

## 〔情報処理研究〕

## VISUAL BASICのプログラミング法 3

## - ファイルの操作；読み込みと表示の基礎 -

門田 幸太郎\*

本稿では、データ・ベースの利用や作成の基礎となるファイルの操作方法を中心にVISUAL BASIC<sup>1</sup>のプログラミングについて述べた。ファイルの操作としては、ファイルの作成、読み込み、表示、検索などの方法の基礎について取り上げた。ファイルの作成方法では、保存方法や変数型の違いについても述べた。ファイルの読み込み方法では、Open、Input、Close文、EOF関数などについて述べた。ファイルの表示方法では、フォーム<sup>2</sup>にデータを表示する方法やデータ・ファイルの変数を対応するテキスト・ボックスに表示する方法、データを順次表示していく方法、以前に表示したデータをさかのぼって表示する方法などについて述べた。

キーワード：VISUAL BASIC，プログラミング，ファイル，日付(Date)型変数，Open，Input，Close，EOF，With，MsgBox，Format

## 目次

- 1 ファイル・データのプリント表示
- 2 ファイル・データのテキスト・ボックスへの表示
- 3 ファイル・データのテキスト・ボックスへの表示 逆方向検索を含む

## はじめに

筆者は、門田（1998）において、VISUAL BASICの特徴とプログラミングの基礎について述べた。また、門田（1999）では、配列データの操作について述べた。ここでは、VISUAL BASICによるデータベース作成に向けて、ファイルの操作法、とくにデータ・ファイルの読み込みと表示方法の基礎について、具体的課題を通して述べる。

## 1 ファイル・データのプリント表示

【例題1】あらかじめ、《参考》で示したようなファイルを作り、それをデータ・ファイルとして読み込み、表示せよ。

---

\* 立命館大学産業社会学部教授

1 ソフトはMicrosoft Visual Basic 5.0 Pro-fessional Editionを用い、ハードはNEC系Pentium機を用いた。

2 基本的なファイル操作法や用語については、門田(1998、1999)を参照。

《参考》d:\Monden¥sample.txt

山田吾一,30,#65-12-09#

岡田登,33,#63-06-14#

権田圭志郎,38,#58-04-02#

有沢克己,40,#58-05-05#

乃木静子,31,#65-11-03#

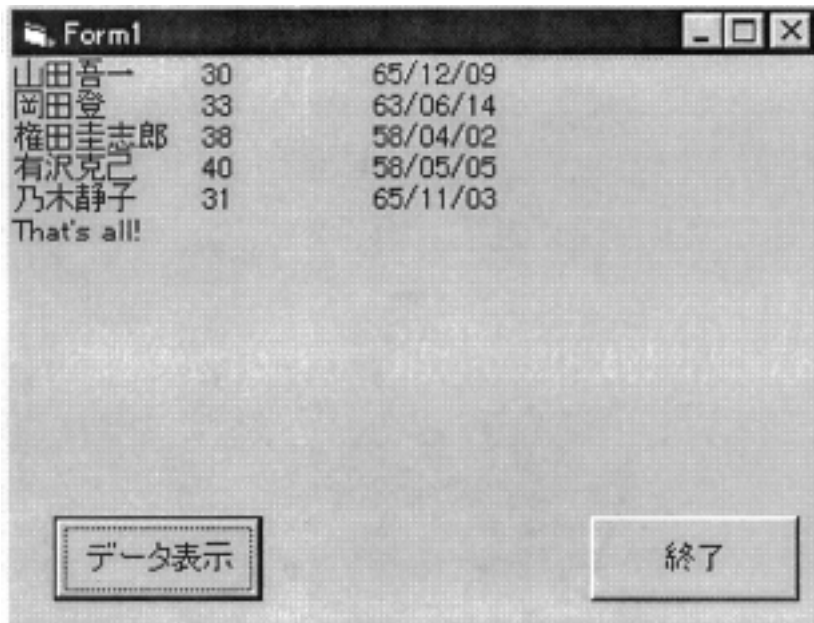


図1 例題1の実行例

ツール・ボックスからフォーム上に2つのコマンド・ボタンを取り込み、実行例のように配置する。Command1のキャプションを「データ表示」とし、Command2のキャプションを「終了」とする。各オブジェクトのプロシージャを次のように定義する。

