

## 〔情報処理研究〕

## VISUAL BASICのプログラミング法 4

## - ファイルの操作；読み込みと表示の応用 -

門田 幸太郎\*

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、検索などの方法の応用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、Form\_Load() プロシージャ<sup>2)</sup>やファイル・データとコードでの変数名との関連性について述べた。ファイルの表示方法では、画面を切り替えて表示する方法、一連のデータのある項目だけを表示しておいて、適宜他の項目も表示できるように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，Form\_Load，Select Case，Exit，Timer

## 目次

- |                          |                    |
|--------------------------|--------------------|
| 1 ファイル・データのテキスト・ボックスへの表示 | 分割表示               |
| 2 ファイル・データのテキスト・ボックスへの表示 | 2項目を順次表示           |
| 3 ファイル・データのテキスト・ボックスへの表示 | コマンド・ボタンによる「解説」の表示 |
| 4 ファイル・データの検索            |                    |

## 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、また、門田(1999a)では、配列データの操作について述べた。門田(1999b)では、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述べた。本稿では、その応用として、英単語とその意味の解説の2項目が対になったデータ・ファイルを一種のデータ・ベースとして、その表示の仕方について述べた。

## 1 ファイル・データのテキスト・ボックスへの表示 分割表示

【例題1】下の《参考》に示したようなテキスト・ファイルの英和辞書を使って、英語と解説を2つ

---

\* 立命館大学産業社会学部教授

1) ソフトはMicrosoft Visual Basic 5.0 Professional 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, 1999a, 1999b)を参照。

のテキスト・ボックスにそれぞれ同時表示せよ。

《参考》d:\¥mondenEJ.txt

a p a i n i n t h e n e c k

i d . 《米略式》いやな人[物]

a b d u c t i o n

n . 誘拐

a b i d e

"v i . 1 とどまる(stay), 住む(live)[at, in]; 居残る(remain) 2 (規則・約束などを)

忠実に守る 遵守[固守]する(stick[keep] to)[by]; ~を甘受する[by]:abide by one's

promise[decision]約束(決心)を守る"

a b o v e b o a r d

a d . a . <事が>公明正大に[な], りっぱに[な], 正直に[な], ガラス張りで[の], A N T .

u n d e r h a n d

a b s c o n d

v i . 逃亡する

a b s o l u t i o n

n . [主にキリスト教](苦行・宗教儀式による)(罪の)許し; 赦免; (義務・約束の)免除; (罪の)許しの言葉, 悪行消滅宣言

《以下省略》



図1 例題1の実行例

ツール・ボックスからフォーム上に、3つのコマンド・ボタンと2つのテキスト・ボックス、2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「次へ」とし、Command2を「戻る」、Command3を「終了」とする。テキスト・ボックスのTextを両方とも取り除く。ラベルのキャプションについては、Label1を「英語」、Label2を「解説」とする。プロパティ・ウィンドウを利用して、テキスト・ボックスのプロパティを設定する。Text1については、フォント名をMS Pゴシック、スタイルをRegular、サイズを18とする。Text2については、フォント名をMS Pゴシック、スタイルをRegular、サイズを12とする。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 ]

```
Option Base 1
```

```
Dim data(3000, 2) As String
```

```
Public i, Last
```

```
Private Sub Form_Load()
```

```
    Dim dat1 As String
```

```
    Dim dat2 As String
```

```
    Open "d:\%mondensEJ.txt" For Input As #1
```

```
    Last = 100
```

