

Microsoft Access VBA による CSV データの作成

Making of CSV data by Microsoft Access VBA

渡 邊 真 人

Masato Watanabe

はじめに

Microsoft Access は、Microsoft 社のデータベースソフトである。VBA (Visual Basic for Applications) は、この Microsoft Access や Microsoft Word、Microsoft Excel など、Microsoft 社の Office 製品群に共通のプログラミング言語である。

現在、Microsoft Excel VBA によるテキストデータからのグラフ作成を実装したアンケート集計システムの開発、運用を行っている。先の近畿大学豊岡短期大学論集第3号では、開発したアンケート集計システムのうちテキストデータから Excel のワークシートへ自動展開する部分について Microsoft Excel VBA のコードとシステムの仕様を中心に報告した¹⁾。このアンケート集計システムは、データの入力を簡略化するために Microsoft Access により入力フォームを作成している。入力されたデータは、Microsoft Access の VBA を用いて、フィールド毎に半角カンマで区切った CSV 形式にデータを成形している。今回は、この Microsoft Access のテーブル内のデータから CSV データを作成する部分の VBA コードとプログラムの仕様を中心に報告する。

処理の方法

開発に使用した Microsoft Access のバージョンは、Microsoft Access 2002 (10.6501.6735) SP3、VB Editor のバージョンは、Microsoft Visual Basic 6.0 である。

入力データは、図-1 のようなアンケートの記入用紙から、図-2 のようなデータシートに入力する。アンケートデータを入力する前にアンケートデータの設題番号、表示番号、アンケートの部課別に人数を設定して、入力件数分のレコードを自動作成する。

データの入力は、この自動作成したレコードをデータシートに表示して、記入用紙より選択された回答の番号を手入力していく。

データシートからの入力データは、データ入力テーブルに格納される。入力終了後、このデータ入力テーブルから、フィールド毎に半角カンマで区切った CSV 形式のテキストデータを自動的に作成する。

アンケート記入用紙 No. 3

アンケート記入用紙 No. 2

アンケート記入用紙 No. 1

部	
課	
回答番号	
設題1	
設題2-(1)	
設題2-(2)	
設題3	
設題4	
設題5	

図-1 アンケートの記入用紙

部	課	連続番号	表示番号	回答番号01	回答番号02	回答番号03	回答番号04	回答番号05
1	1	1	設題1	1				
1	1	1	設題2-(1)	2				
1	1	1	設題2-(2)	3				
1	1	1	設題3	4	5	6		
1	1	1	設題4			7		
1	1	1	設題5			8		
1	1	2	設題1			9		
1	1	2	設題2-(1)			11		
1	1	2	設題2-(2)			13		
1	1	2	設題3					
1	1	2	設題4					
1	1	2	設題5					
1	1	3	設題1					
1	1	3	設題2-(1)					
1	1	3	設題2-(2)					
1	1	3	設題3					
1	1	3	設題4					
1	1	3	設題5					
1	1	4	設題1					
1	1	4	設題2-(1)					
1	1	4	設題2-(2)					
1	1	4	設題3					
1	1	4	設題4					
1	1	4	設題5					
1	1	5	設題1					
1	1	5	設題2-(1)					
1	1	5	設題2-(2)					
1	1	5	設題3					
1	1	5	設題4					
1	1	5	設題5					

図-2 データシート

Microsoft Access は、VBA を用いて、データベースを操作するとき、ADO (ActiveX Data Objects) や DAO (Data Access Objects) を使用する。今回、データベースへの接続に DAO を使用している。その際、Microsoft Access 2002では、モジュールの参照設定の既定値が「Microsoft ActiveX Data

Objects 2.1 Library」となっているので、DAOで接続する場合、メニューバーのツールから、図-3のように「参照設定」を選択して、「Microsoft DAO 3.6 Object Library」の指定が必要となる。



図-3 参照設定

1. 入力データの仕様

テーブルのデザインは、表-1のとおりである。テーブル名は、データ入力テーブル、主キーは、重複無しの連番で長整数型とする。部、課、連続番号、設題番号は、最大選択番号、開始行は、整数型とし、表示番号、回答番号はテキスト型とする。

