

## —— ピボットテーブルをキマクロで作成する ——

(株) 竹内ハガネ商行 竹内寿一郎

## 1. はじめに

前回エクセル 2007 でのピボットテーブルを作成するマクロ用ボタンの作り方を紹介し [1]、その後エクセル 2003 で同様にキマクロを作成し実行してみたところ、飛んでもないことが生じたので、ここで改めて紹介する次第である。キマクロを作成してボタンを作り実行させる手順は 2007 も 2003 も全く同じようにすれば良いが、たまたまここで例として取り上げたピボットテーブルの作成に際してはちょっとした注意が必要であった。現象としては記録したキの履歴から作成したマクロを実行してみると、多くの場合誤動作となり、私のちょっとした VB の知識ではその誤動作の理由が分からなかった。

```
ActiveWorkbook.PivotCaches.Add(SourceType:=xlDatabase, SourceData:= _
"受取手形管理!R1C1:R339C11").CreatePivotTable TableDestination:=Range("A5"), _
TableName:="ピボットテーブル 1"
```

```
ActiveSheet.PivotTables("ピボットテーブル 1").SmallGrid = False
```

の 2 行のどちらかが常にエラーとなったのである。上の 1 行は実際には 3 行で書かれているが、上 2 つの末尾に\_(Continue 続く)のマークがあるので実質は 1 行である。

後でわかったことではあるが、Range("A5")のところ、""だけであれば新しい Sheet に集計表が出来、それなりに誤動作とはならないが、このままでは集計された表がどのシート上に作成されるか決められないので、Sheets("集計表").Select を宣言しておく必要があった。またその際に TableDestination のところで Range("A5")の代わりに R5C1 としてもよく、Range("R1C1")では当然エラーとなる。ただし、Range("A5")の代わりに Sheets("集計表").Range("A5")としてもよいらしいが、今度は次の ActiveSheet.PivotTables がどこのシートから作成するかが分からないというエラーが生ずるので、Sheets("集計表").Select を追加しなければならなくなる。しかもこれらのマクロを動かすには必ずシート「集計表」はクリアされていなければならない、エラーの度に文法のエラーなのか、あるいは既に表が存在する為のエラーなのかも確かめなければならなかった、非常に苦労したのである。

そこでこの小冊子では単にエクセル 2003 でマクロボタンを作るということだけではなく、「ピボットテーブルを作成するキマクロ」を登録するためのボタンを「エクセル 2003 では如何に作成するか」という主旨の小冊子であるということをおきたい。

## 2. Visual Basic Editor(VB Editor)

2007 と同様まず、マクロを使うためには Visual Basic を使う為の準備が必要である。そこで、「ツール(T)」ボタンから「マクロ(M)」を選び、「Visual Basic Editor(V)」を選ぶと、Visual Basic エディタが立ち上がる (次ページ図 1、図 2 参照)。

このメニューで「マクロ(M)」→「マクロ(M)...」を選ぶとマクロプログラムの一覧表が現れ、ここからマクロプログラムの実行および編集をすることが出来る。また、「新しいマクロの記録(R) ...」を選ぶと、後述するがこれ以降キマクロを記録してゆく。ここでは「Visual Basic Editor(V)」を選ぶ。通常何ら

かのマクロプログラムが登録されていればそれが現れるが、何もないと空欄になってしまう。



図1. マクロのメニュー

BV Editor 画面で、「挿入 (I)」から「標準モジュール(M)」を選ぶと「Module1」というシートが表示され、そこに Visual Basic でプログラムを記録することになる。これらの Module シートは画面上から消しても無くなることはなく、Module をクリックすると現れ、エディタも何回消しても再び立ち上げることは可能で、「Module の解放」を行なったり、Module シート上で「プログラムを消去」しない限りマクロプログラムは保存され続ける。

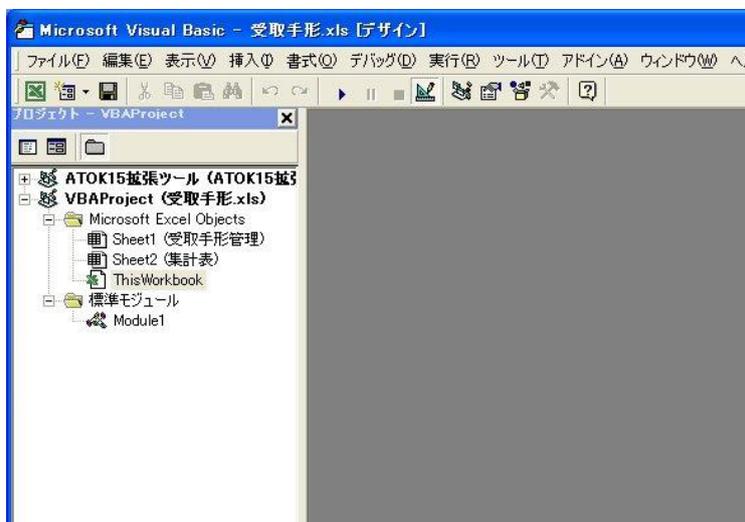


図2. Visual Basic Editor 画面

上の図では Module1 シートの内容が右側に表示されていないが、左側の Module1 をダブルクリックするとその内容が表示される。勿論この画面はいつでも消去可能で、邪魔になるようであれば消しても構わないし、「ツール」から「マクロ」、「Visual Basic Editor」を選べば再び表示するが出来る。