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

```
        Input #1, dat1 ①
```

```
        Input #1, dat2 ②
```

```
        data(i, 1) = dat1
```

```
        data(i, 2) = dat2
```

```
    Next
```

```
    Close #1
```

```
    i = 0
```

```
    MsgBox "読み込み完了！！「次へ」をクリックしてください", vbOKOnly
```

```
    Cls
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
    i = i + 1
```

```
    If i > Last Then Print "No More Data" Else
```

```
        Text1.Text = data(i, 1)
```

```
Text2.Text = data(i, 2)
End Sub

Private Sub Command2_Click()
    i = i - 1
    If i < 1 Then
        Print "First of Data"
        i = i + 1
    Else
        Text1.Text = data(i, 1)
        Text2.Text = data(i, 2)
    End If
End Sub

Private Sub Command3_Click()
    End
End Sub
```

ファイル操作の応用として、英和辞書を利用するためのプログラムを作成する。まず《参考》に示したように、英語とその解説を交互に入力したテキスト・ファイルを作っておく。この時、データの区切りはリターンによって行う。ただし、「解説」のテキスト・ボックスでの表示において、任意の位置で改行したい場合はダブル・クォーテーションを利用する。リターンを含めたデータ文全体を「"」で挟み、さらにリターンを付ける。これによって、テキスト・ボックス内に表示される文の所定の位置で改行ができる。データの区切りには、半角のコンマ「,」も使われる。表示される文中で半角のコンマを利用する場合は、上述のように、半角のコンマを含む文をダブル・クォーテーションで挟んだ後にリターンを入れるという方法を使わなければならない。

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。例題2で触れたように、一般にVISUAL BASICではユーザーの反応がイベントとなり、プロシージャが実行される。例題3でも、「開始」ボタンをクリックすることによってCommand1\_Click()のプロシージャが稼働された。しかし、フォームのロード自体をひとつのイベントとしてプロシージャを記述することができるので、ユーザーの反応を待たずして、プロシージャを稼働させることができる。これによって、ユーザー側からいえば、「開始」ボタンをクリックするというステップを省くことができる。また、プログラミングする側からいえば、コマンド・ボタンを取り込んで、プロパティを設定するという手順を省き、フォーム上のスペースを節約することができる。変数宣言の後、dドライブのルート・ディレクトリにある¥ mondenEJ.txtからデータが取り込まれる。①で、dat1に英語、②でdat2にその解説が読み込ま

れる。それらを配列data()に代入する。ファイルが閉じられ、メッセージ・ボックスに「読み込み完了!!」「次へ」をクリックしてください」と表示される。

Command1\_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。これにより、データが最後でないかぎり、テキスト・ボックスのText1に英語、Text2に解説が表示される。

Command2\_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、データが最初の方に戻ってテキスト・ボックスに表示される。

Command3\_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。

以上のようなプロシージャの記述により、テキスト・ファイルの英和辞書を使って、英語と解説を同時にテキスト・ボックスに表示し、必要に応じて戻ることもできるようになる。プログラムの実行に伴うフォームのロードにより、ファイルからデータが読み込まれ、配列に代入される。「次へ」ボタンをクリックすると、英語とその解説が同時にテキスト・ボックスに表示される。さらに、「次へ」ボタンをクリックすると、データ・ファイルにおける次の英語とその解説が同時にテキスト・ボックスに表示される。「戻る」ボタンをクリックすると、データ・ファイルにおけるひとつ前の英語とその解説が同時にテキスト・ボックスに表示される。さらに、「戻る」ボタンをクリックし続けることにより、順次、データ・ファイルをさかのぼって、順番の若い英語とその解説を同時にテキスト・ボックスに表示することができる。

実行例は、次のようにして得られたものである。プログラムを実行した後に、「読み込み完了!!」「次へ」をクリックしてください」がメッセージ・ボックスに表示される。「OK」をクリックして、「次へ」ボタンを4回クリックすると、「aboveboard」が表示される。図1は、この時の実行例である。さらに、「次へ」ボタンをクリックすると、辞書ファイルの次の英語である「abscond」とその解説が、「戻る」ボタンをクリックすると、ひとつ前の英語である「abide」とその解説が表示されることになる。