[ プログラム例 ]

```
Private Sub Command1_Click()
```

```
    Dim dat1 As String
```

```
    Dim dat2 As Long
```

```
    Dim dat3 As Date
```

①

```
    Open "d:\Monden¥sample.txt" For Input As #1
```

②

```
Top:
```

③

```
    If Not EOF(1) Then
```

④

```

Input #1, dat1, dat2, dat3           ⑤
Print dat1, dat2, dat3
GoTo Top                             ⑥
Else
Close #1                             ⑦
Print "That's all!"
End If                                ⑧
End Sub

Private Sub Command2_Click()
End
End Sub

```

プログラミングに先立ち、《参考》で示したようなsample.txt ファイルを作る。d:\Monden\sample.txt というパス名は、dドライブのルート・ディレクトリにあるMondenというサブ・ディレクトリの中に含まれているsample.txtを指定している。サブ・ディレクトリ構造は関連する複数のファイルを一括して管理するためのものである。ドライブ名やサブ・ディレクトリ名はシステムに応じて記述しなければならない。sample.txtの「.txt」はこのファイルが文字(テキスト)型データであることを示す拡張子である。その内容は、氏名、年齢、誕生日を各個人ごとに入力した5人分の架空データである。誕生日のデータは「#」で囲まれている。これは、日付(date)型データであることを示すものである。ファイルの作成にあたっては、「MSWORD」や「一太郎」などのワープロ・ソフトやWINDOWSの「アクセサリ」内の「メモ帳」、「ワードパッド」などを利用することができる。ただし、作成したファイルを保存する場合、テキスト・ファイルとして保存する必要がある。

Command1\_Click()は「データ表示」のコマンド・ボタンのクリックに対応するプロシージャである。これにより、ファイルからデータが読み込まれ、表示される。入力用としてファイルを開き、データがあるかぎりそれを取り込んで表示する。データがなくなれば、「That's all!」と表示する。このプロシージャの冒頭部分において、「Dim dat1 As String」によりdat1を文字型変数として、「Dim dat2 As Long」によりdat2を長整数型変数として宣言している。①では変数名dat3を日付型データとして定義している。dat1は名前用として、dat2は年齢用として、dat3は誕生日用として宣言されている。②、⑤、⑦はファイル操作に関する文(ステートメント)で、シーケンシャル・アクセスのインプット・モードといわれるものである。これにより、指定されたファイルからデータが取り込まれる。構文としては、一般に

```

Open パス名 For Input As #ファイル番号
Input #ファイル番号, 変数名リスト
Close #ファイル番号

```

という形式で用いられる。パス名は，《参考》のd:\Monden¥sample.txtのように、目的のファイルを含むパス名を指定する。このパス名で指定されたファイルを「#」に続いて指定されたファイル番号に割り振る。②での設定以後、ファイルの特定は、このファイル番号によって行われることになる。変数名リストとは「変数名，変数名，…，変数名」というように、一連の変数名が列記されたものを指す。⑤では，dat1，dat2，dat3の3変数が，先ほどのファイル番号で特定されたファイルから取り込まれる。⑦は，②で開いたファイルを閉じるためのものである。

③はラベルといわれ，プログラムの制御のために用いられる。この③は，⑥の「Go To Top」に対応している。プログラムの実行，i.e.フロー制御が⑥に達した場合は，かならず に戻ることを意味している。④から⑧までが，一連のIf文である。④のEOF(1)は（ ）内のファイル番号のファイル・データが最後（End Of File）になったら，Trueを返す関数である。ここでは，Not EOF(1)はファイル番号1のデータが最後でないという条件式が真であるかぎり，つまり，データがあるかぎり，⑤から⑥までの手続きが行われる。Not EOF(1)の条件式が成り立たない場合，つまり，ファイル番号1のデータが最後になった場合，Elseから⑧までが実行される。⑤はdat1，dat2，dat3の3変数を，②で指定されたファイル番号のファイルから読み込むためのものである。たとえば，sample.txtの最初の個人データは「山田吾一,30,#65-12-09#」である。「山田吾一」がdat1に，「30」がdat2に，「#65-12-09#」がdat3に読み込まれることになる。⑦は，②で開いたファイルを閉じるためのものである。ファイル操作では，ファイルを開く操作に対して，かならずファイルを閉じる操作が伴う。