### 3. キーマクロの作り方(ピボットテーブルを例として)

マクロプログラムは本来 Visual Basic 言語を勉強しなければ書けない言語であるが、エクセルにはロータス 1-2-3 と同様にキーマクロ、すなわち押して行ったキの記録をとりそれを覚えさせ、後でそれを覚えさせた順に自動的に実行させるという機能を備えている。エクセルもロータス 1-2-3 も以前のバージョンでは押されたキの記録をそのまま記憶させたが、最近のエクセルではその記録が Visual Basic で記述されるようになってきているので、エクセルのユーザーはキを打ち込むだけで知らず知らずのうちに Visual Basic でマクロプログラムを書くことができるのである。

ここで例として「ピボットテーブルレポートを作成する」ことを取り上げるが、厄介なことにこのレポートはすでにその集計表が指定されたところに出来上がっていると、そこに上書きすることなくマクロエラーとなり、その先の作業を中止してしまうのである。さらに、そのエラーはある程度の Visual Basic の知識がないとすぐには解らないという性質のものである。それゆえこの場合、デバッグにあたっては常にこのことを年頭に入れておかなければならない。

さて集計される表はシート名[手形管理.xls]受取手形管理である(図3)。項目は受取手形の入金先、金額、銀行コード、手形期日、銀行名、年月日、名目年月日(1～5日、6～13日、14～25日、26～月末、に纏めたものにする名目の日付)で、年、月、日は手形期日から求め、銀行名および名目年月日は VLOOKUP 関数で右側に示した表から引いてくるようになっている。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	入金先	金額	銀行	手形期日	銀行名	年	月	日	名目年	名目月	名目日				
2	星田鉄鋼㈱	412,000	5	H10.03.05	5E信金	98	3	5	平成10年	3月	5日		0	5日	
3	星田鉄鋼㈱	412,000	3	H10.04.05	3C銀行	98	4	5	平成10年	4月	5日		6	13日	
4	星田鉄鋼㈱	412,000	6	H10.05.05	6F地銀	98	5	5	平成10年	5月	5日		14	25日	
5	星田鉄鋼㈱	412,000	1	H10.06.05	1A銀行	98	6	5	平成10年	6月	5日		26	末日	
6	星田鉄鋼㈱	412,000	3	H10.07.05	3C銀行	98	7	5	平成10年	7月	5日		31	末日	
7	星田鉄鋼㈱	412,000	6	H10.08.05	6F地銀	98	8	5	平成10年	8月	5日				
8	星田鉄鋼㈱	412,000	3	H10.09.05	3C銀行	98	9	5	平成10年	9月	5日				
9	星田鉄鋼㈱	412,000	3	H10.10.05	3C銀行	98	10	5	平成10年	10月	5日				
10	星田鉄鋼㈱	412,000	4	H10.11.05	4D信託	98	11	5	平成10年	11月	5日				
11	星田鉄鋼㈱	412,000	5	H10.12.05	5E信金	98	12	5	平成10年	12月	5日				
12	星田鉄鋼㈱	412,000	6	H11.01.05	6F地銀	99	1	5	平成11年	1月	5日				
13	星田鉄鋼㈱	412,000	6	H11.02.05	6F地銀	99	2	5	平成11年	2月	5日				
14	星田鉄鋼㈱	412,000	6	H11.03.05	6F地銀	99	3	5	平成11年	3月	5日				
15	星田鉄鋼㈱	412,000	4	H11.04.05	4D信託	99	4	5	平成11年	4月	5日				
16	星田鉄鋼㈱	412,000	3	H11.05.05	3C銀行	99	5	5	平成11年	5月	5日				
17	星田鉄鋼㈱	412,000	4	H11.06.05	4D信託	99	6	5	平成11年	6月	5日				
18	大下金型技研㈱	4,423,960	2	H10.04.30	2B銀行	98	4	30	平成10年	4月	末日		10	10月	
19	大下金型技研㈱	4,423,960	6	H10.05.31	6F地銀	98	5	31	平成10年	5月	末日		11	11月	
20	轉岩田鋼商店	3,000,000	3	H10.03.31	3C銀行	98	3	31	平成10年	3月	末日		12	12月	
21	協和鋼材(資)	775,437	1	H10.03.12	1A銀行	98	3	12	平成10年	3月	13日				
22	轉岩田鋼商店	8,149,849	1	H10.03.31	1A銀行	98	3	31	平成10年	3月	末日				
23	轉岩田鋼商店	3,000,000	3	H10.04.30	3C銀行	98	4	30	平成10年	4月	末日				
24	上野井鋼材㈱	757,523	6	H10.03.12	6F地銀	98	3	12	平成10年	3月	13日				
25	中上金属工業㈱	451,292	3	H10.01.31	3C銀行	98	1	31	平成10年	1月	末日				
26	轉岩田鋼商店	8,646,172	3	H10.04.30	3C銀行	98	4	30	平成10年	4月	末日				
27	轉岩田鋼商店	3,000,000	3	H10.05.31	3C銀行	98	5	31	平成10年	5月	末日				
28	轉大和機械製作	218,508	5	H10.02.20	5E信金	98	2	20	平成10年	2月	25日				
29	協和鋼材(資)	700,000	1	H10.03.13	1A銀行	98	3	13	平成10年	3月	13日				
30	協和鋼材(資)	490,841	4	H10.04.13	4D信託	98	4	13	平成10年	4月	13日				
31	城南機械製作㈱	518,082	5	H10.02.28	5E信金	98	2	28	平成10年	2月	末日				
	合計														

