Berechnung an. Dadurch wird vermieden, daß das Ergebnis einer Berechnung durch eine möglicherweise falsche Eingabe überschrieben werden kann.



Für unser Beispiel geben wir im ersten Schritt nur die Überschriften an. Die Daten wollen wir später über ein Makro eingeben und verwalten.

Die Überschriften lauten:

Empfänger BANK, KONTONR, BLZ, Verwendungszweck1 und Verwendungszweck2

## Hilfstabelle

Die Hilfstabelle soll alle zusätzlichen Informationen und Definitionen aufnehmen.

	A	B	
1	<b>Klassifizierung</b>	<b>AnzahlÜberweisungen</b>	
2	Abschlag		
3	Pfändung		
4	Sonstiges		
5	Überweisung		
6			

Das Tabellenblatt mit den Hilfsdaten nennen wir „Hilfstabelle“. Die Festlegung eines Namens sollte immer „sprechend“ sein, so daß wir aufgrund des Namens erkennen können, welche Funktionen oder Informationen sich in diesem Blatt befinden. Auf den Namen müssen wir später bei der Steuerung zugreifen.

## Historie

Die Historie nimmt Informationen über alle gedruckten Überweisungen auf.

In dieser Tabelle werden alle Daten der Überweisung mit Datum und Uhrzeit vermerkt, so daß später eine Übersicht entsteht.

In der ersten Spalte werden Datum und Uhrzeit vermerkt. In den weiteren Spalten alle Daten der Überweisung und zusätzlich die Klassifizierung.

Zum Erstellen der Historie können wir die Stammdatentabelle mit den Empfängern kopieren und das Datumfeld sowie die Klassifizierung einfügen.



## Formular - und Eingabeblatt

Wie schon erwähnt, ist das Erstellen von Formularen in Excel nicht so einfach.

Zur Erstellung des Eingabeformulars benutzen Sie bitte die Vorlage aus dem Beispiel Verzeichnis.

Kopieren Sie das Formular innerhalb der Arbeitsmappe und verknüpfen Sie die Felder des Eingabeblattes mit dem Formular .

	A	B	C	D	E	F	G	H
1								
2								
3			=Eingabe!C3					
4								
5			=Eingabe!C5					=Eingabe!H5
6								
7			=Eingabe!C7					
8							=Eingabe!G8	
9								
10			=Eingabe!C10					
11								
12			=WENN(Eingabe!C12=0;"";Eingabe!C12)					
13			=Eingabe!C13					
14			=Eingabe!C14					
15								
16								

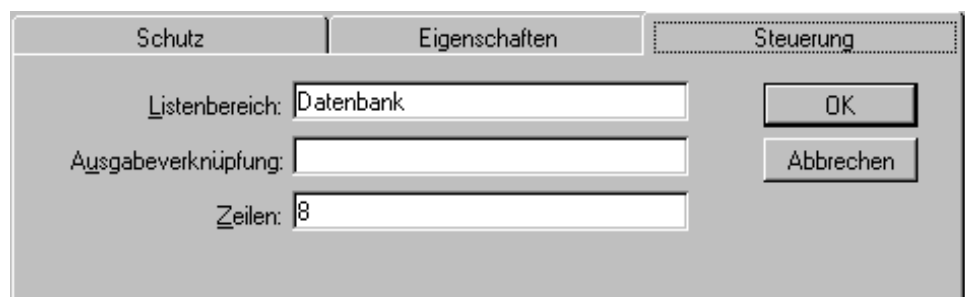
In dem Eingabeformular müssen jetzt Verbindungen zu der Stammdatentabelle und Hilfstabelle erstellt werden, die es uns ermöglichen Daten auszuwählen.

Dazu benutzen wir Dropdownlistboxen. Eine Dropdownlistbox ist ein Objekt, das man in ein Tabellenblatt oder ein Dialogblatt einbinden kann. Dieses Objekt ermöglicht uns, aus einer Liste einen Eintrag auszuwählen und anschließend mit dem Ergebnis (ist eine Zahl) weitere Anweisungen durchzuführen.

Weitere Erläuterungen zum DropDown finden Sie in den Unterlagen VBA Seite 57. Dort finden Sie eine ausführliche Erklärung.

Über die Symbolleiste „Dialog“ wählen wir das DropDown Element aus und setzen es im Eingabeformular in die Zelle Empfänger.

Über den Menüpunkt „Objekt formatieren“ weisen wir dem DropDown die Datenbank zu!



Weitere Eigenschaften müssen in diesem Beispiel nicht eingestellt werden!

Auch für die Klassifizierungen benutzen wir ein DropDown, das wir als Listenbereich dem Bereich in der Hilfstabelle zuweisen. Die Steuerung der DropDowns übernimmt später ein Makro!





Das Eingabeformular ist fertig und kann wie in der Abbildung formatiert werden.

	B	C	D	E	F	G	H	I	J	K
1										
2		<b>Empfänger</b>				<b>Art der Zahlung</b>				
3		<input type="text"/>				<input type="text"/>				
4		<b>Kontonummer</b>				<b>BLZ</b>				
5										
6		<b>Bank</b>								
7						<b>DM</b>				
8						0,00				
9		<b>Verwendungszweck 1</b>								
10										
11		<b>Verwendungszweck 2</b>								
12										
13		Sabine Mustermann								
14		4210444000								

## Programminformation

Zur Anzeige der Programminformation benutzen wir ein Dialogblatt, in dem alle Informationen zum Programm angezeigt werden sollen. Wenn ein neues Dialogblatt eingefügt wurde, wird von Excel eine Standard Dialogbox erstellt, die man dann für die jeweiligen Anforderungen verändern kann.

Mehr Informationen bekommen Sie im Kapitel „VBA Benutzerdefinierte Dialogfelder und Elemente“ auf Seite 45. Die Dialogbox für die Überweisungen kann wie in der folgenden Abbildung aufgebaut sein.