Command2\_Click()は終了のためのプロシージャである。「終了」のコマンド・ボタンをクリックするとプログラムは終了する。

以上のようにプロシージャを記述することにより，あらかじめ作成されたテキスト・ファイルを読み込み，表示することができるようになる。プログラムの動きとしては，「データ表示」のコマンド・ボタンをクリックすることによって，指定されたファイルを開き，データを読み込み，表示する。読み込むべきデータがなくなればファイルは閉じられる。プログラムの終了は「終了」のコマンド・ボタンで行われる。

実行例はプログラムを実行した後に「データ表示」のコマンド・ボタンをクリックすることによって，フォームにデータの内容が表示された状態である。

## 2 ファイル・データのテキスト・ボックスへの表示

【例題2】例題1において，データを項目別にテキスト・ボックスに表示せよ。終了直前に終了操作を促すメッセージ・ボックスを下図のように表示せよ。

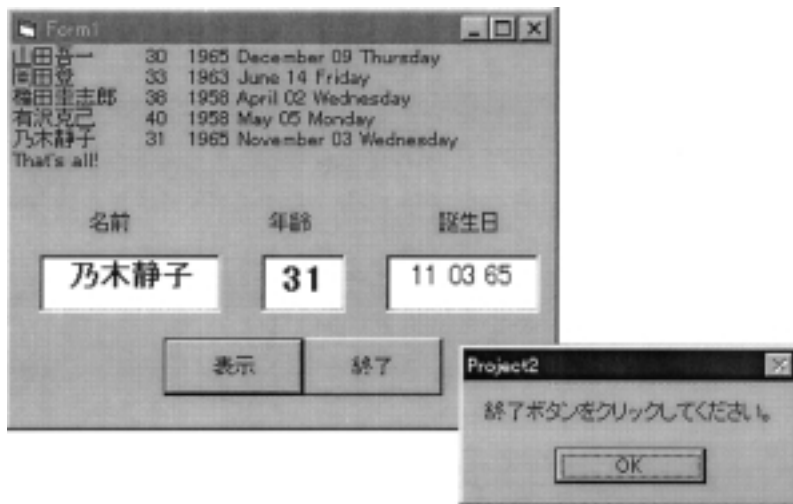


図2 例題2の実行例

ツール・ボックスからフォーム上に2つのコマンド・ボタンと3つのテキスト・ボックス、3つのラベルを取り込み、実行例のように配置する。コマンド・ボタンのキャプションについては、Command1を「表示」とし、Command2を「終了」とする。ラベルのキャプションについては、Label1を「名前」とし、Label2を「年齢」、Label3を「誕生日」と設定する。テキスト・ボックスのTextプロパティはText1、Text2、Text3の3つともプロパティ・ウィンドウにおいて、消去しておく。各オブジェクトのプロシージャは次のようにコード記述される。

## [ プログラム例 ]

```
Private Sub Form_Load() ①
    Dim dat1 As String
    Dim dat2 As Long
    Dim dat3 As Date
    Open "d:\Monden\sample.txt" For Input As #1
End Sub

Private Sub Command1_Click()
    If Not EOF(1) Then
        Input #1, dat1, dat2, dat3
        With Text1 ②
            .FontSize = 13 ③
            With .Font ④
```

```

        .Bold = True                                ⑤
    End With                                       ⑥
End With                                         ⑦
Print dat1, Format(dat2, "   ###   "); Format (dat3, "yyyy mmmm dd dddd") ⑧
Text1.Text = dat1                                ⑨
Text2.Text = dat2                                ⑩
Text3.Text = Format(dat3, " mm dd yy ")           ⑪
With Text2                                       ⑫
    .FontSize = 16
    .Font.Bold = True
End With                                         ⑬
With Text3                                       ⑭
    .FontSize = 12
    .Font.Bold = False
End With                                         ⑮
Else
    Close #1
    Print "That's all!"
    MsgBox "終了ボタンをクリックしてください。", vbYes ⑯
End If
End Sub

Private Sub Command2_Click()
    End
End Sub