図3. 集計される元表

ピボットテーブルによる集計表作成の手順は次の通りである。

- (1) シート「集計表」に移動し、A5～J90をドラッグし、「編集(E)」→「クリア(A)」→「すべて(A)」を選択する
- (2) シート「受取手形管理」に移動し、表の枠内の任意のセルにポインタを置いて、「データ(D)」→「ピボットテーブルとピボットグラフ レポート(P)」を選択する

(3) ウィザード 1/3 が出てくるので、「エクセルのリスト/データベース」と「ピボットテーブル」を確認して、「次へ」

(4) ウィザード 2/3、使用するデータ範囲は\$A\$1～\$K\$399であることを確認する。「次へ」

(5) ウィザード 3/3、ここで新規のワークシートにすると毎回新しいシートが増えてゆくので、「既存のワークシート」に印を付けて、ワークシート「集計表」の A5 をクリックすると、「集計表!\$A\$5」となるので「OK」、その後「レイアウト」をクリック

(6) レイアウトの画面が現れる。この画面が出ないときは、作業を完全に中止し、表の四方が空いていること(左が A 列のときは上下と右の三方)、先頭行は必ず項目名であること、項目名は空欄がなく全て異なった名前であることを確かめること。その後キマクロをクリアーした後、はじめからやり直す

(7) 列に「銀行名」、行に上から順に「名目年」、「名目月」、「名目日」を重ねてドラッグする

(8) データ欄に「金額」をドラッグし、「合計：金額」と表示されれば「OK」、そうでなければ例えば「データの個数：金額」になっていれば、そこをダブルクリックして「合計」を選択して「OK」