## 2 ファイル・データのテキスト・ボックスへの表示 2項目を順次表示

【例題2】例題1の英和辞書を使って、データをテキスト・ボックスに英語と解説とを順次、1項目ずつ交互に表示せよ。

ツール・ボックスからフォーム上にコマンド・ボタンを3つとテキスト・ボックス1つ、ラベル1つを取り込み、実行例のように配置する。Command1のキャプションを「次へ」とし、Command2のキャプションを「戻る」とし、Command3のキャプションを「終了」とする。プロパティ・ウィンドウにおいてテキスト・ボックスのTextを取り除く。Label1は「英語・解説」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。

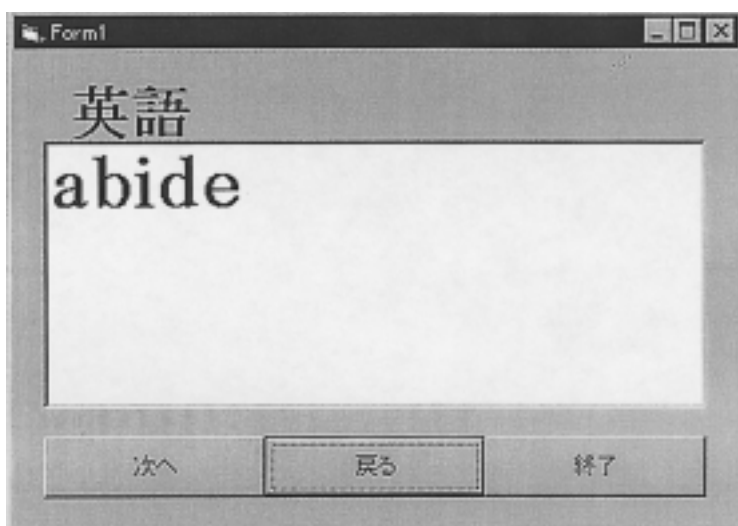


図2 例題2の実行例1 英語項目の表示

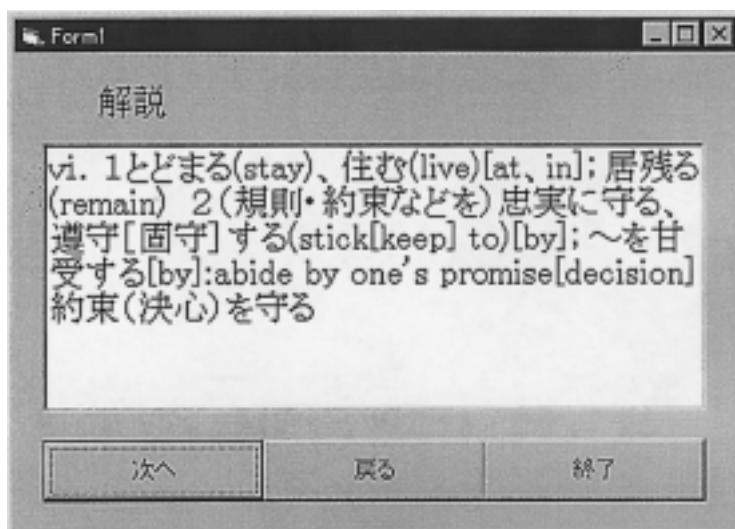


図3 例題2の実行例2 解説項目の表示

[ プログラム例 ]

Option Base 1

Dim data(3000)

Public i, Last

Private Sub Form\_Load()