```

①のForm\_Load()はプログラムの実行に伴って稼働するプロシージャである。VISUAL BASICではイベントに対応して設定されたプロシージャが稼働するようになっている。これをイベント・ドリブンという。一般に、このイベントはユーザーによるマウスやキーボード操作などによって引き起こされるものである。一方、プログラムが実行されるということはフォームがロードされることを意味する。このフォームのロード自体は実行に伴うものであり、ユーザーによるイベントではないが、これも一つのイベントとして扱うことができる。例題1では、「データ表示」のコマンド・ボタンをクリックすることによって、テキスト・データが読み込まれ、表示された。しかし、最初から一定の手続きが想定される場合には、プログラムの実行に伴うフォームのロードに対するプロシージャを記述しておくことができる。これによって、ユーザーからの反応を待たずして、一定のプロシージャを実行さ

せることができる。ここでは、プログラムの実行に伴い、sample.txtが開かれる。

Command1\_Click()は「表示」のコマンド・ボタンのクリックに対応するプロシージャである。これにより、「表示」ボタンがクリックされるたびに、sample.txtからデータが読み込まれ、個々人ごとのデータの内容が、フォーム上には3変数が、テキスト・ボックスには3項目がそれぞれ表示されるようになる。②のWith文から⑦のEnd With文までは、このテキスト・ボックスに表示する場合のプロパティを設定している。③は、Text1のフォント・サイズを13とするものである。④から⑥はText1のフォントに限定して、そのプロパティをさらに設定するためのものである。ここでは、⑤でフォントをBold(太字)に設定する。この設定は⑫から⑬のような形式で記述することもできる。特定のプロパティをまとめて設定する場合は、④から⑥のようにすることができる。

⑧はFormat関数を伴ったPrint文である。これによって、dat1、dat2、dat3が表示される。dat1については、データの内容がそのまま表示される。dat1とdat2との間には「,」が付けられている。これによって、dat2は一定間隔をおいて表示される。dat2については、Format関数により" ### "という形式に変換されて表示される。年齢のデータを「#」の数だけ桁数を使って表示することになる。ここでは2桁の数字を3桁で表示している。「###」の前後の空白が半角で2つと3つある。dat3については、「yyyy mmmm dd dddd」という形式に変換されて表示される。「yyyy」は年を4桁の数字で表示させることを意味する。「yy」は年の下2桁を表示させる。「mmmm」は月をアルファベットで省略せずに表示させることを意味する。「mmm」は月をアルファベットの省略形で表示させる。「mm」は月を2桁の数字で表示させる。「dddd」は曜日をアルファベットで省略せずに表示させることを意味する。「ddd」は曜日をアルファベットの省略形で表示させる。「dd」は日を2桁の数字で表示させる。ここでは、実行例のフォーム上に表示されているように、「65」という年を「yyyy」で「1965」と4桁で表示し、「12」という月を「mmmm」で「December」というようにアルファベットで省略せずに表示し、さらに、日を「dd」で「09」と2桁で表示し、最後に、曜日を「dddd」で「Thursday」とアルファベットで省略せずに表示している。ちなみに、曜日を「ddd」とすると、「Thur」とアルファベットの省略形で表示される。