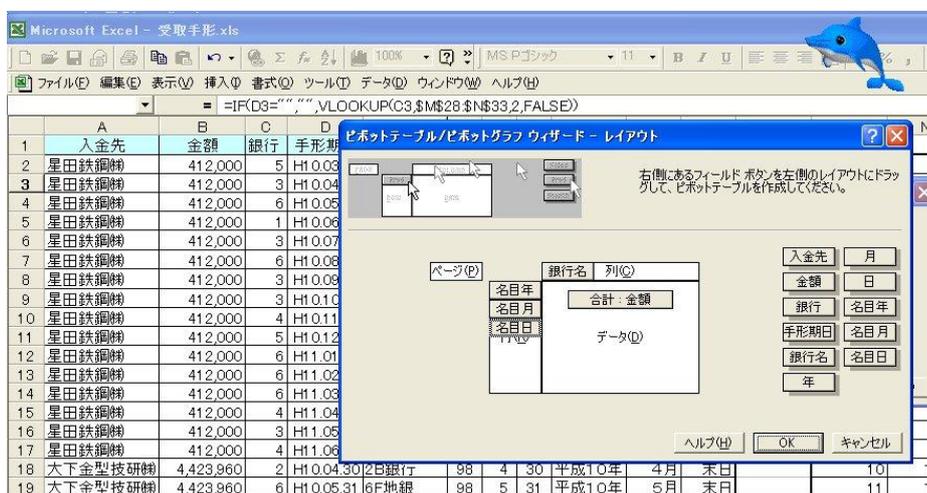


図 4. ピボットテーブル/ピボットグラフウィザード レイアウト

(9) 再びレイアウト画面に戻ったら「完了」をクリックで完成、D7～J90 をドラッグしてカンマのツールボタンをクリックして完成したのが下図である。

合計:金額			銀行名						
名目年	名目月	名目日	1A銀行	2B銀行	3C銀行	4D信託	5E信金	6F地銀	総計
平成10年	1月	13日					230,000	259,938	489,938
		25日					957,931		957,931
		末日				451,292	7,887,867	606,519	8,945,678
	1月計					451,292	8,117,867	1,824,388	10,393,547
	2月	5日				1,545,276		213,360	1,758,636
		13日				209,797	1,010,520	1,535,844	2,756,161
		25日	968,415	1,428,770	459,910	906,174	1,829,168	492,155	6,184,592
		末日	32,192,234	30,422,019	12,827,474	30,671,649	14,046,259	312,492	120,472,127
	2月計		33,160,649	31,850,789	15,042,457	32,588,343	17,724,631	804,647	131,171,516
	3月	5日	3,324,004	200,000	1,084,353	2,641,043	6,130,215	840,000	14,219,615
		13日	2,810,499	6,870,137	2,041,713	2,681,655	6,777,520	1,167,119	22,348,573

図 5. ピボットテーブルの完成

さて、ここで手順(1)～(9)で実際にキーマクロを作成してみよう。

「ツール(T)」ボタンから「マクロ(M)」、「新しいマクロの記録(R)...」を選択すると次のような図が現れる。



図6. 新しいマクロの記録

この図はこれから作成するキーマクロについて、(1)マクロ名を「集計表」にすること、(2)このマクロの実行にあたって、ショートカットキーを使う場合、CNTL+アルファベットの1文字を指定することが出来る、(3)出来上がったキーマクロの登録先を作業中のブックにする、(4)マクロの記録日とユーザー名を指定する事が出来る、ことを示している。

キーマクロを作成するにあたり、いくつか注意点を述べておく。マクロのスタートシートやセルを特に指定しない若しくは、セルの相対アドレス指定でプログラムを作っておくと、任意のシートやセルからマクロを始めることが出来る。これをマクロの相対アドレス指定といい、エクセル 2007 では相対アドレス指定のマクロを作成することが出来るが、エクセル 2003 ではその機能は無く、マクロの作成時にシートやセルポインタを Select や Active にしなければ現在使われているシート、またはセルポインタからマクロが実行される。シートの内容を消すなどの動作をするマクロの場合は、シート確認やセルを指定するコマンドを用意するか、シートやセル番地の確認後、マクロをスタートするようにプログラムしておく方が良い。またキーマクロは押したキーマクロがそのまま記録されるので、間違っただけの場合の履歴もそのまま記録されるので、2～3回練習をした後の、間違いの無いキーマクロをとるよう心がけよう。

「OK」ボタンを押すと次ページの図が現れ、「記録終了(R)」が押されるまで忠実にキーマクロがなされる。従って、記録は誤ってキーマクロしたものを修正して入力した場合でも、キーマクロしたそのまま記録されてしまう。記録されたキーマクロすなわち Visual Basic は基本的に、新しく挿入された Module に記録される。



図7. キーマクロの「記録終了」がある図

キーマクロは通常新しい Module に記録されることが多いので、新しいマクロや Module シートの数はどんどん増えてゆくので覚悟して欲しい。それがいやならば、(1)キーマクロを作り直すときは既に存在する同じ名前を付ける、(2)事後、コピー&ペーストでシートの整理を行う、とよい。このとき改めてマクロ名を変更することも出来る。

「Visual Basic Editor(V)」ボタンをクリックすると Visual Basic などの編集を行うウィンドウ (Visual Basic エディタとも言う)が開かれ、いつでも現在登録されているマクロプログラムを見る事が出来、またそれらを編集したり、修正することが出来る。このウィンドウは画面上から消しても、点けても良く、プログラム上から無くなることはないので、いつでも見たり、編集したりして再び消して、また点けることも出来る。

さて、記録されたマクロは以下のようなになるはずである。

Sub 集計表()

' 集計表 Macro

' マクロ記録日 : 2011/8/17 ユーザー名 : 竹内寿一郎

Sheets("集計表").Select

Range("A5").Select

ActiveWindow.LargeScroll Down:=2

Range("A5:J90").Select

Selection.Clear

Sheets("受取手形管理").Select

Range("A2").Select

ActiveWorkbook.PivotCaches.Add(SourceType:=xlDatabase, SourceData:= \_

"受取手形管理!R1C1:R339C11").CreatePivotTable TableDestination:=Range("A5"), \_

TableName:="ピボットテーブル 1"