    Open "d:\ymondenEJ.txt" For Input As #1

```
Last = 2170
For i = 1 To Last
    Input #1, data(i)           ①
Next
Close #1
i = 0
MsgBox "準備完了！！「次へ」をクリックしてください", vbOKOnly
End Sub
```

```
Private Sub Command1_Click()
    i = i + 1
    If i > Last Then Print "No More Data" Else
        Call Selection(i)       ②
    End Sub
```

```
Private Sub Command2_Click()
    i = i - 1
    If i < 1 Then
        Print "First of Data"
        Cls
        i = i + 1
    Else
        Call Selection(i)       ③
    End If
End Sub
```

```
Private Sub Command3_Click()
    End
End Sub
```

```
Private Sub Selection(i)       ④
    Select Case i Mod 2        ⑤
        Case 1                ⑥
            Label1.Caption = "英語" ⑦
            Label1.FontSize = 24     ⑧
```

```

Label1.FontBold = True           ⑨
Text1.FontSize = 30             ⑩
Text1.FontBold = True          ⑪
Case 0
Label1.Caption = "解説"        ⑫
Label1.FontSize = 14           ⑬
Label1.FontBold = False        ⑭
Text1.FontSize = 14            ⑮
Text1.FontBold = False        ⑯
End Select                       ⑰
Text1.Text = data(i, 1)
End Sub                           ⑱

```

プログラムの実行に伴いForm\_Load()のプロシージャが稼働する。これにより、¥mondenEJ.txtから英語とその解説の合計2170個のデータが読み込まれる。そのデータが①で配列data(i)に代入される。これによって、iが奇数の場合には英語が、偶数の場合には解説が代入されることになる。

Command1\_Click()のプロシージャでは「次へ」のコマンド・ボタンがクリックされるたびに、テキスト・ボックスに英語と解説が交互に表示される。その表示のためのサブルーチンSelection(i)が③のCall文で呼び出される。この引数iは表示する配列data(i)の引数となっている。

Command2\_Click()のプロシージャでは「戻る」のコマンド・ボタンがクリックされると、引数が1減らされて、サブルーチンSelection(i)がCallされる。

Command3\_Click()は終了のためのプロシージャである。

サブルーチンSelection(i)では、Call文によって引き継がれた引数iが、奇数の場合には英語の画面が、iが偶数の場合には解説の画面がそれぞれ表示される。⑤はSelect Case文である。If文は、条件式の真か偽かに対応した処理方法しか設定することができない。しかし、Select Case文は条件式の値に応じて、複数の処理を対応させることができる。Select Case文の構文の形式は

```

Select Case 条件式
    Case 値 ( to 値 )
        処理
    Case 値 ( to 値 )
        処理
    . . .
    Case Else
        処理
End Select

```



という構文となる。条件式の値がCaseの右側の値と一致した場合、または、「値 to 値」という一定の範囲内の値をとる場合に、その処理を行うことになる。ここでは、「 $i \text{ Mod } 2$ 」が条件式となる。「 $x \text{ Mod } y$ 」は、 $x$ を $y$ で割った場合の余りを求める演算である。「 $i \text{ Mod } 2$ 」の場合は、 $i$ が奇数であるか、偶数であるかを判断している。「Case 1」は $i$ を2で割った場合の余りが1の場合となる。つまり、 $i$ が奇数の時はラベルのキャプションを「英語」とし、テキスト・ボックスのフォントの大きさを30とする。「Case 0」は $i$ を2で割った場合の余りが0の場合となる。つまり、 $i$ が偶数の時はラベルのキャプションを「解説」とし、テキスト・ボックスのフォントの大きさを14とする。これによって、配列data( $i$ )の内容、すなわち $i$ が奇数の時は英語が、偶数の時は解説が、テキスト・ボックスにそれぞれのプロパティで表示される。

以上のプロシージャにより、プログラムの実行とともにファイルが読み込まれる。「次へ」ボタンにより、英語と解説が順次表示される。「戻る」ボタンにより、配列順番の若いデータにさかのぼって、英語と解説が表示される。

実行例について見てみる。プログラムを実行した後に、「読み込み完了!! 「次へ」をクリックしてください」がメッセージ・ボックスに表示される。「OK」をクリックして、「次へ」ボタンを5回クリックしたのが実行例1(図2)で、さらにもう1回クリックしたのが実行例2(図3)である。このように、英語の画面(実行例1)と解説の画面(実行例2)とを、「次へ」か「戻る」のボタンをクリックするたびに、交互に表示させることができる。

