

Excel マクロ – Visual Basic の基本 –

1. Excel ファイルの構成

Excel ファイルは、右図のように構成されている。

一般に Excel と言えば、右図で「Excel スプレッドシートの世界」と名付けた部分、すなわち、Excel を起動したときに表示されるスプレッドシート（1 ページの場合もあり、数ページの場合もある）のみであるように思われている。

Excel ファイルには、右図で「Visual Basic の世界」と名付けた部分も含まれる。すなわち、「Excel スプレッドシートの世界」と「Visual Basic の世界」とをあわせた全体が Excel ファイルとなる。

一般に、Excel ファイルの保存は、「Excel スプレッドシートの世界」の方で実施するが、自動的に「Visual Basic の世界」の部分も一緒に保存される。

Excel スプレッドシートの世界（図 2）と Visual Basic の世界（図 3）とは別のウィンドウで開くため、それぞれを別々に保存しなければならないように考えてしまうが、図 3 の「Visual Basic の世界」を表示するウィンドウ（以下「VBA ウィ

ンドウ」という）は、「編集」のために開かれるだけであり、保存せずに閉じたとしても「消えてしまう」わけではない。もちろん、図 2 の「Excel スプレッドシートの世界」のウィンドウ（以下「Excel ウィンドウ」という）を保存せずに閉じてしまえば、「Visual Basic の世界」の部分も一緒に「消えてしまう」ことになる。