ActiveSheet.PivotTables("ピボットテーブル 1").SmallGrid = False

ActiveSheet.PivotTables("ピボットテーブル 1").AddFields RowFields:=Array("名目年", \_

"名目月", "名目日"), ColumnFields:="銀行名"

ActiveSheet.PivotTables("ピボットテーブル 1").PivotFields("金額").Orientation = \_

```

xlDataField
Range("D7").Select
ActiveWindow.LargeScroll Down:=1
Range("D7:J61").Select
Selection.Style = "Comma [0]"
Sheets("受取手形管理").Select

End Sub

```

先頭行 `Sheets("集計表").Select` から5行は集計表をクリアしている。ピボットテーブルを作成するとき、既に指定した場所に集計表が存在しているとエラー扱いになるので、まず表を作る場所をクリアしておくことにする。『`ActiveWorkbook.PivotCaches.Add(SourceType:=xlDatabase, SourceData:= "受取手形管理!R1C1:R339C11").CreatePivotTable TableDestination:=Range("A5"), TableName:="ピボットテーブル1"`』がピボットテーブル作成コマンドの本体である。これ以降は行と列とデータフィールドの項目を指定していて、最後に D7~J62 までの範囲に対して3桁ずつカンマを付けるフォーマットを指定して、最後にシート「受取手形管理」へ移動して終了する。

実はこのマクロで実行すると、元のデータが書き換えられてしまうという誤動作を引き起こす。キーマクロで記録したにも拘わらず思わぬ結果をもたらすのである。エクセル 2003 でのピボットテーブル作成時の厭なところは、キー操作が間違っていないのにその通りに働かない点にある。ピボットテーブルを新しいシートに作成すればこの種のエラーは生じないことは確かである。その原因は「`Table Destination:=Range("A5")`」にある。`TableDestination` つまりピボットテーブルを作成する場所を A5 で指定すると、このままでは「受取手形管理」というシートの A5 であると解釈されてしまうからである。何故なら `SourceData` を指定するとき移動したシート「受取手形管理」がずっと活きているからである。そこで、`TableDestination:= ""`、これは新しいシートにピボットテーブルを作成するか、もしくは『`Sheets("受取手形管理").Select Range("A2").Select`』を削除して『`Sheets("集計表").Select`』とすると、この場合、シート「集計表」が選ばれ、正常に動作することが確かめられた。

```

Microsoft Visual Basic - 受取手形.xls - [Module1 (コード)]
ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) デバッグ(D) 実行(R) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
プロジェクト - VBAProject
  ATOK15拡張ツール (ATOK15拡張)
  VBAProject (受取手形.xls)
    Microsoft Excel Objects
      Sheet1 (受取手形管理)
      Sheet2 (集計表)
    ThisWorkbook
      標準モジュール
      Module1
  (General)
  集計表
  Sub 集計表()
  ? 集計表 Macro
  ? マクロ記録日 : 2011/8/17 ユーザー名 : 竹内寿一郎
  ?
  ?
  ? Sheets("集計表").Select
  ? Range("A5").Select
  ? ActiveWindow.LargeScroll Down:=2
  ? Range("A5:J90").Select
  ? Selection.Clear
  ? Sheets("受取手形管理").Select
  ? Range("A2").Select
  ? Sheets("集計表").Select
  ? ActiveWorkbook.PivotCaches.Add(SourceType:=xlDatabase, SourceData:=
  ? "受取手形管理!R1C1:R339C11").CreatePivotTable TableDestination:=Range("A5"), _
  ? TableName:="ピボットテーブル1"
  ? ActiveSheet.PivotTables("ピボットテーブル1").SmallGrid = False
  ? ActiveSheet.PivotTables("ピボットテーブル1").AddFields RowFields:=Array("名目年", _
  ? "名目年", "名目年"), ColumnFields:="銀行名"
  ? ActiveSheet.PivotTables("ピボットテーブル1").PivotFields("金額").Orientation = _
  ? xlDataField
  ? Range("D7").Select
  ? ActiveWindow.LargeScroll Down:=1
  ? Range("D7:J61").Select
  ? Selection.Style = "Comma [0]"
  ? Sheets("受取手形管理").Select
  End Sub

```

図8. 集計表のマクロ

#### 4. マクロの確認と実行

キーマクロが思ったように出来ているかどうか、出来あがった Visual Basic を読んだり、試しに実行してみることが出来る。まず、マクロの内容を見るには「ツール(T)」→「Visual Basic Editor(V)」をクリックする。Module の内容が表示されていない場合は左のプロジェクトウィンドウにある Module のところをダブルクリックすれば、内容が表示される。なお、左にプロジェクトウィンドウが無い場合は上にあるツールボタンからプロジェクト表示ボタンをクリックするとプロジェクトが表示される。