表-1 データ入力テーブル

フィールド名	データ型	フィールドサイズ
主キー	数値型	長整数型
部	数値型	整数型
課	数値型	整数型
連続番号	数値型	整数型
設題番号	数値型	整数型
表示番号	テキスト型	20
回答番号01	テキスト型	4
回答番号02	テキスト型	4
回答番号03	テキスト型	4

回答番号04	テキスト型	4
回答番号05	テキスト型	4
回答番号06	テキスト型	4
回答番号07	テキスト型	4
回答番号08	テキスト型	4
回答番号09	テキスト型	4
回答番号10	テキスト型	4
最大選択番号	数値型	整数型
開始行	数値型	整数型

2. 出力データの仕様

出力データの仕様および出力レコードの例は、それぞれ、表-2、表-3のとおりである。レコード先頭より、「部,課,連続番号,設題番号,表示番号,回答番号1,2,3,4,5,最大選択番号,開始行」の順で、フィールドの区切りはカンマ、レコードの終了は改行でシーケンシャルファイルとし、データベースファイルがあるフォルダと同じフォルダ内へ出力する。ファイル名は、test.txt とする。

表-2 出力データフィールド一覧

フィールド名	フィールドタイプ
部	数字2桁
課	数字2桁
連続番号	数字4桁
設題番号	数字4桁
表示番号	文字20桁
回答番号1	数字2桁
回答番号2	数字2桁
回答番号3	数字2桁
回答番号4	数字2桁
回答番号5	数字2桁
最大選択番号	数字2桁
開始行	数字2桁

表-3 出力レコード例

```

1, 1, 1, 1, 設題 1, 1, ..., 3, 5
1, 1, 1, 2, 設題 2-(1), 1, ..., 3, 12
1, 1, 1, 3, 設題 2-(2), 1, ..., 3, 18
1, 1, 1, 4, 設題 3, 1, ..., 4, 24
1, 1, 1, 5, 設題 4, 1, ..., 4, 31
1, 1, 1, 6, 設題 5, 1, ..., 5, 38
1, 1, 2, 1, 設題 1, 2, ..., 3, 5
1, 1, 2, 2, 設題 2-(1), 2, ..., 3, 12
1, 1, 2, 3, 設題 2-(2), 2, ..., 3, 18
1, 1, 2, 4, 設題 3, 2, ..., 4, 24
1, 1, 2, 5, 設題 4, 2, ..., 4, 31
1, 1, 2, 6, 設題 5, 2, ..., 5, 38
1, 1, 3, 1, 設題 1, 3, ..., 3, 5
1, 1, 3, 2, 設題 2-(1), 3, ..., 3, 12
1, 1, 3, 3, 設題 2-(2), 3, ..., 3, 18
1, 1, 3, 4, 設題 3, 3, ..., 4, 24
1, 1, 3, 5, 設題 4, 3, ..., 4, 31
1, 1, 3, 6, 設題 5, 3, ..., 5, 38
1, 1, 4, 1, 設題 1, 1, ..., 3, 5
1, 1, 4, 2, 設題 2-(1), 1, ..., 3, 12
1, 1, 4, 3, 設題 2-(2), 1, ..., 3, 18
1, 1, 4, 4, 設題 3, 4, ..., 4, 24
1, 1, 4, 5, 設題 4, 4, ..., 4, 31
1, 1, 4, 6, 設題 5, 4, ..., 5, 38

```

処理の概要

アンケート集計システムの処理の概要は、図-4 のとおりである。

データ入力テーブルから、カンマ区切りの CSV データ (test.txt) を作成する図-4 の点線で囲んだ部分が、Access で処理した部分である。アンケートの回答を Access のフォームで作成したデータシートに入力後、CSV データの作成を自動で行う。