## Programmierung der Applikation

In den folgenden Abschnitten werden alle Programmschritte zur Steuerung der Eingabe erklärt.

Für jede Teilfunktion werden Unterprogramme benutzt, die einen strukturierten Aufbau ermöglichen und auch noch zu einem späteren Zeitpunkt eine Analyse erleichtern.

Als erstes fügen wir ein Modulblatt ein, was den Namen „VBA Steuerung“ erhält.

### Öffnen der Arbeitsmappe

Das erste Unterprogramm enthält die Anweisungen, die beim Öffnen der Excel Datei ausgeführt werden.

```
Sub Auto_open()  
    Dialog_info  
    StandardDrucker_Einstellen  
    Worksheets("Eingabe").Select  
    ActiveWindow.WindowState = xlMaximized  
    EingabeFelder_Init  
    Application.OnSheetActivate = "TB_Wechseln"  
End Sub
```

Der Name „Auto\_open“ ist eine Excel Standard Prozedur . Wenn Excel beim Öffnen einer Datei diese Prozedur findet, werden alle Anweisungen innerhalb der Prozedur ausgeführt. Sie wird in unserem Beispiel dazu benutzt, Einstellungen für das weitere Arbeiten mit der Datei vorzunehmen.

In der ersten Zeile wird eine Prozedur aufgerufen, die uns die Programminformation anzeigt. Nach Abschluß der Prozedur „Dialog\_Info“ wird die nächste Zeile ausgeführt. Diese stellt den Standarddrucker ein. Als nächstes wird in das Tabellenblatt „Eingabe“ gewechselt.

Die Anweisung „ActiveWindow.WindowState = xlMaximized“ ist eine Excel Eigenschaft für das aktive Fenster. Mit Zuweisung der Konstanten „xlMaximized“ wird das Fenster maximiert. Nächster Schritt ist die Initialisierung der Eingabefelder und Zurücksetzen der Werte in der Hilfstabelle.

Mit der Anweisung „Application.OnSheetActivate = "TB\_Wechseln"“ können wir beim Wechseln der Blätter über das Excelregister bestimmte Aktionen ausführen. Wenn ein Anwender über das Register das Blatt wechselt, wird eine Prozedur mit dem Namen TB\_Wechseln ausgeführt. Die Anweisungen in dieser Prozedur betrachten wir zu einem späteren Zeitpunkt.



## Eingabe Stammdaten

Über einen Makroaufruf soll die Eingabe der Empfänger realisiert werden.

Die Eingabe kann über Excel-Funktionalitäten erreicht werden. Da unsere Liste eine Datenbank ist, wollen wir die Standardmaske von Excel für Eingaben von Daten benutzen.

Die Maske kann nur aufgerufen werden, wenn man sich auf dem Tabellenblatt befindet, in dem sich die Daten befinden.

Über die Makroaufzeichnung erstellen wir das Unterprogramm.

Das Ergebnis könnte wie folgt aussehen:

```
Sub Makro1()  
    Sheets("Empfänger").Select  
    ActiveSheet.ShowDataForm  
End Sub
```

Da wir nach dem Wechsel in das Blatt Empfänger nicht wissen, ob die aktive Zelle im Datenbankbereich ist, müssen wir per Anweisung sicherstellen, daß vor Aufruf der Maske die aktive Zelle im Datenbankbereich liegt.

Das kann über die Anweisung `Range("Datenbank").Select` geschehen.

Anschließend geben wir dem Unterprogramm noch einen sinnvollen Namen, z.B. `Empfänger_eingeben`.

Nach Schließen der Eingabemaske soll wieder auf das Eingabeformular gewechselt werden.

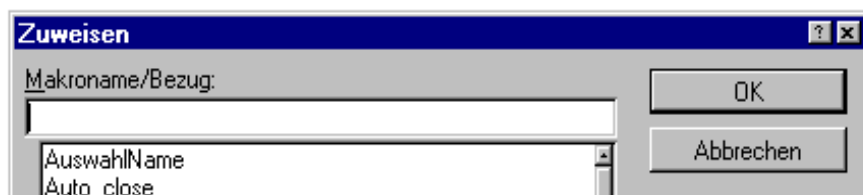
Das erreichen wir mit der Anweisung `Sheets(„Eingabe“).Select`. Das fertige Unterprogramm könnte wie folgt aussehen.

```
Sub Empfänger_eingeben()  
    Sheets("Empfänger").Select  
    Range("Datenbank").Select  
    ActiveSheet.ShowDataForm  
    Sheets("Eingabe").Activate  
End Sub
```

Das Unterprogramm soll über eine Schaltfläche im Eingabeblatt aufgerufen werden.

Im Eingabeblatt wird über die Dialogsymbolleiste eine Schaltfläche selektiert und auf das Eingabeblatt gesetzt.

In dem nun folgenden Excel Dialog kann man der Schaltfläche ein Makro zuweisen.



Wir wählen den Makro `Empfänger_eingeben` aus.

Anschließend können wir die Funktionalität der Prozedur testen.



## Auswahl über DropDown

Als nächstes benötigen wir eine Prozedur, die das Auswählen der Namen aus der Datenbank ermöglichen. Wenn ein Anwender einen Eintrag in der *DropDown* auswählt, müssen alle Daten des Empfängers in das Formular eingetragen werden.

Zuerst initialisieren wir die Variablen, die uns die Eingabe der Prozedur erleichtern sollen.

Die erste Variable soll die Nummer des Eintrages aus dem *DropDown* aufnehmen, der ausgewählt wurde. Diese Variable muß vom Typ *Integer* sein. Die zweite Variable ermöglicht es, ein komplettes Objekt aufzunehmen, so daß wir im folgenden weniger Schreibaufwand haben und die Prozedur übersichtlicher wird. Diese Variable muß vom Typ *Object* sein.