⑨では名前が読み込まれているdat1の内容が、「名前」というキャプションの付いた左側のテキスト・ボックスText1に代入され、その結果、名前が表示されることになる。表示する場合の設定としては、②から⑦でプロパティ設定されている。⑩では年齢が読み込まれているdat2の内容が、「年齢」というキャプションの付いた中央のテキスト・ボックスText2に代入され、年齢が表示される。そのプロパティ設定は⑫から⑬までで、フォント・サイズが16で、Boldで表示される。⑪では誕生日が読み込まれているdat3の内容が、「誕生日」というキャプションの付いた右側のテキスト・ボックスText3に誕生日が表示される。表示形式としては、Format関数を用いて、月日年という順に2桁の数字で表示される。⑭から⑮では表示される数字のプロパティを設定する。⑭から⑮で、テキスト・ボックスの右端にある「誕生日」のText3のプロパティを、フォントのサイズが12で、太字を用いないものと設定する。ここで、「.Font.Bold = False」というのは、Boldで表示しないということの意味する。⑯はメッセージ・ボックスを表示する命令である。実行例にあるように、メッセージ・ボック

スに表示したい内容を「」で囲む。ここでは「終了ボタンをクリックしてください。」としている。

以上のようなプロシージャの記述により、データを項目別にテキスト・ボックスに表示し、終了直前に終了操作を促すメッセージ・ボックスを表示することができる。プログラムの実行とともに、Open文で指定されたファイルからデータを読み込む準備ができる。「表示」ボタンをクリックするたびにデータが1人分ずつ読み込まれ、フォーム上にデータの内容が表示されると同時に、3つのテキスト・ボックスのそれぞれに、対応するデータ内容が設定されたプロパティで表示される。

Command2\_Click()は「終了」のコマンド・ボタンに対応するプロシージャである。これをクリックすることにより、プログラムをいつでも終了させることができる。

実行例は、プログラムが実行されて、「表示」ボタンにより、データ内容がフォームとテキスト・ボックスに表示されるという操作が繰り返され、最後のデータまで、読み込まれて、メッセージ・ボックスが表示された状態を示している。この後、「OK」がクリックされ、さらに「終了」ボタンがクリックされることによりプログラムが終了することになっている。

### 3 ファイル・データのテキスト・ボックスへの表示 逆方向検索を含む

【例題3】例題1で用いたデータ・ファイルを読み込んで、データをテキスト・ボックスに順に表示せよ。また、データをさかのぼって表示することもできるようにせよ。

ツール・ボックスからフォーム上に4つのコマンド・ボタンと3つのテキスト・ボックス、3つのラベルを取り込み、実行例のように配置する。コマンド・ボタンのキャプションについては、Command1を「開始」とし、Command2を「終了」、Command3を「次へ」、Command4を「戻る」とする。ラベルのキャプションについては、例題2と同様に、Label1に「名前」、Label2に「年齢」、Label3に「誕生日」を設定する。各オブジェクトに対応するプロシージャを次のように記述する。

名前	年齢	誕生日
有沢克己	40	58/05/05

開始      戻る      次へ      終了

図3 例題3の実行例



## [ プログラム例 ]

```
Option Base 1
```

```
Dim data(1000, 3) ①
```

```
Public i, Last
```

```
Private Sub Command1_Click()
```

```
    Dim dat1 As String
```

```
    Dim dat2 As Long
```

```
    Dim dat3 As Date
```

```
    Open "d:\Monden\sample.txt" For Input As #1
```

```
    Last = 5
```

```
    For i = 1 To Last
```

```
        Input #1, dat1, dat2, dat3
```

```
        data(i, 1) = dat1 ②
```

```
        data(i, 2) = dat2 ③
```

```
        data(i, 3) = dat3 ④
```

```
    Next
```

```
    Close #1
```

```
    i = 0 ⑤
```

```
    Print "データの読み込みが完了!"
```

```
    MsgBox "「次へ」ボタンをクリックしてください。", vbOKOnly
```

```
    Cls
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
    End
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
    i = i + 1 ⑥
```

```
    If i > Last Then ⑦
```

```
        Print "No More Data"
```

```
        MsgBox "「戻る」か「終了」をクリックしてください。", vbOKOnly
```

```
        Cls
```

```
    Else
```

```

Text1.Text = data(i, 1)           ⑧
Text2.Text = data(i, 2)           ⑨
Text3.Text = data(i, 3)           ⑩
End If
End Sub

Private Sub Command4_Click()
    i = i - 1                       ⑪
    If i < 1 Then                     ⑫
        Print "First of Data"
        MsgBox "「次へ」ボタンをクリックしてください。", vbOKOnly
        i = i + 1                     ⑬
    Else
        Text1.Text = data(i, 1)       ⑭
        Text2.Text = data(i, 2)       ⑮
        Text3.Text = data(i, 3)       ⑯
    End If
End Sub