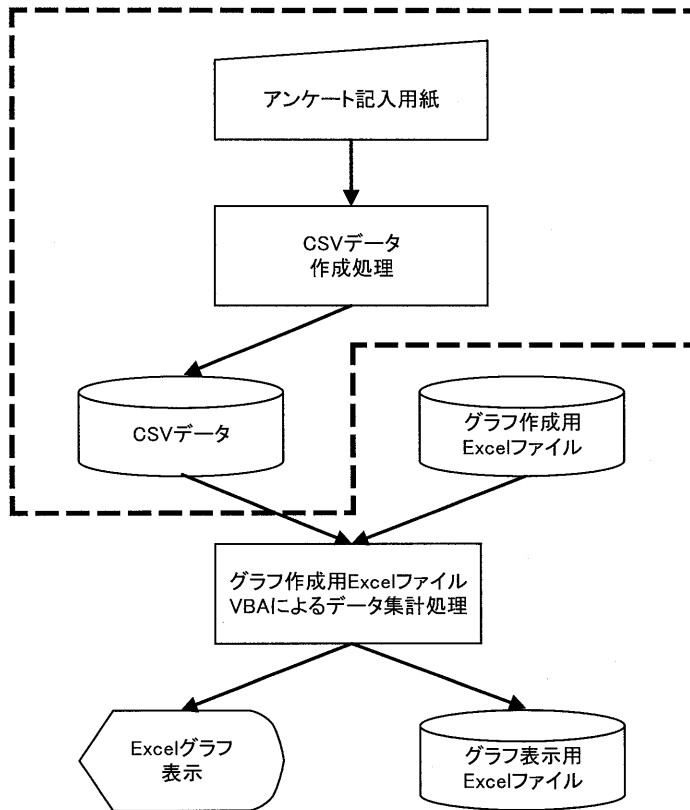


図-4 処理の概要

システムの考察

カンマ区切りの CSV データを手入力で作成することは、たいへん煩雑な作業で入力の間違いも入りやすい。そこで、アンケートデータを記入表の様式で入力できるように、Access のフォームでデータシートを準備し、データ入力後、VBA により CSV データの出力を自動化することとした。

次ページに処理コードを示す。

文字列の比較方法は、モジュールの宣言セクションに、Option Compare Database ステートメントを記述して、Database モードに設定している。Microsoft Visual Basic ヘルプには、Option Compare Database モードは、「Microsoft Access でのみ使用でき、このモードの文字列比較の並べ替え順序は、

データベースの文字列比較に適用される国別の ID によって決まる」とある。²⁾

```

'文字列比較方法を Database モードに設定する
Option Compare Database
'モジュール内のすべての変数に対して、明示的な宣言を強制する
Option Explicit
Public Function MyTxt_Out() As Integer
'Access2000以降、モジュールの参照設定の既定値がADO(ActiveX Data Object)
'となったのでDAO(Data Access Object)で参照する場合次の設定が必要
'メニューバーのツール→参照設定
'Microsoft DAO 3.6 Object Library をチェック

Dim UsDB           As DAO.Database
Dim UsDataInRs     As DAO.Recordset
Dim UsFileNo       As Integer

    MyTxt_Out = 0

'カレントデータベースに接続する
'CurrentDb 関数の戻り値を Database オブジェクトに代入する
    Set UsDB = CurrentDb()
'テーブルを参照する
'テーブルタイプの Recordset オブジェクトを作成する
    Set UsDataInRs = UsDB.OpenRecordset("データ入力テーブル", dbOpenTable)
    UsDataInRs.Index = "PRIMARYKEY"
    If UsDataInRs.EOF Then
        MsgBox ("データ入力テーブルにデータがありません。")
        UsDataInRs.Close
        UsDB.Close
        Exit Function
    End If

    UsDataInRs.MoveLast

    Debug.Print "データ入力テーブルレコード件数:" & UsDataInRs.RecordCount & "件"

    If UsDataInRs.RecordCount = 0 Then
        MsgBox ("データ入力テーブルのデータが0件です。")
        UsDataInRs.Close
        UsDB.Close
        Exit Function
    End If

    UsDataInRs.MoveFirst

    UsFileNo = FreeFile
    Open Application.CurrentProject.Path & "test.txt" For Output As #UsFileNo

    Do Until UsDataInRs.EOF
        Print #UsFileNo, UsDataInRs[部] & "," & UsDataInRs[課] _
            & "," & UsDataInRs[連続番号] & "," & UsDataInRs[設題番号] _
            & "," & Trim(UsDataInRs[表示番号]) _
            & "," & Trim(UsDataInRs[回答番号 01]) _
            & "," & Trim(UsDataInRs[回答番号 02]) _
            & "," & Trim(UsDataInRs[回答番号 03]) _
            & "," & Trim(UsDataInRs[回答番号 04]) _
            & "," & Trim(UsDataInRs[回答番号 05]) _
            & "," & UsDataInRs[最大選択番号] & "," & UsDataInRs[開始行]
        UsDataInRs.MoveNext
    Loop

    Close #UsFileNo

    UsDataInRs.Close
    UsDB.Close

    MyTxt_Out = 1

End Function