Die nun folgende Anweisung speichert die Nummer des ausgewählten Eintrages in die Variable *li\_Zeile*. Mit Hilfe dieses Wertes können wir die restlichen Informationen aus der Stammdatentabelle auslesen. Die Nummer des Eintrages spiegelt die Position in der Datenbank wieder.

Als nächstes setzen wir mit Hilfe der Anweisung *SET* die Objektvariable auf das Objekt *Worksheets("Eingabe")*. Im nächsten Schritt müssen wir jedem Feld des Eingabeblattes den Wert aus der Datenbank zuweisen. Dazu benutzen wir die Methoden und Eigenschaften des Tabellenblattes. Diese sind im einzelnen:

*RANGE*, *CELLS* und *VALUE*

Damit nicht in jeder Zeile doppelte Anweisungen stehen, benutzen wir zusätzlich die Anweisung *WITH*. Mit Hilfe dieser Anweisung können Informationen von Objekten ausgelesen bzw. gesetzt werden.

Ausführliche Informationen zu *WITH*, *RANGE*, *CELLS* und *VALUE* finden Sie in der Excel Hilfe.

Die fertige Prozedur könnte wie folgt aussehen:

```
Sub AuswahlName()  
  Dim li_Zeile As Integer  
  Dim lo_Eingabe As Object  
  li_Zeile = Worksheets("Eingabe").DropDowns("dd_name").Value  
  Set lo_Eingabe = Worksheets("Eingabe")  
  With Worksheets("Empfänger")  
    lo_Eingabe.Range("c3:c3").Value = .Cells(li_Zeile; 2).Value  
    lo_Eingabe.Range("c7:c7").Value = .Cells(li_Zeile; 3).Value  
    lo_Eingabe.Range("c5:c5").Value = .Cells(li_Zeile; 4).Value  
    lo_Eingabe.Range("h5:h5").Value = .Cells(li_Zeile; 5).Value  
    lo_Eingabe.Range("c10:c10").Value = .Cells(li_Zeile; 6).Value  
    lo_Eingabe.Range("c12:c12").Value = .Cells(li_Zeile; 7).Value  
    lo_Eingabe.Range("g8:g8").Value = 0  
  End With  
End Sub
```

Für die Auswahl der Klassifizierung benötigen wir kein Makro. Der Inhalt wird später über das Makro zum Drucken des Formulars ausgelesen.



## Drucken der Überweisung

Zum Drucken der Überweisung benötigen wir ein Makro, das nach Drucken der Überweisungen alle nötigen Einträge in den Tabellenblättern *Liste* und *History* durchführt.

Als erstes deklarieren wir die benötigten Variablen. Anschließend weisen wir den Variablen die entsprechenden Werte zu. Wir benutzen dazu wieder Variablen vom Typ *Integer* und *Object*.

Ein wichtige Zuweisung ist die aktuelle Nummer der *DropDown* Arten. Aus dieser Variablen können wir später in der Prozedur den Wert auslesen.

```
Dim li_Zeile As Integer
Dim lo_dpdw; lo_Eingabe; lo_Stamm As Object
Set lo_Eingabe = Worksheets("Eingabe")
Set lo_dpdw = lo_Eingabe.DropDowns("dd_arten")
Set lo_Stamm = Worksheets("Empfänger")
```

Nach Zuweisung der Variablen erhöhen wir die Anzahl der Einträge für die aktuelle Liste und der Historie.

```
Worksheets("HilfsTabelle").Range("AnzAktuell").Value = _
    Worksheets("HilfsTabelle").Range("AnzAktuell").Value + 1
Worksheets("HilfsTabelle").Range("AnzGesamt").Value = _
    Worksheets("HilfsTabelle").Range("AnzGesamt").Value + 1
```

Anschließend wird der Formulardrucker eingestellt und die Überweisung gedruckt. Da wir eine direkte Verknüpfung mit dem Eingabeformular haben, müssen keine weiteren Werte vor dem Druck eingestellt werden. Vor dem Drucken muß noch der Formulardrucker ausgewählt werden und anschließend wird über die Anweisung *Worksheets("Formular").PrintOut* . das Überweisungsformular gedruckt. Nach dem Drucken wird wieder der Standarddrucker ausgewählt.

Im nächsten Schritt werden alle nötigen Eintragungen in den Tabellenblättern *Liste* und *History* durchgeführt. Dazu benutzen wir die Methoden und Eigenschaften von *Worksheets*.

Als erstes holen wir den aktuellen Wert aus der Hilfstabelle und weisen den Wert der Integer Variablen zu. Diese Variable wird als Offset benötigt, um die Eintragungen am Ende der Liste vorzunehmen.

```
li_Zeile = Worksheets("HilfsTabelle").Range("B2").Value + 2
```

Genauso verfahren wir für die Einträge in der Historie. Da der Wert der Historie einen anderen Wert als die Liste hat, müssen wir den Wert extra betrachten und der Integer Variablen zuweisen.

```
li_Zeile = Worksheets("HilfsTabelle").Range("AnzGesamt").Value + 2
```

In der Historie muß zusätzlich noch die Klassifizierung eingetragen werden.

Das wird mit der Anweisung `.Cells(li_Zeile; 9).Value = Worksheets("Hilfstabelle")._`  
`Cells(lo_dpdw.Value + 1; 1).Value`

realisiert.