### 3 ファイル・データのテキスト・ボックスへの表示 コマンド・ボタンによる「解説」の表示

【例題3】例題1の英和辞書を使って、まず英語だけを順次表示せよ。次に「解説」ボタンをクリックすることにより表示された英語に対応する解説も表示できるようにせよ。

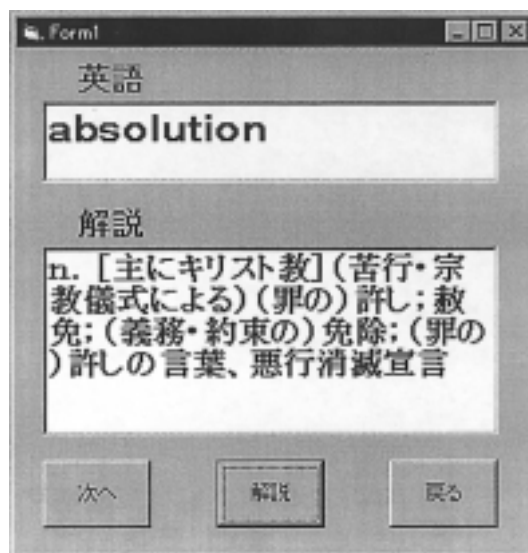


図4 例題3の実行例

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2つのテキスト・ボックス, 2つのラベル, 1つのタイマーを取り込み, 実行例のように配置する。ただし, タイマーは「オブジェクトの表示」画面には表れるが, 実行画面には表示されない。コマンド・ボタンのキャプションは, Command1を「次へ」とし, Command2を「終了」, Command3を「解説」, Command4を「戻る」とする。ラベルのキャプションについては, Label1を「英語」とし, Label2を「解説」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 ]

Option Base 1

Dim data(3000, 2)

Public i, Last, t ①

Private Sub Form\_Load() 'ロード

Dim dat1 As String

Dim dat2 As String

Open "d:\ymondenEJ.txt" For Input As #1

Last = 1085

For i = 1 To Last

Input #1, dat1

Input #1, dat2

data(i, 1) = dat1

data(i, 2) = dat2

Next

Close #1

i = 0 ②

Text1.Text = "All set!!!"

Text2.Text = "さあ、はじめましょう。"

End Sub

Private Sub Command1\_Click() '次へ ③

i = i + 1 ④

If i > Last Then ⑤

Print "No More Data"

t = 1 ⑥

Else

```
Text1.Text = data(i, 1)           ⑦
Text2.Text = ""                   ⑧
End If
End Sub

Private Sub Command2_Click() '終了
Text1.Text = "バイバイ！！"
End
End Sub

Private Sub Command3_Click() '解説
If i > Last Then
Text1.Text = "No More Data"
Else
Text2.Text = data(i, 2)           ⑨
End If
End Sub

Private Sub Command4_Click() '戻る
i = i - 1
If i < 1 Then
Print "First of Data"
i = i + 1
Else
Text1.Text = data(i, 1)
Text2.Text = ""
End If
End Sub

Private Sub Timer1_Timer() 'タイマー ⑩
If t = 0 Then Exit Sub           ⑪
Timer1.Interval = 1500          ⑫
Select Case t                     ⑬
Case 1
Label1.Caption = "E N D"
```

```
Label2.Caption = "終了"
Text1.Text = "「終了」ボタンをクリックしてください。"
Text2.Text = "またがんばってください。"
```

Case 2

```
Label1.Caption = "終了"
Label2.Caption = "E N D"
Text1.Text = "「終了」ボタンをクリックしてください。"
Text2.Text = "No More Data"
```

Case 3

```
Label1.Caption = "E N D"
Label2.Caption = "終了"
Text1.Text = "「終了」ボタンをクリックしてください。"
Text2.Text = "もうこれ以上データはありません。"
```