図8の左側がプロジェクトウィンドウでありこれを見ると、受取手形.xls が Sheet1(受取手形管理)、Sheet2(集計表)および Module1(標準 Module)からできていることが分かる。図の右側は Module1 の内容を示している。右側に目的とするものが何も現れていないときは、Module1 をダブルクリックすると、このような図が得られる。この内容がキーマクロの記録である。その内容を読むと、まずシート「集計表」を選び、集計表の範囲 A5～J90 を全てクリアーしている。(実際にはクリアー作業は、スクロールなど無しにして、範囲を選んで消す作業だけで十分なのである。)

Module1 のシート上では勿論コピー&ペーストが可能で、無駄な命令を全て消したり、コメントに修正したりして実行することも出来る。マクロを実行するにはマクロプログラム中にカーソルを置いて、Microsoft Visual Basic ウィンドウのツールボタンの実行「」(右三角)をクリックすれば良い。ちなみに、エラーの修正後のリセットには「」、実行中の一時停止は「」のボタンをクリックする。

マクロを実行する方法は幾通りものやり方がある。

- (1) エクセルの「ツール(T)」ボタンから「マクロ(M)」のボタンをクリックしてマクロプログラムを選んで実行させる。
- (2) あらかじめ定義しておいた CNTL+アルファベットでマクロを起動する。
- (3) Visual Basic Editor ウィンドウでカーソルを実行すべき Module の中に置いて実行キーをクリックする。
- (4) エクセルシート上に新たにボタンを作成して、そのボタンをクリックすることによりマクロを起動させる。

等である。次節にエクセルのシート上にボタンを作成してマクロを起動する方法について述べる。

なお、標準 Module を挿入してそこに Visual Basic のプログラムを書けば、自動的にマクロプログラムとして登録され実行することが出来る。このことは、ここでマクロプログラムを修正すれば、マクロの内容やマクロ名を変更することが出来ることをも意味している。

マクロプログラムを記述する Module(標準 Module)を新たに追加するには、Visual Basic ウィンドウでのメニュー「挿入」から標準モジュールを選ぶか、新たにキーマクロを作成するとよい。そうすれば標準モジュールはどんどん増加する。見たい標準モジュールをダブルクリックすることによっていつでも好きなモジュールがウィンドウに表示できるし、それらをコピー&ペーストすることにより、適当に編集することが出来る。

そして標準モジュールを消すには標準 Module 上で右クリックをして、ショートカットメニューから「Module の解放」を選ぶ。消す前にエクスポートしておけば Module を保存することが出来る。Module をエクスポートしなければその Module は完全に消去されることになる。

### 5. マクロボタンの作り方

「表示(V)」→「ツール(T)」→「コントロール ツールボックス」を選択するとツールボックスが現れ、その中のコマンドボタンをクリックして、+マークをドラッグすることにより適当な大きさのボタンボックスを描く。

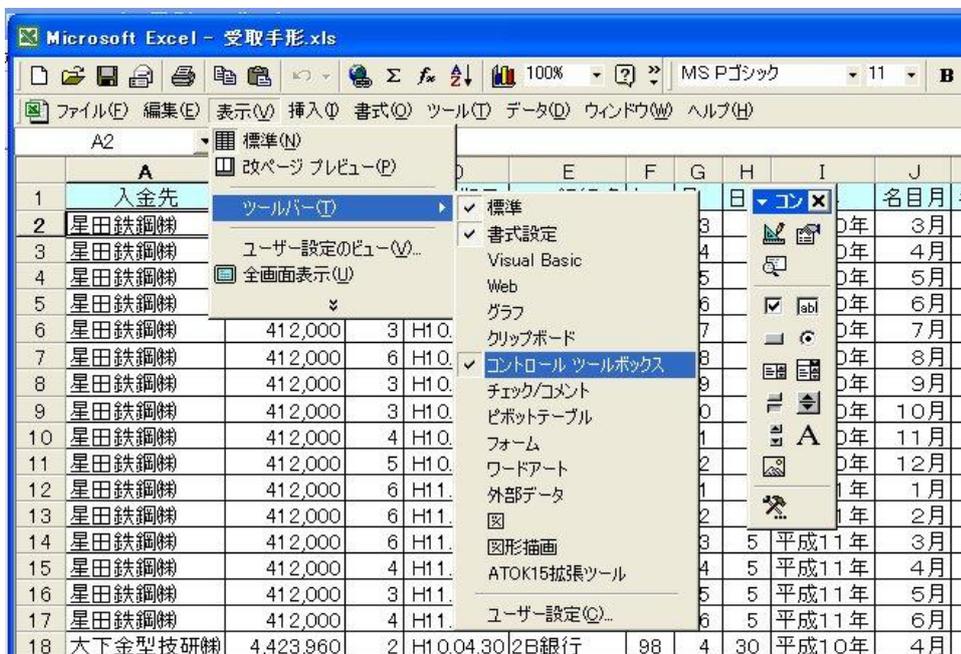


図9. コマンドボタンの作り方1

ボックスの大きさ、名前(CommandButton1 などの表記)、ボタンの色や文字の色と大きさ、フォントなどは後で自由に変更されるから、初めは本当に適当で良い。