Wenn alle Informationen übernommen wurden, wird wieder das Eingabeformular aktiviert.

```
lo_Eingabe.Activate
```



Die fertige Prozedur könnte wie folgt aufgebaut sein:

```
Sub Formular_Drucken()  
Dim li_Zeile As Integer  
Dim lo_dpdw; lo_Eingabe; lo_Stamm As Object  
Set lo_Eingabe = Worksheets("Eingabe")  
Set lo_dpdw = lo_Eingabe.DropDowns("dd_arten")  
Set lo_Stamm = Worksheets("Empfänger")  
Worksheets("HilfsTabelle").Range("AnzAktuell").Value = _  
    Worksheets("HilfsTabelle").Range("AnzAktuell").Value + 1  
Worksheets("HilfsTabelle").Range("AnzGesamt").Value = _  
    Worksheets("HilfsTabelle").Range("AnzGesamt").Value + 1  
  
Worksheets("Formular").Aktivieren  
FormularDrucker_Einstellen  
Worksheets("Formular").PrintOut  
StandardDrucker_Einstellen  
  
li_Zeile = Worksheets("HilfsTabelle").Range("B2").Value + 2  
With Worksheets("Liste")  
    .Cells(li_Zeile; 1).Value = lo_Eingabe.Range("c3:c3").Value  
    .Cells(li_Zeile; 2).Value = lo_Eingabe.Range("c7:c7").Value  
    .Cells(li_Zeile; 3).Value = lo_Eingabe.Range("c5:c5").Value  
    .Cells(li_Zeile; 4).Value = lo_Eingabe.Range("h5:h5").Value  
    .Cells(li_Zeile; 5).Value = lo_Eingabe.Range("c10:c10").Value  
    .Cells(li_Zeile; 6).Value = lo_Eingabe.Range("c12:c12").Value  
    .Cells(li_Zeile; 7).Value = lo_Eingabe.Range("g8:g8").Value  
End With  
li_Zeile = Worksheets("HilfsTabelle").Range("AnzGesamt").Value + 2  
With Worksheets("History")  
    .Cells(li_Zeile; 1).Value = Now  
    .Cells(li_Zeile; 2).Value = lo_Eingabe.Range("c3:c3").Value  
    .Cells(li_Zeile; 3).Value = lo_Eingabe.Range("c7:c7").Value  
    .Cells(li_Zeile; 4).Value = lo_Eingabe.Range("c5:c5").Value  
    .Cells(li_Zeile; 5).Value = lo_Eingabe.Range("h5:h5").Value  
    .Cells(li_Zeile; 6).Value = lo_Eingabe.Range("c10:c10").Value  
    .Cells(li_Zeile; 7).Value = lo_Eingabe.Range("c12:c12").Value  
    .Cells(li_Zeile; 8).Value = lo_Eingabe.Range("g8:g8").Value  
    .Cells(li_Zeile; 9).Value = Worksheets("HilfsTabelle"). _  
        Cells(lo_dpdw.Value + 1; 1).Value  
End With  
lo_Eingabe.Activate  
End Sub
```

Die Aktivierung des Makros wird über eine Schaltfläche auf dem Eingabeformular realisiert.



### **Ausgabe der aktuellen Liste**

In dieser Prozedur soll die Ausgabe der Liste realisiert werden. Im Normalfall bräuchte man für die Ausgabe kein Makro, aber in unserem Fall wollen wir ein kleines Bonbon einbauen. Auf diesem Tabellenblatt sollen nicht nur alle aktuell gedruckten Überweisungen stehen, sondern am Ende der Liste die Summe aller Beträge. Da die Liste dynamisch ist, muß die Summenformel immer an der richtigen Stelle und mit den richtigen Bezügen aufgebaut werden. Dazu bietet Excel die Möglichkeit über Makros Tabellenfunktionen einzufügen.

Als erstes definieren wir die benötigten Variablen vom Typ Integer für die aktuelle Anzahl der Zeilen. Zwei Variablen vom Typ String. Diese Variablen sollen die Adresse des Summenbereiches aufnehmen. Und eine Variable vom Typ Objekt, die eine Zeile beschreiben soll. Die erste Anweisung aktiviert das Tabellenblatt *Liste*. Aus dem Tabellenblatt *Hilfstabelle* weisen wir der Integer Variablen, die Anzahl der bisher gedruckten Einträge zu. Mit den folgenden Anweisungen legen wir die Excel spezifische Adressierung von Zellen fest.

Dieses wird mit der Methode *ADDRESS* und den Eigenschaften *rowabsolute*, *columnabsolute* und *ReferenceStyle* vorgenommen. Wir benötigen für die Formel den Anfangs -und Endbereich der zu addierenden Werte. Den Anfangsbereich weisen wir der String Variablen *ls\_rel1* zu und den Endbereich *ls\_rel2*. Anschließend tragen wir die Summenformel in der letzten Zeile ein, die wir aufgrund der Anzahl der Einträge ermitteln können.

Der Zelle, in der die Formel eingetragen wird, weisen wir noch ein Zahlenformat zu.

*.NumberFormat = "#.##0,00 DM"*.

In der Zelle neben der Formel tragen wir den Text „Gesamt“ ein und formatieren die Zelle rechtsbündig.

Die fertige Prozedur könnte wie folgt aufgebaut sein:

```
Sub ListeZeigen()  
Dim li_Anzahl As Integer  
Dim ls_rel1; ls_rel2 As String  
Dim lo_zelle As Object  
Worksheets("Liste").Activate  
li_Anzahl = Worksheets("HilfsTabelle").Range("B2").Value  
ls_rel1 = Cells(3; 7). Address(rowabsolute:=False; columnabsolute:=False; _  
ReferenceStyle:=xlA1)  
ls_rel2 = Cells(li_Anzahl+2;7). Address(rowabsolute:=False  
columnabsolute:=False; ReferenceStyle:=xlA1)  
With Cells(li_Anzahl + 3; 7)  
.Formula = "=SUM(" & ls_rel1 & ":" & ls_rel2 & ")"  
.NumberFormat = "#.##0,00 DM"  
End With  
Set lo_zelle = Cells(li_Anzahl + 3; 6)  
With lo_zelle  
.FormulaR1C1 = "Gesamt:"  
.HorizontalAlignment = xlRight  
.VerticalAlignment = xlBottom  
.WrapText = False  
.Orientation = xlHorizontal  
End With  
End Sub
```



Die nun folgende Prozedur wird benötigt, um zu gewährleisten, daß immer dann die vorherige Prozedur ausgeführt wird, wenn in das Tabellenblatt *Liste* gewechselt wird.