End Select

```
t = t + 1 ⑭
```

```
If t > 3 Then t = 1 ⑮
```

End Sub

①のパブリック変数の宣言で、*i* は配列data()の引数として、Lastはデータ数を示すために、*t* はタイマーを制御するために用いられる。

プログラムの実行とともにForm\_Load()のプロシージャが稼働する。Open文で指定されたファイルからデータが入力される。ここではデータの数が1085組となっている。dat1に英語が読み込まれ、配列data(*i*, 1)に順に代入される。dat2に解説が読み込まれ、配列data(*i*, 2)に順に代入される。たとえば、第*n*番目のデータは英語が配列data(*n*, 1)に、解説が配列data(*n*, 2)に代入されることになる。ファイルが閉じられて、②で配列の引数*i* が0に戻される。テキスト・ボックスの上に「All set!!!」、下に「さあ、はじめましょう。」と表示される。

「次へ」のコマンド・ボタンがクリックされるとCommand1\_Click()のプロシージャが実行される。このプロシージャにより、データがあるかぎり、テキスト・ボックスに英語のみを表示し、データがなくなれば、「No More Data」と表示することができる。③の文末にある「次へ」は注釈である。「*i*」は、それ以下が注釈であることを示す。注釈は文末なら、プログラムのどの行にでも付けることができる。まず、④で「*i* = *i* + 1」の右辺の*i* に0が代入される。これは上述のForm\_Load()プロシージャの②における「*i* = 0」を受けている。その結果、④の左辺の*i* には1が入ることになる。以後、このプロシージャが実行されるたびに、④で*i* は1ずつ増えることになる。次の⑤のIf文で、引数*i* がデータ数を越えているかどうかを判断する。データ数を越えている場合は、フォーム上に「No More Data」と表示される。データ数を越えていない場合は、⑦でテキスト・ボックスの

Text1にdata(i, 1)を表示し、⑧でText2には何も表示しない。

「終了」のコマンド・ボタンがクリックされるとCommand2\_Click()が実行され、プログラムは終了する。

「解説」のコマンド・ボタンがクリックされるとCommand3\_Click()が実行される。引数 i がデータ数を越えていると、テキスト・ボックスのText1に「No More Data」と表示される。引数 i がデータ数を越えていない場合は、⑨で下のテキスト・ボックスのText2に解説のデータdata(i, 2)が表示される。

Command4\_Click()は「戻る」のコマンド・ボタンがクリックされた場合に実行されるプロシージャである。このプロシージャが実行されるたびに、引数の i は1ずつ減らされる。i が1より小さくならないかぎり、つまり0とならないかぎり、Command1\_Click()と同様、テキスト・ボックスのText1にdata(i, 1)を表示し、Text2には何も表示しない。i が1より小さくなった場合には、「i = i + 1」により、左辺の i を1とする。

⑩はタイマーに関するプロシージャである。VISUAL BASICは一般にユーザーが起こすイベントによってプログラムが制御される。これに対して、タイマーは、ユーザーが起こすイベントとは無関係にプロシージャが稼働する。⑪の「Exit Sub」はプロシージャを強制的に抜け出すためのフロー制御文である。t が0であるかぎり、タイマーは起動しないようになっている。⑫はタイマーが稼働するインターバルを設定している。単位はミリ秒である。ここでは1.5秒ごとにプロシージャが起動する。⑬は t の値に応じて、3種類の処理が行われ、ラベルには「終了」と「END」が、テキスト・ボックスには「No More Data」と「もうこれ以上データはありません。」「またがんばってください。」がそれぞれ設定されている。 t が1ずつ増加する。⑭で増加した t が3を越える場合、⑮で t の値を1に戻す。これは、⑬のSelect Case文で t の値が1, 2, 3を繰り返すための設定である。