```

①の「Dim data(1000, 3)」は2次元の配列をバリエーション型(Variant)型変数として宣言するものである。一般に、変数の宣言は

```
Dim 変数名 As データ型
```

という形式でなされる。とくに、「As データ型」の部分を省略した場合は、変数の型はバリエーション型となる。バリエーション型はメモリー領域を16バイト消費する。これは、整数(Integer)型の2バイト、長整数(Long)型の4バイトと比べて、費やすメモリー量が大きいので、メモリーの利用効率はよくない。しかし、バリエーション型は、データが数値型であろうと、文字型であろうと格納することができるという便利さがある。

VISUAL BASICでは、原則として、変数はプロシージャごとに独立している。したがって、たとえ同じ変数名であっても、それが用いられているプロシージャが異なれば、その内容は異なっている。しかし、場合によっては、どのプロシージャにも共通した変数を用いることが必要となることがある。配列data()はCommand1\_Click()のプロシージャ内でも、Command3\_Click()のプロシージャ内でも、Command4\_Click()のプロシージャ内でも用いられる。また、iはCommand1\_Click()とCommand3\_Click()、Command4\_Click()のプロシージャ内で、LastはCommand1\_Click()とCommand4\_Click()のプロシージャ内で用いられる。このように複数のプロシージャで、同じ配列や変数を用いる場合は、最初に宣言しておく必要がある。

Command1\_Click()は「開始」のコマンド・ボタンに対応するプロシージャである。このプロシージャにより、ファイルを開いて、データを配列に読み込むことができる。読み込みが完了した時点で、メッセージ・ボックスで「次へ」ボタンをクリックしてください。」と表示することになる。まず、dat1を文字型、dat2を長整数型、dat3を日付型と宣言する。文字型宣言をすると、1文字に対して2バイトのメモリーが確保され、記憶すべき文字数に応じて、自動的にメモリー領域が拡大される。長整数型の場合、4バイトのメモリーが割り振られる。1バイトは8ビット、すなわち、2の8乗であるので、4バイトのメモリーには、2の8乗のさらに4乗である4294967296種類の数値を格納することができる。これを正と負とさらに0に分けると、-2147483648から0を含めて、2147483647までの数値を記憶させることができることになる。日付型変数の場合、8バイトのメモリー領域が使われる。これによって、西暦100年1月1日～西暦9999年12月31日までの日付と時刻を記憶させることができる。

次に、あらかじめ作成されたsample.txtからデータを読み込むべくファイルを開く。ここでは、データ数が5とわかっているので、「Last = 5」とする。For . . . Next文を使ってdat1, dat2, dat3の3変数を、先に宣言した変数型として読み込み、配列data()に代入する。②, ③, ④で文字型のdat1をdata(i, 1)に、長整数型のdat2をdata(i, 2)に、日付型のdat3をdata(i, 3)にそれぞれ代入する。配列への代入のために開いたファイルを「Close #1」で閉じる。For . . . Next文が終わった時点では、iはLastの値5となっている。⑤で、「i = 0」としたのは、⑥でテキスト・ボックスに配列data()を表示するという処理をする時に、iを初期化しておく必要があるからである。フォームに「データの読み込みが完了!」と表示し、メッセージ・ボックスに「次へ」ボタンをクリックしてください。」とする。画面をクリアにしてこのプロシージャは終わる。