Der Aufruf der Prozedur geschieht über die Zuweisung in der *Auto\_open* Prozedur .

Die erste Anweisung prüft, ob das aktuelle Tabellenblatt den Namen *Liste* hat, wenn ja, wird die Prozedur *ListeZeigen* aufgerufen, ansonsten wird die Ausführung ohne weitere Aktionen fortgesetzt.

```
Sub TB_Wechseln()  
If ActiveSheet.Name = "Liste" Then  
    ListeZeigen  
End If  
End Sub
```

### **Löschen von Einträgen**

Eine wichtige Prozedur für das Arbeiten mit der Liste ist das Löschen von Einträgen.

Wenn der Anwender eine Zeile auswählt und sie löscht, werden die Anzahl der Einträge nicht automatisch aktualisiert. Das kann dazu führen, daß die Summenformel an der falschen Stelle eingetragen wird.

In Excel gibt es keine Möglichkeit das Löschen von Zeilen abzufangen. An dieser Stelle muß man dem Anwender durch Hilfsfunktionen das Löschen der Einträge ermöglichen.

Die folgende Prozedur löscht die aktuelle Zeile und aktualisiert den Wert in der Hilfstabelle.

```
Sub Liste_EintragLöschen()  
Dim li_Anzahl As Integer  
li_Anzahl = Worksheets("HilfsTabelle").Range("B2").Value  
If li_Anzahl > 0 Then  
    Selection.EntireRow.Delete  
    Worksheets("HilfsTabelle").Range("B2").Value = _  
        Worksheets("HilfsTabelle").Range("B2").Value - 1  
End If  
End Sub
```

Über die Anweisung *Selection.EntireRow.Delete* wird immer eine ganze Zeile gelöscht.

Anschließend wird der Zähler um eins herunter gezählt.

Vollständigkeitshalber muß noch erwähnt werden, daß auch die Historie um diesen Eintrag reduziert werden muß.

Der Aufruf der Prozedur kann über eine Schaltfläche auf dem Tabellenblatt *Liste* vorgenommen werden.

Ein kleiner Tip :

Wenn die Schaltfläche groß genug ist, wird der Anwender niemals auf die Idee kommen, über das Bearbeiten Menü einen Eintrag zu löschen!





### **Drucken der aktuellen Liste**

Über eine weitere Schaltfläche kann der Druck der Liste durchgeführt werden. Das hat den Vorteil das immer der richtige Drucker ausgewählt wird. Die Auswahl des Druckers wird über die Prozedur *StandardDrucker\_Einstellen* realisiert.

Das Einstellen eines Druckers mit Hilfe von VBA ist nicht so einfach. Der Name des Druckers muß genauso angegeben werden wie er in der Systemsteuerung benannt ist, ansonsten kommt es zu Fehlermeldungen. Um den Namen eines Druckers zu ermitteln, benutzt man am besten eine Makroaufzeichnung. Die Schritte zur Ermittlung des Druckers im einzelnen: Menü Datei, Drucken, Drucker auswählen und anschließend *Abbrechen* betätigen. Makro Aufzeichnung stoppen. In dem nun erzeugten Modul müßte der Aufruf zur Druckereinstellung enthalten sein.

```
Sub StandardDrucker_Einstellen()  
    Anwendung.AktiverDrucker = "HP LaserJet 4P/4MP auf LPT1:"  
End Sub
```

Als nächstes benötigen wir eine Prozedur, die das Drucken der Liste steuert. Immer wenn die Liste ausgedruckt wird, soll der Benutzername, Datum und Uhrzeit in der Fußzeile plaziert werden. Da man nicht weiß, ob der Anwender die aktuellen Druckereinstellungen verändert hat muß diese Prozedur gewährleisten, daß diese Einträge immer mit ausgedruckt werden. Da die Liste dynamisch ist, muß auch der Druckbereich immer der Größe der Liste angepaßt werden.

Als erstes werden wieder die Variablen deklariert. Wir benötigen in dieser Prozedur nur eine Integer Variable, der wir die Anzahl der Einträge zuweisen.

Die erste Anweisung in der Prozedur ruft die Unterprozedur *StandarDrucker\_Einstellen* auf. Anschließend wird das Tabellenblatt *Liste* mit der Anweisung *Worksheets("Liste").Activate* aktiviert.

Mit der Anweisung

```
Range(Cells(3; 1); Cells(li_Anzahl + 7; 7)).Select
```

selektieren wir den Druckbereich der Tabelle. Die Größe des Bereiches können wir aufgrund der Anzahl der bisher gedruckten Überweisungen, die in der Hilfstabelle eingetragen sind, ermitteln.

Die nächsten Anweisungen dienen der Seiteneinrichtung.

Welche Eintragungen für die Formatierung nötig sind, kann man am besten durch Aufzeichnung eines Makros ermitteln. Vor Aufruf der Seiteneinrichtung wird nur die Aufzeichnung eines Makros gestartet.

Die letzte Anweisung druckt schließlich das Tabellenblatt aus.



Die fertige Prozedur könnte wie folgt aufgebaut sein:

```
Sub ListeDruck()  
    Dim li_Anzahl As Integer  
    StandardDrucker_Einstellen  
    Worksheets("Liste").Activate  
    li_Anzahl = Worksheets("HilfsTabelle").Range("AnzAktuell").Value + 2  
    Range(Cells(3; 1); Cells(li_Anzahl + 7; 7)).Select  
    With Worksheets("Liste").PageSetup  
        .PrintTitleRows = "$2:$2"  
        .PrintArea = Selection.Address  
        .Orientation = xlQuerformat  
        .PrintTitleColumns = ""  
        .LeftFooter = "&8" & Application.UserName & _  
            " " & ActiveWorkbook.Path & _  
            "\&N" & " [" & Date & "/" & Time & "]"  
        .CenterFooter = ""  
        .RightFooter = "Seite &P von &N"  
    End With  
    Worksheets("Liste").PrintOut  
End Sub
```

Die Schaltflächen auf dem Tabellenblatt könnten wie in der folgenden Abbildung positioniert werden.