このようにコード記述されたプロシージャにより、まず英語だけを順次示し、必要に応じて、解説も表示させることができるようになる。プログラムの実行に伴い、ファイルからデータが読み込まれる。「次へ」ボタンにより、上のテキスト・ボックスに英語データが順次、表示されていく。表示された英語の解説を見たい場合、「解説」ボタンをクリックすると、下のテキスト・ボックスに解説が表示される。

実行例(図4)は、英語データを順に表示し、「absolution」が表示された時に、「解説」ボタンをクリックして解説が表示されたところである。

#### 4 ファイル・データの検索

【例題4】例題3において検索の機能を追加せよ。英語をテキスト・ボックスに入力して、「検索」ボタンをクリックすることにより、その英語がデータ・ファイルに含まれている場合は、解説を表示させよ。

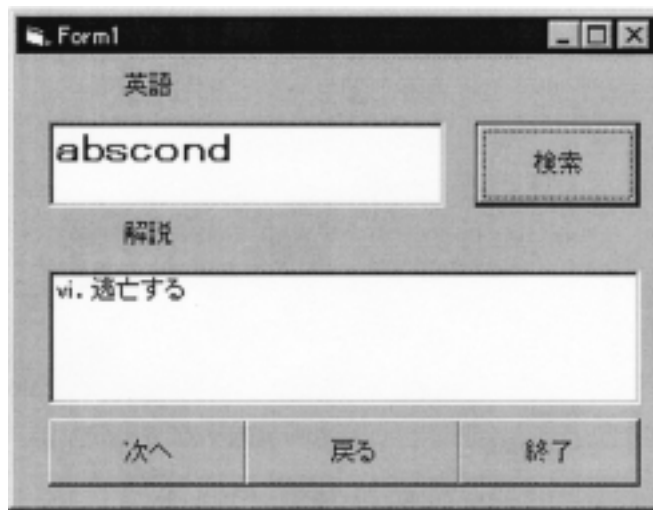


図5 例題4の実行例

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2つのテキスト・ボックス、2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「次へ」とし、Command2を「戻る」、Command3を「終了」、Command4を「検索」とする。ラベルのキャプションについては、Label1を「英語」とし、Label2を「解説」とする。各オブジェクトのプロシージャを次のようにコード記述する。

[ プログラム例 ]

```
Option Base 1
```

```
Dim data(3000, 2)
```

```
Public i, Last
```

```
Private Sub Form_Load()
```

```
    Dim dat1 As String
```

```
    Dim dat2 As String
```

```
    Open "k:\ymondenEJ.txt" For Input As #1
```

```
    Last = 1085
```

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

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

```
        data(i, 1) = dat1
```

```
        data(i, 2) = dat2
```

```
    Next
```

```
Close #1
i = 0
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 Command1_Click()
    i = i + 1
    If i > Last Then
        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)
        Text2.Text = data(i, 2)
    End If
End Sub

Private Sub Command2_Click()
    i = i - 1
    If i < 1 Then
        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
    Else
        Text1.Text = data(i, 1)
        Text2.Text = data(i, 2)
    End If
End Sub

Private Sub Command3_Click()
    End
End Sub
```

```
Private Sub Command4_Click()
```

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

```
        If data(i, 1) = Text1.Text Then ①
```

```
            Text2.Text = data(i, 2) ②
```

```
        Exit For ③
```

```
    End If
```

```
Next
```

```
End Sub
```

プログラムが実行されると、まず、Form\_Load()のプロシージャが働く。ファイルからデータが読み込まれ、英語が配列data(i, 1)に、その解説がdata(i, 2)に代入される。

Command1\_Click()のプロシージャは、「次へ」のコマンド・ボタンのクリックというイベントに対応している。これにより、読み込まれたデータがテキスト・ボックスに表示される。データが最後まで行った時は、「No More Data」とフォームに表示し、メッセージ・ボックスに「戻る」か「終了」をクリックしてください。」と表示する。