Command2\_Click()は終了のためのプロシージャである。

Command3\_Click()は「次へ」のコマンド・ボタンのクリックに対応するプロシージャである。このプロシージャにより、データがあるかぎり、次の一連のデータをテキスト・ボックスに表示することができる。⑥は右辺のiに0が代入される。これは、⑤で「i = 0」とされたためである。これにより、⑥の左辺のiは1となる。iがデータの数Lastを越えた場合の処理方法を定めたものが⑦である。この時、フォームに「No More Data」を表示し、「戻る」か「終了」をクリックしてください。」とメッセージ・ボックスが表示され、「vbOKOnly」オプションで「OK」のボタンが表示される。

iがデータ数を越えない場合の処理を定めているのが「Else」から「End If」までである。⑧で、dat1が代入されているdata(i, 1)をテキスト・ボックスのText1に表示する。Text1の枠の上には「名前」というキャプションが付いている。⑨では、dat2が代入されているdata(i, 2)をテキスト・ボックスのText2に表示する。Text2の枠の上には「年齢」というキャプションが付いている。⑩では、dat3が代入されているdata(i, 3)をテキスト・ボックスのText3に表示する。Text3の枠の上には「誕生日」というキャプションが付いている。

Command4\_Click()は「戻る」のコマンド・ボタンのクリックに対応するプロシージャである。これにより、以前に表示されたデータを順にさかのぼって表示することができる。⑪の右辺「i - 1」の

結果を左辺に代入することにより、左辺の  $i$  は1だけ若い数字となる。この操作を繰り返して、 $i$  が0以下となった場合の処理が⑫のIf文である。「Then」以下の処理で、フォームに「First of Data」と表示し、メッセージ・ボックスに「次へ」ボタンをクリックしてください。」と表示する。⑬で、 $i$  が1となるようにする。 $i$  が1以上になった場合、⑧、⑨、⑩と同様に、「Else」から「End If」までの⑭、⑮、⑯で、対応するテキスト・ボックスに配列data()の内容が表示される。

ここで、配列data()を制御する変数  $i$  に注目してみると、 $i$  は次のように変化していることがわかる。まず  $i$  は、複数のプロシージャで用いられるためパブリック変数とされる。この時、 $i$  は初期化されていて、0である。「開始」のコマンド・ボタンをクリックされるとCommand1\_Click()が実行される。ファイルから5人分のデータが取り込まれる。この時、 $i$  はFor...Next文の引数として用いられ、1から5まで変化する。ファイルが閉じられた時点では、 $i$  は5だが、⑤でふたたび0とされる。「次へ」のコマンド・ボタンをクリックされると、Command3\_Click()が実行される。⑥で、右辺の「 $i + 1$ 」の  $i$  に⑤で代入された0がはいる。このため、右辺は  $0 + 1$  となる。この結果、⑥の左辺の  $i$  は1となる。⑦で  $i$  の1とLastの5が比較され、「 $i > \text{Last}$ 」が「 $1 > 5$ 」となり、条件式は偽となる。制御はElse以下の⑧に移り、名前というキャプションが付いた左端のテキスト・ボックスにはdata( $i, 1$ ) i.e.「山田吾一」が表示され、「年齢」というキャプションが付いた中央のテキスト・ボックスにはdata( $i, 2$ ) i.e.「30」が表示され、「誕生日」というキャプションが付いた右端のテキスト・ボックスにはdata( $i, 3$ ) i.e.「12 09 65」が表示される。もう一度、「次へ」のコマンド・ボタンをクリックされると、⑥の右辺で  $i$  は1が加算されて2となり、data(2, 1), data(2, 2), data(2, 3)の内容が表示され、さらに「次へ」がクリックされると、 $i$  は3となり、data(3, 1), data(3, 2), data(3, 3)の内容が表示される。

仮に、もうあと2回、「次へ」がクリックされると、 $i$  は4、5と変化する。さらに「次へ」がクリックされたとすると、⑥で右辺は「 $5 + 1$ 」となり左辺の  $i$  には6が代入される。このため、⑦の条件式は「 $6 > 5$ 」となり、真となる。とすると、Then以下の3行が実行され、「No More Data」の表示とともに、「戻る」か「終了」をクリックしてください。」のメッセージ・ボックスが表示され、フォームはクリアされることになる。