	A	B	C	D	E	F	G
1	Liste drucken	Eintrag löschen					
2	Empfänger	Bank		BLZ	YVZ1		Betrag



## **Allgemeine Prozeduren**

Für die weitere Steuerung der kleinen Applikation werden noch einige Hilfsfunktionen benötigt.

### **Blätter wechseln**

Der Anwender soll die Möglichkeit haben, von dem Eingabeformular zu den einzelnen Tabellenblättern zu wechseln. Dazu benötigen wir zusätzliche Schaltflächen auf dem Eingabeformular.

Über eine Schaltfläche könnte man z.B. zur Historie wechseln.

Die Prozeduren haben jeweils nur eine Anweisung:

```
Sub Dialog_info()  
    DialogSheets("info").Show  
End Sub  
Sub EingabeMaske()  
    Worksheets("Eingabe").Activate  
End Sub  
Sub History_Zeigen()  
    Worksheets("History").Activate  
End Sub
```



## Initialisierung

Benötigt wird auch eine Schaltfläche, die die Eingabefelder und die Liste zurücksetzt. Das realisieren wir in zwei Prozeduren:

Als erstes schreiben wir eine Prozedur, die alle Einträge aus der Tabelle Liste entfernt. Aufgrund des Wertes aus der Hilfstabelle können wir ermitteln wieviele Einträge in der Tabelle existieren.

Über eine Schleife löschen wir jeden Eintrag mit der Anweisung

```
.Delete Shift:=xlUp
```

Die letzte Anweisung ruft eine Prozedur auf, die alle Eingabefelder und DropDowns zurücksetzt.

Die fertigen Prozeduren könnten wie folgt aufgebaut sein:

```
Sub Programm_Init()  
  Dim li_Anzahl ;li_Zähler As Integer  
  li_Anzahl = Worksheets("HilfsTabelle").Range("B2").Value  
  For li_Zähler = 1 To li_Anzahl + 7  
    Worksheets("Liste").Rows(3).Delete Shift:=xlUp  
  Next  
  EingabeFelder_Init  
End Sub
```

```
Sub EingabeFelder_Init()  
  With Worksheets("Eingabe")  
    .Range("c3").Value = ""  
    .Range("c5").Value = ""  
    .Range("h5").Value = ""  
    .Range("c7").Value = ""  
    .Range("c10").Value = ""  
    .Range("c12").Value = ""  
    .DropDowns("dd_name").Value = 0  
    .DropDowns("dd_arten").Value = 0  
  End With  
End Sub
```



## Steuerung über Tasten

Die Anwendung wäre in der Grundstruktur fertig, und wir können den Anwender darauf loslassen.

Zusätzlich könnte man das Eingabeformular über eine Tastenkombination aufrufen. Dazu wählen Sie aus dem Menü Extras den Eintrag Makro...

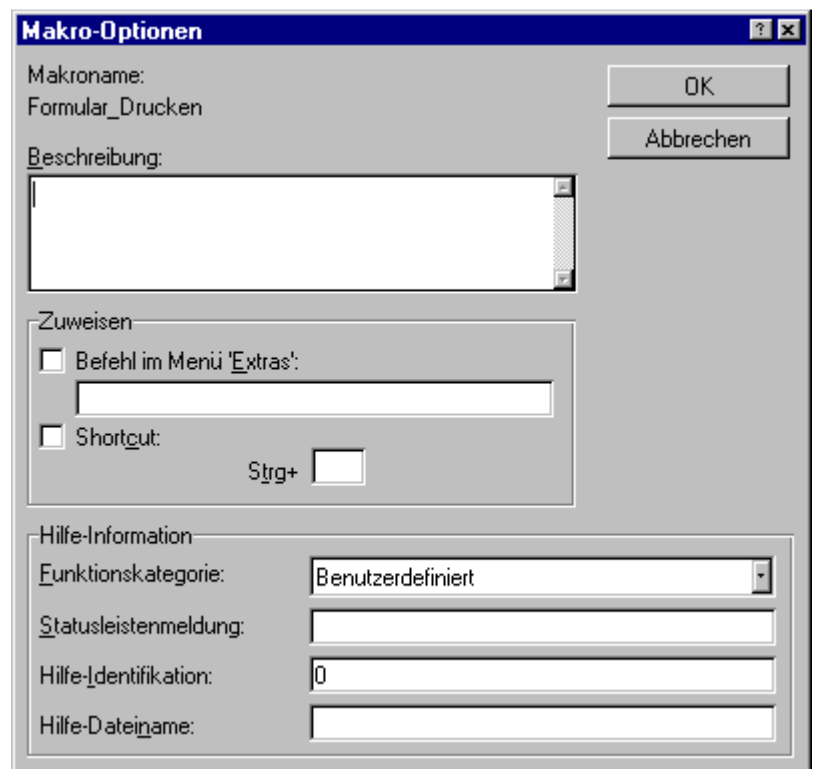
In dem angezeigten Dialog können Sie weitere Einstellungen an den Makros vornehmen. Wenn Sie einen Makro auswählen, werden die Schaltflächen „Enabled“. Wenn Sie die Schaltfläche *Optionen* klicken, haben Sie die Möglichkeit zu dem Makro weitere Optionen einzustellen.