```

Visual Basic では、宣言していない変数を使用してもエラーにならない。プログラムの中に新しい変数名を見つけると、自動的にバリエーション型の変数として解釈し、処理を行ってしまうため変数名を間違ったときにミスを見つけにくくなる。モジュールの宣言セクションで Option Explicit ステートメントを記述しておく、宣言していない変数を使うとコンパイルエラーとなるので、ミスをみつけやすくなる。そこで、プログラムの先頭で、Option Explicit を宣言して、プログラム内で使用する変数は、すべて使用前に宣言している。

Access 2002では、1つのデータベースで ADO と DAO の両方を使用することができる。どちらのオブジェクトモデルにも Recordset オブジェクトが含まれている。そこで、オブジェクト変数の宣言では、ADO と DAO のどちらを宣言したか明確にするために Dim UsDataInRs As DAO.Recordset のようにオブジェクトを提供するコンポーネントの名前を付けて宣言する。

データベースに接続するには、DAO では database オブジェクトを利用する。カレントデータベースに接続するには、Dim UsDB As DAO.Database のように DAO の Database オブジェクトを宣言して、Set UsDB = CurrentDb()で、CurrentDb 関数の戻り値を Database オブジェクトに代入する。

Microsoft Access のテーブルに登録されているデータは、Recordset オブジェクトで管理される。OpenRecordset メソッドにより、「データ入力テーブル」のレコードセットを取得する。Set UsDataInRs = UsDB.OpenRecordset("データ入力テーブル", dbOpenTable)で、テーブルタイプの Recordset オブジェクトを作成している。テーブルタイプの Recordset オブジェクトは、新規にレコードを追加、更新ができる。³⁾

RecordCount プロパティを使用して、Recordset オブジェクト内のアクセス済みのレコード数を調べることができるが、すべてのレコードがアクセス済みになるまで、そのオブジェクトに含まれている合計レコード数は RecordCount プロパティに示されない。最後のレコードをアクセス済みにするため、UsDataInRs.MoveLast のように、Recordset オブジェクトに対して MoveLast メソッドを使用して、「データ入力テーブル」のレコード件数をチェックしている。「データ入力テーブル」のレコード件数が0件のときは、メッセージを表示して終了する。

Open Application.CurrentProject.Path & "¥test.txt" For Output As #UsFileNo により CSV データの出力先を指定している。Access のデータベースファイルが保存されている Path を取得して、同じフォルダに test.txt のファイル名で出力するよう指定している。

CSV データの作成は、Do...Loop ステートメントによる繰り返しの中で読み込んだ「データ入力テーブル」のレコードから、それぞれのフィールドに","を連結して Print # ステートメントによりシーケンシャル出力している。

おわりに

近畿大学豊岡短期大学において、前年に引き続いて、平成18年度の「学生生活の実態を調査するためのアンケート」の集計およびグラフ作成についての依頼があった。アンケートの記入表からの

データ入力、今回の報告にあるような Microsoft Access で作成したデータシート形式の入力フォームを利用している。また、データ入力後、Microsoft Access VBA による CSV 形式のテキストデータ作成から Microsoft Excel VBA によるテキストデータからのグラフ作成まで処理の流れは、本アンケートシステムを利用している。

参考文献

- 1) 渡邊真人：Microsoft Excel マクロによるテキストデータからのグラフ作成、近畿大学豊岡短期大学論集、3、59-69、2006.
- 2) Microsoft Corp：Microsoft Visual Basic ヘルプ、Microsoft Corp、2003.
- 3) 谷尻かおり：Access VBA 応用プログラミング、412-421、技術評論社、2000.