Command2\_Click()のプロシージャは、「戻る」のコマンド・ボタンのクリックというイベントに対応している。これにより、その時点でテキスト・ボックスに表示されているデータのひとつ前のデータが表示されることになる。データの最初まで戻った場合は、「First of Data」とフォームに表示し、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。

Command3\_Click()のプロシージャは、「終了」のコマンド・ボタンのクリックというイベントに対応している。これにより、プログラムを終了する。

Command4\_Click()のプロシージャは、「検索」のコマンド・ボタンのクリックというイベントに対応している。Form\_Load()で、ファイル・データがすべて配列data(i, 1)とdata(i, 2)に読み込まれている。上のテキスト・ボックスのText1に、検索したい英語を入力して「検索」ボタンをクリックする。この時の入力、ファイル作成したときと同じ入力の形式である全角で入力しなければならない。①のIf文で、そのテキスト・ボックスの内容、つまり、Text1.Textと同じものがあるかどうか、英語の配列data(i, 1)の中から検索される。もし同じものがあれば②で、その英語の配列data(i, 1)の引数iに対応する配列data(i, 2)が、下のテキスト・ボックスText2に表示される。③の「Exit For」はここでフロー制御がFor...Next文のループから強制的に抜け出ることを意味する。これにより、引数iには、ループを飛び出した時のiの値が保存されていることになる。この後、「次へ」か「戻る」のコマンド・ボタンがクリックされると、保存されたiの値が受け継がれ、検索で、見出された英語の次、または前の英語とその解説を表示することができるようになる。

以上、5つのプロシージャにより、テキスト・ボックスに英語を入力して、「検索」ボタンにより、その英語がデータ・ファイルに含まれている場合は、解説を表示させる検索の機能を持たせることができるようになる。「次へ」ボタンか「戻る」ボタンにより、データを順に表示したり、初めの



方に戻ったりすることができる。さらに、「検索」ボタンにより、Text1のテキスト・ボックスに入力された英語が、データ・ファイルに含まれているかどうかを検索することができる。もし、含まれていたら、その解説をText2のテキスト・ボックスに表示する。

実行例は上のテキスト・ボックスに「abscond」と入力して「検索」ボタンをクリックした時の状態である。

## まとめ

VISUAL BASICによるデータベースの作成に向けて、ファイルの操作法、とくに表示方法のバリエーションについて述べた。「1 ファイル・データのテキスト・ボックスへの表示 分割表示」では、あらかじめ作られたファイルから英語と解説の2項目からなるデータを読み込んでおいて、対応するテキスト・ボックスに表示する。この時、「次へ」のコマンド・ボタンをクリックすると順方向に、「戻る」のコマンド・ボタンをクリックすると逆方向にデータが表示される。「2 ファイル・データのテキスト・ボックスへの表示 2項目を順次表示」では、「英語」のテキスト・ボックスの画面と「解説」のテキスト・ボックスの画面とが交互に表示される。「3 ファイル・データのテキスト・ボックスへの表示 コマンド・ボタンによる「解説」の表示」では、まず「英語」だけを表示しておいて、つぎに「解説」のコマンド・ボタンをクリックすることにより、「解説」のテキスト・ボックスに解説項目が表示される。「4 ファイル・データの検索」では、「英語」のテキスト・ボックスに英語を入力して、「検索」のコマンド・ボタンをクリックすることにより、「解説」のテキスト・ボックスに解説項目が表示される。これにより、英語のデータ・ファイルをデータベースとして、検索するプログラムの作成方法について述べた。

## 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」立命館大学産業社会論集 第34巻 第3号  
 門田幸太郎 1999a「VISUAL BASICのプログラミング法 2 配列データの操作」立命館大学産業社会論集 第34巻 第4号  
 門田幸太郎 1999b「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」立命館大学産業社会論集 第35巻 第1号