In dem ersten Feld „Beschreibung“ können Sie die Funktionalität des Makros beschreiben. In dem Bereich Zuweisen können Sie entweder einen neuen Eintrag im Menü Extras vornehmen oder einen sogenannten *ShortCut* zuweisen. Dieser ermöglicht dann den Aufruf über eine Tastenkombination.

In dem Bereich Hilfsinformationen können Eintragungen zur Anwenderführung eingegeben werden, die dem Anwender durch die Anwendung führt.

In unserem Beispiel weisen wir dem Makro *Formular\_drucken* die Tastenkombination *STRG+F* zu und dem Makro *EingabeMaske* die Kombination *STRG+E*.

Anschließend kann der Anwender ohne die Maus zu benutzen den Formulardruck und das Eingabeformular wählen. Für die Makros zur Auswahl der Liste und History können ebenfalls noch ShortCuts definiert werden.





## Sourcen

\*\*\*\*\*

### Sub *Auto\_open()*

```
Dialog_info  
StandardDrucker_Einstellen  
Worksheets("Eingabe").Select  
ActiveWindow.WindowState = xlMaximized  
EingabeFelder_Init  
Application.OnSheetActivate = "TB_Wechseln"
```

### End Sub

\*\*\*\*\*

' wird beim schließen der Arbeitsmappe aufgerufen

\*\*\*\*\*

### Sub *Auto\_close()*

```
ActiveWindow.Close
```

### End Sub



\*\*\*\*\*

' Aufruf der EingabeMaske für die Datenbank

\*\*\*\*\*

**Sub Empfänger\_eingeben()**

Sheets("Empfänger").Activate

Range("Datenbank").Select

ActiveSheet.ShowDataForm

Sheets("Eingabe").Activate

**End Sub**





\*\*\*\*\*

' wird Ausgeführt wenn ein Name aus der DropDown ausgewählt

' wurde

\*\*\*\*\*

**Sub AuswahlName()**

**Dim** li\_Zeile **As Integer**

**Dim** lo\_Eingabe **As Object**

li\_Zeile = Worksheets("Eingabe").DropDowns("dd\_name").Value

**Set** lo\_Eingabe = Worksheets("Eingabe")

**With** Worksheets("Empfänger")

lo\_Eingabe.Range("c3:c3").Value = .Cells(li\_Zeile; 2).Value

lo\_Eingabe.Range("c7:c7").Value = .Cells(li\_Zeile; 3).Value

lo\_Eingabe.Range("c5:c5").Value = .Cells(li\_Zeile; 4).Value

lo\_Eingabe.Range("h5:h5").Value = .Cells(li\_Zeile; 5).Value

lo\_Eingabe.Range("c10:c10").Value = .Cells(li\_Zeile; 6).Value

lo\_Eingabe.Range("c12:c12").Value = .Cells(li\_Zeile; 7).Value

lo\_Eingabe.Range("g8:g8").Value = 0

**End With**

**End Sub**



\*\*\*\*\*

' Diese Procedure wechselt in das Blatt Liste und  
' trägt die Summenformel ein

\*\*\*\*\*

**Sub** *ListeZeigen()*

**Dim** li\_Anzahl **As Integer**

**Dim** ls\_rel1; ls\_rel2 **As String**

**Dim** lo\_zelle **As Object**

Worksheets("Liste").Activate

li\_Anzahl = Worksheets("HilfsTabelle").Range("B2").Value

ls\_rel1 = Cells(3; 7). \_

    Address(rowabsolute:=False; \_

    columnabsolute:=False; \_

    ReferenceStyle:=xlA1)

ls\_rel2 = Cells(li\_Anzahl + 2; 7). Address(rowabsolute:=False; columnabsolute:=False; \_

    ReferenceStyle:=xlA1)

**With** Cells(li\_Anzahl + 3; 7)

    .Formula = "=SUM(" & ls\_rel1 & ":" & ls\_rel2 & ")"

    .NumberFormat = "#.##0,00 DM"

**End With**

**Set** lo\_zelle = Cells(li\_Anzahl + 3; 6)

**With** lo\_zelle

    .FormulaR1C1 = "Gesamt:"



```
.HorizontalAlignment = xlRight  
.VerticalAlignment = xlBottom  
.WrapText = False  
.Orientation = xlHorizontal
```

End With

End Sub



\*\*\*\*\*

' Löscht einen Eintrag aus der Aktuellen Liste

\*\*\*\*\*

**Sub** *Liste\_EintragLöschen()*

**Dim** li\_Anzahl **As Integer**

li\_Anzahl = Worksheets("HilfsTabelle").Range("B2").Value

**If** li\_Anzahl > 0 **Then**

    Selection.EntireRow.Delete

    Worksheets("HilfsTabelle").Range("B2").Value = Worksheets("HilfsTabelle").Range("B2").Value - 1

**End If**

**End Sub**



\*\*\*\*\*

' Initialisiert die Eingabefelder

\*\*\*\*\*

**Sub** *EingabeFelder\_Init()*

**With** Worksheets("Eingabe")

.Range("c3").Value = ""

.Range("c5").Value = ""

.Range("h5").Value = ""

.Range("c7").Value = ""

.Range("c10").Value = ""

.Range("c12").Value = ""

.DropDowns("dd\_name").Value = 0

.DropDowns("dd\_arten").Value = 0

**End With**

**End Sub**



\*\*\*\*\*

' Alle Listeneinträge löschen

\*\*\*\*\*

**Sub Programm\_Init()**

**Dim** li\_Anzahl **As Integer**

'Hole die aktuelle Anzahl der Überweisungen

'und lösche die Einträge

li\_Anzahl = Worksheets("HilfsTabelle").Range("B2").Value

**For** Zähler = 1 **To** li\_Anzahl + 7

    Worksheets("Liste").Rows(3).Delete Shift:=xlUp

**Next**

EingabeFelder\_Init

**End Sub**



\*\*\*\*\*

' Aufrufe der Tabellenblätter über Makros

\*\*\*\*\*

**Sub Dialog\_info()**

DialogSheets("info").Show

End Sub

**Sub EingabeMaske()**

Worksheets("Eingabe").Activate

End Sub

**Sub History\_Zeigen()**

Worksheets("History").Activate

End Sub

**Sub TB\_Wechseln()**

If ActiveSheet.Name = "Liste" Then

ListeZeigen

End If

End Sub



\*\*\*\*\*

' VBA Drucker

\*\*\*\*\*

**Sub StandardDrucker\_Einstellen()**

' an dieser Stelle muß der Name des Standarddruckers eingetragen werden

'Beispiele :