出来上がったボックスの辺と四隅にある小さな8個の□印をドラッグすると、大きさが自由に変更出来る、+印が出ているときにドラッグするとボタンの位置が変更出来る、ボタンを移動させることが出来る。



図10. コマンドボタンの作り方2

## 6. ボタンのプロパティ(属性)

ボタンの上にポインタを置いて右クリック(ショートカットメニュー)もしくはツールボタンからプロパティを選んでボタンのいろいろな内容を指定することが出来る。この節ではここで設定できるボタンのいくつかの設定項目についてのべることにする。(この節ではマクロとコードは同じ意味で使用している。)



図 1 1. ショートカットメニュー

このようにプログラムリスト等を画面に出すようにするのが、「ツール(T)」→「Visual Basic Editor(V)」で、この操作は消したり、点けたり自由なので何回繰り返しても内容(コード)が消失されることはない。また作成したボタンに対するコードを見る為には、プロジェクト欄からボタンの存在するシートをダブルクリックするか、デザインモード  ボタンがへっこんでいる状態のときショートカットメニューで「コードの表示(V)」をクリックすると、コードが表示される。

左の図はボタンのプロパティを設定するために、ショートカットメニューを表示させたところである。このメニューで「プロパティ」を選択するとプロパティウィンドウが表示される。「コードの表示」を選択して「集計表」と書いてマクロを走らせるように設定する。この Editor 画面の左は、プロジェクトボタン  またはプロパティボタン  でそれぞれの画面を表示できるが、プロパティ画面の方が優先されるので注意しよう (図 1 2)。

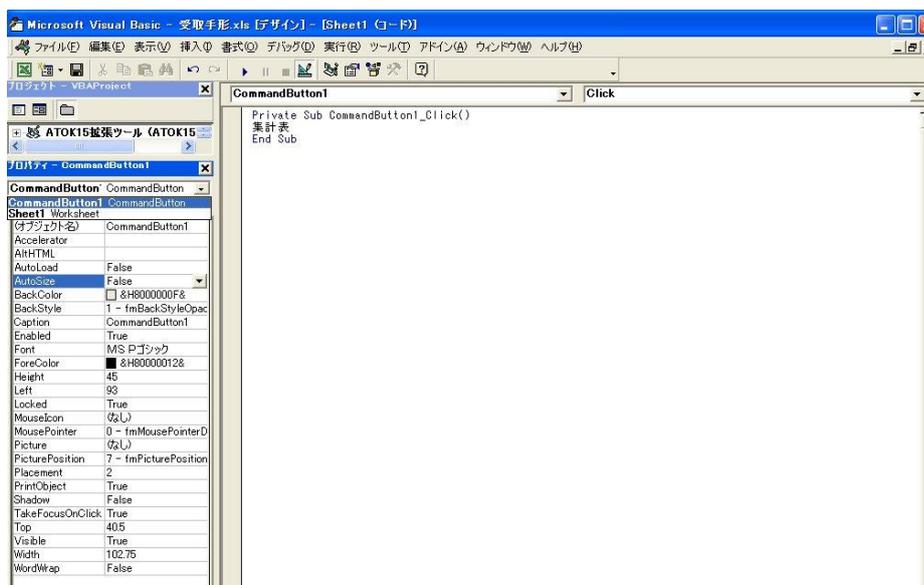


図 1 2. プロパティ画面

また先ほどのショートカットメニューのところ、**「コードの表示」**の代わりに**「プロパティ」**クリッ

クしてもプロパティ画面を表示することもできる。Visual Basic Editor のプロパティのツールボタンをクリックした後、CommonButton を選択することによりプロパティ画面を表示することもできる。

ここで、プロパティの全ては分からないが、少なくとも必要と思われる重要な項目だけでも分かる範囲で述べておきたい（アルファベット順に並べている）。

BackColor ... ボタンの背景の色、&H8000013&(いくつかの色が選べる)

Caption ... ボタンの名前、『集計』が良い(どんな名前をつけても良い)

Font ... ボタンのフォント名とスタイルと大きさ、MS P ゴシック・標準・11ポイント  
(いくつかの書体、大きさ等が選べる)

ForeColor ... ボタンの文字の色、&H80000012&(いくつかの色が選べる)