ふたたび、 $i$  が3の時点に戻って  $i$  の変化をトレースしてみる。ここで、「戻る」のコマンド・ボタンをクリックされるとする。Command4\_Click()のプロシージャが実行される。⑪の右辺「 $i - 1$ 」の  $i$  に3が代入される。3 - 1の演算結果2が左辺の  $i$  に代入される。⑫の条件式は「 $2 < 1$ 」となり、成立しなくなる。そのため、制御は「Else」に移り、⑭、⑮、⑯でdata(2,1), data(2,2), data(2,3)のデータが表示される。さらに「戻る」ボタンをクリックすると、⑪で右辺は「 $2 - 1$ 」となり、左辺の  $i$  には1が代入される。⑫で「 $1 < 1$ 」が偽となり、ふたたび、「Else」へ飛び、data(1,1), data(1,2), data(1,3)が表示される。ここでもう一度「戻る」がクリックされると、⑪で「 $1 - 1$ 」となり左辺の  $i$  には0が代入される。⑫で条件式「 $0 < 1$ 」が真となり、制御はThen以下の3行を実行する。⑬で、右辺は「 $0 + 1$ 」となり、左辺の  $i$  には1が代入される。このため、さらにもう一度「戻る」がクリックされても、 $i$  は0より小さくなることはない。

各プロシージャを以上のように記述することにより、データ・ファイルを読み込んで、データをテキスト・ボックスに順に表示し、必要に応じてデータをさかのぼって表示することができるようになる。「開始」ボタンにより、データ・ファイルを開いて、配列data()にデータを代入する。「次へ」ボタンをクリックすると、データを順に表示することができ、「戻る」ボタンをクリックすると、データを元に戻って表示することができる。「終了」ボタンにより、いつでもプログラムを終了することができる。

実行例は「開始」により、データが配列に読み込まれ、「次へ」ボタンのクリックを4回繰り返して、テキスト・ボックスに3変数が表示された時の画面である。

## まとめ

VISUAL BASICによるデータベースの作成に向けて、その基礎となるファイルの操作法について述べた。「1 ファイル・データのプリント表示」では、あらかじめ作られたファイルを読み込み、その内容をフォーム上にプリント表示する方法について述べた。「2 ファイル・データのテキスト・ボックスへの表示」では、読み込んだデータを、それぞれ対応するキャプションがつけられたテキスト・ボックスに、指定したプロパティで表示する方法について述べた。また、日付(Date)型変数についても述べた。データを最後まで読み込んでしまった場合、プログラムを終了させるために、メッセージ・ボックスに「終了ボタンをクリックしてください。」と表示し、終了のコマンド・ボタンをクリックするように指示する方法についても述べた。「3 ファイル・データのテキスト・ボックスへの表示 逆方向検索を含む」では、対応するテキスト・ボックスにデータの内容を表示する場合に、データの最初から最後への順方向に表示させるだけでなく、以前に表示したデータにさかのぼって表示させる逆方向検索のプログラミング法についても解説した。

## 参考文献

- Microsoft Corporation 1997 「Microsoft Visual Basic 5.0 ランゲージ リファレンス Part 1」アスキー出版局  
Microsoft Corporation 1997 「Microsoft Visual Basic 5.0 ランゲージ リファレンス Part 2」アスキー出版局  
江藤潔 1997 「Visual Basicで始めるプログラミング」講談社  
川口輝久・河野勉1997 「Visual Basic 5.0 基礎編」技術評論社  
川口輝久・河野勉1997 「Visual Basic 5.0 コントロール編」技術評論社  
中島省吾1998 「Visual Basic 5.0 入門」スパイク  
門田幸太郎 1998 「VISUAL BASICのプログラミング法 1 その特徴とプログラミングの基礎」立命館産業社会論集 第34巻 第3号  
門田幸太郎 1999 「VISUAL BASICのプログラミング法 2 配列データの操作」立命館産業社会論集 第34巻 第4号