'Application.ActivePrinter = "HP LaserJet 4P/4MP auf LPT3:"

'Application.ActivePrinter = "HP LaserJet 4 Plus auf drucker\_4516@personal@spandau"

End Sub

**Sub FormularDrucker\_Einstellen()**

' an dieser Stelle muß der Name des Formulardruckers eingetragen werden

'Beispiel :

'Application.ActivePrinter = "NEC Pinwriter P6\_1 auf LPT1:"

End Sub





\*\*\*\*\*

' Einstellen des Druckbereiches und Benutzerdefiniertes

' Papierformat (wird eigentlich nicht benötigt)

\*\*\*\*\*

**Sub FormularDruckRangeEinstellen()**

FormularDrucker\_Einstellen

**With** Worksheets("Formular").PageSetup

.PrintTitleColumns = ""

.LeftFooter = ""

.CenterFooter = ""

.RightFooter = ""

**'Benutzerdefiniertes Papierformat einstellen**

.PaperSize = xlPaperUser

.Zoom = 100

**End With**

StandardDrucker\_Einstellen

**End Sub**



\*\*\*\*\*

'Diese Procedure aktiviert den Formulardruck  
' nach Druck werden die Einträge in das Tabellenblatt Liste  
' sowie History übernommen

\*\*\*\*\*

**Sub** *Formular\_Drucken()*

**Dim** li\_Zeile **As Integer**

**Dim** lo\_dpdw; lo\_Eingabe; lo\_Stamm **As Object**

**Set** lo\_Eingabe = Worksheets("Eingabe")

**Set** lo\_dpdw = lo\_Eingabe.DropDowns("dd\_arten")

**Set** lo\_Stamm = Worksheets("Empfänger")

Worksheets("HilfsTabelle").Range("AnzAktuell").Value = \_  
    Worksheets("HilfsTabelle").Range("AnzAktuell").Value + 1

Worksheets("HilfsTabelle").Range("AnzGesamt").Value = \_  
    Worksheets("HilfsTabelle").Range("AnzGesamt").Value + 1

'Worksheets("Formular").Activate

'FormularDrucker\_Einstellen

'Worksheets("Formular").PrintOut

'StandardDrucker\_Einstellen

' Eintragen der Überweisung in eine Liste

li\_Zeile = Worksheets("HilfsTabelle").Range("B2").Value + 2



With Worksheets("Liste")

```
.Cells(li_Zeile; 1).Value = lo_Eingabe.Range("c3:c3").Value  
.Cells(li_Zeile; 2).Value = lo_Eingabe.Range("c7:c7").Value  
.Cells(li_Zeile; 3).Value = lo_Eingabe.Range("c5:c5").Value  
.Cells(li_Zeile; 4).Value = lo_Eingabe.Range("h5:h5").Value  
.Cells(li_Zeile; 5).Value = lo_Eingabe.Range("c10:c10").Value  
.Cells(li_Zeile; 6).Value = lo_Eingabe.Range("c12:c12").Value  
.Cells(li_Zeile; 7).Value = lo_Eingabe.Range("g8:g8").Value
```

End With

```
li_Zeile = Worksheets("HilfsTabelle").Range("AnzGesamt").Value + 2
```

With Worksheets("History")

```
.Cells(li_Zeile; 1).Value = Now  
.Cells(li_Zeile; 2).Value = lo_Eingabe.Range("c3:c3").Value  
.Cells(li_Zeile; 3).Value = lo_Eingabe.Range("c7:c7").Value  
.Cells(li_Zeile; 4).Value = lo_Eingabe.Range("c5:c5").Value  
.Cells(li_Zeile; 5).Value = lo_Eingabe.Range("h5:h5").Value  
.Cells(li_Zeile; 6).Value = lo_Eingabe.Range("c10:c10").Value  
.Cells(li_Zeile; 7).Value = lo_Eingabe.Range("c12:c12").Value  
.Cells(li_Zeile; 8).Value = lo_Eingabe.Range("g8:g8").Value  
.Cells(li_Zeile; 9).Value = Worksheets("Hilfstabelle").Cells(lo_dpdw.Value + 1; 1).Value
```

End With

```
lo_Eingabe.Activate
```

End Sub

```
*****
```



' Ausdruck der Aktuellen ÜberweisungsListe  
' Einstellen des Druckbereiches/Kopf und Fusszeile  
' über VBA Makros

\*\*\*\*\*

**Sub** *ListeDruck()*

**Dim** li\_Anzahl **As Integer**

StandardDrucker\_Einstellen

Worksheets("Liste").Activate

li\_Anzahl = Worksheets("HilfsTabelle").Range("AnzAktuell").Value + 2

Range(Cells(3; 1); Cells(li\_Anzahl + 7; 7)).Select

With Worksheets("Liste").PageSetup

.PrintTitleRows = "\$2:\$2"

.PrintArea = Selection.Address

.Orientation = xlQuerformat

.PrintTitleColumns = ""

.LeftFooter = "&8" & Application.UserName & " " & ActiveWorkbook.Path & \_  
"\&N" & " [" & Date & "/" & Time & "]" "

.CenterFooter = ""

.RightFooter = "&8&S von &A"

**End With**

Worksheets("Liste").PrintOut

**End Sub**