Height ... ボタンの縦の長さ、45

Left ... ボタンの左上のエクセルでの横座標の位置、93

Picture ... ボタンの中に絵を入れるか?、無し

PicturePosition ... 絵をボタン中のどの位置に入れるか、無し

Shadow ... ボタンに影を付けるか?、無し(False)

Top ... ボタンの左上のエクセルでの縦座標の位置、40.5

Width ... ボタンの幅の長さ、102.75

## 7. その他

マクロを起動するだけではなく、より一般的にはボタンを更に拡張した、いろいろな種類のボタンやリスト、テーブルを加えたりすることが出来る「フォーム機能」を使った方法もある。

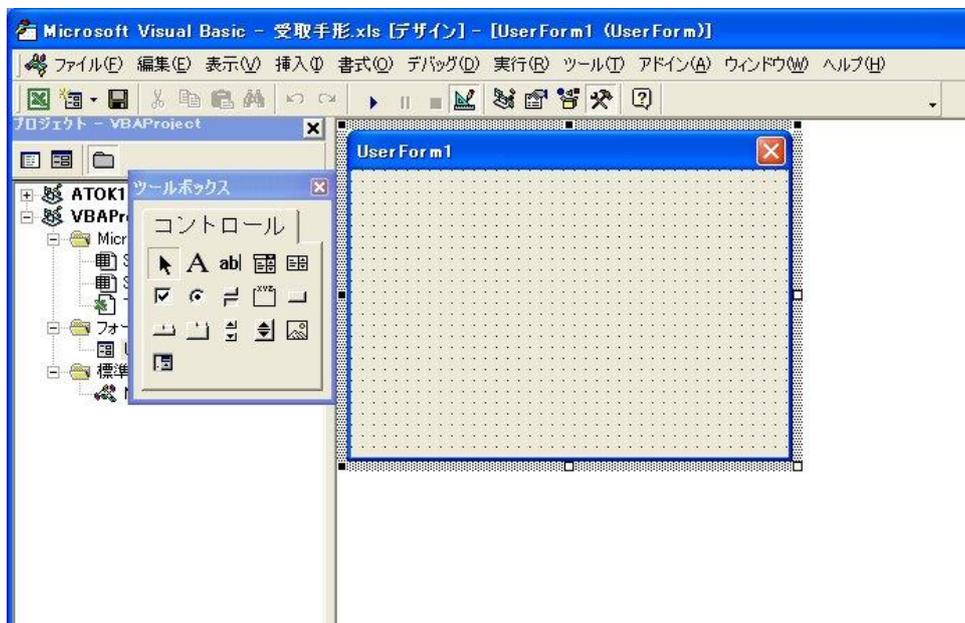


図13. ユーザーフォーム

Visual Basic Editor 画面でのメニューバーから「挿入(I)」→「ユーザーフォーム(U)」を選択すると、上のような図が得られ、左の「コントロール」から任意のボタンを選んだ後、右のフォームの任意の位置にドラッグすると、適当な大きさのコントロールボタンを作成することが出来る。このボタンに

プロパティを設定し、コード(マクロすなわち Visual Basic)を定義することにより、上述してきたようなマクロプログラムを走らせることも出来る。これは J 言語にも用意されているフォームと全く同じで、これについては西川、志村、竹内らによって既に J 研究会のチュートリアルセッション等で述べられている [3] [4]。

#### 【参考文献】

- 【1】 竹内寿一郎(2011)：エクセル 2007 でのマクロ用ボタンの作り方、JAPLA 研究会 6 月資料 2011.6.25 [http://www.japla.sakura.ne.jp/workshop/workshop/2011/Takeuchi\\_june2011.pdf](http://www.japla.sakura.ne.jp/workshop/workshop/2011/Takeuchi_june2011.pdf)
- 【2】 Excel Tips by pPoy(2006)：ピボットテーブル作成マクロ 3、《複数行設定・集計行非表示》 2006 年 3 月 11 日 <http://www.nurs.or.jp/~ppoy/access/excel/xlM048.html>
- 【3】 西川利男(2006)：J のウィンドウズ・プログラミングとそのグラフィックス入門、JAPLA シンポジウム 2006 資料 2006 年 12 月 9 日 [http://www.japla.sakura.ne.jp/workshop/symposium/2006/nishikawa\\_sympo2006.pdf](http://www.japla.sakura.ne.jp/workshop/symposium/2006/nishikawa_sympo2006.pdf)
- 【4】 竹内寿一郎(2009)：バリアオプションにフォームを付ける、JAPLA 研究会 6 月資料 2009 年 6 月 27 日 <http://www.japla.sakura.ne.jp/workshop/workshop/2009/Takeuchi200906.pdf>