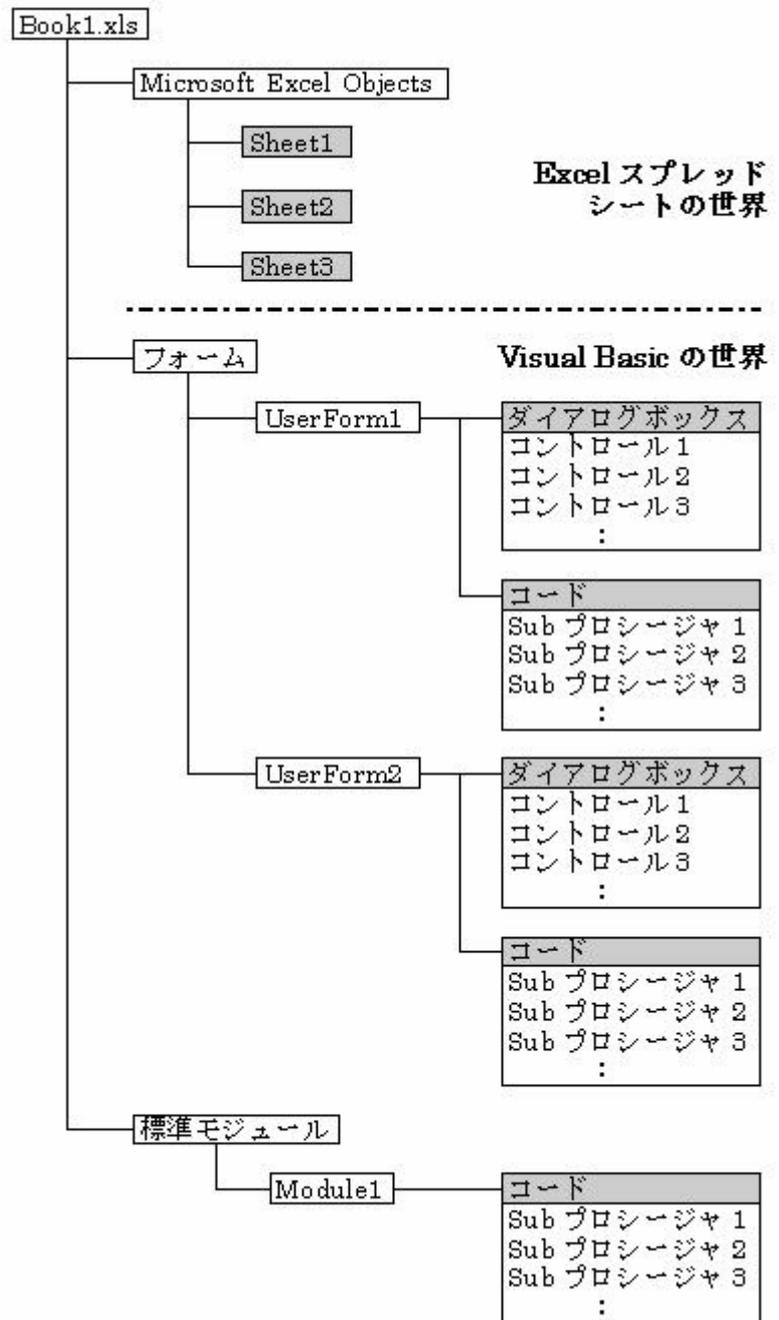


図 1 Excel ファイル (Book1.xls) の構成



図2 「Excel スプレッドシートの世界」のウィンドウ

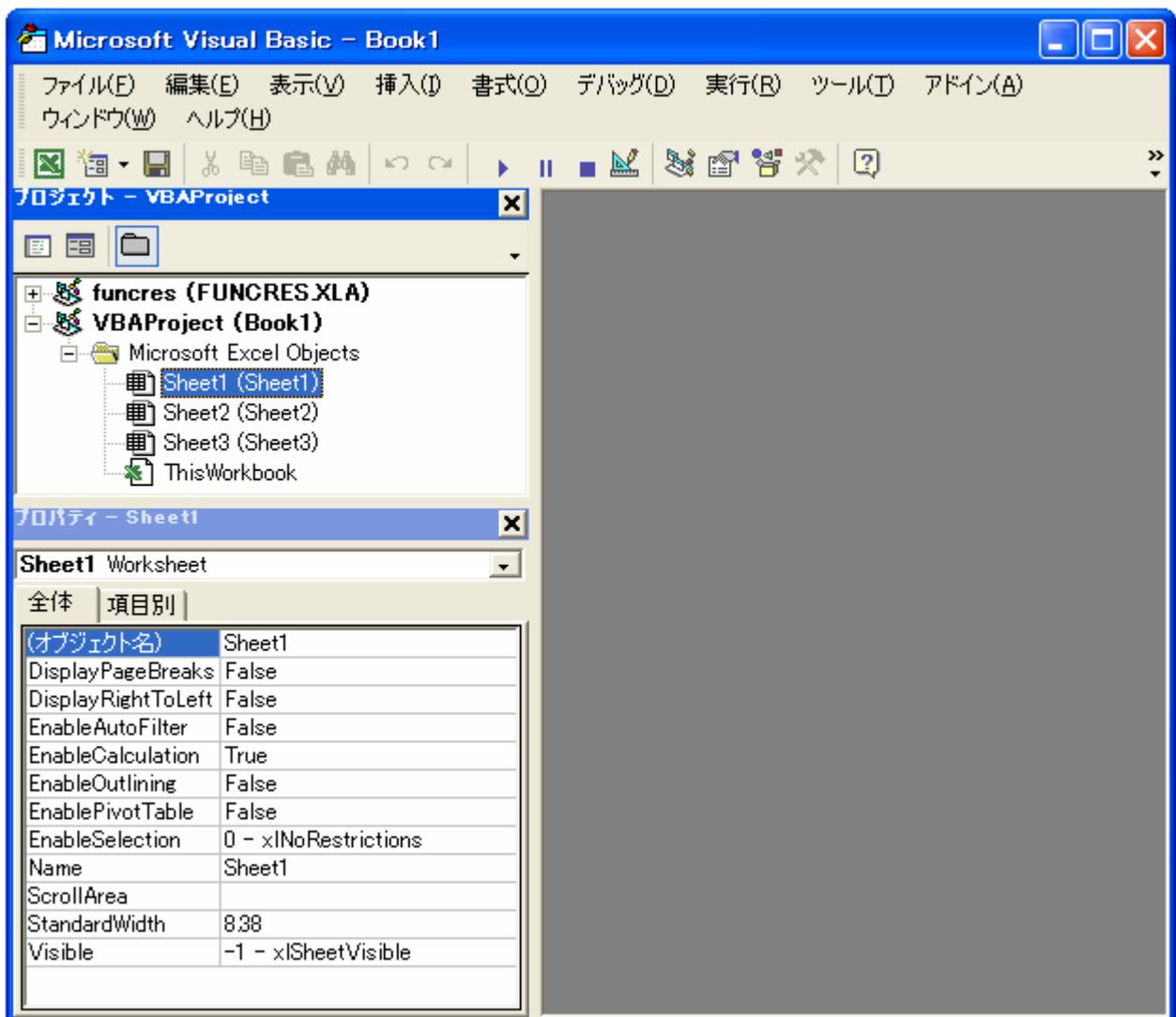


図3 「Visual Basic の世界」のウィンドウ

図3のVBAウィンドウは、図2のExcelウィンドウのメニューの「ツール」をクリックし、開かれるプルダウンメニューで「マクロ/Visual Basic Editor」をクリックすることで開くことができる。

「Visual Basic の世界」には、「フォーム」と呼ばれる部分と「標準モジュール」と呼ばれる部分がある。

「フォーム」は1つないし複数の「ユーザーフォーム」を含むものであり、各「ユーザーフォーム」は1つの「ダイアログボックス」と1つの「コード」で構成される。「ダイアログボックス」は、Excel ウィンドウ上に表示し、データの入力・表示等の処理を行うためのダイアログボックスを編集するためのものであり、対応した「コード」はダイアログボックスで実施すべき処理のマクロ（プログラム）を記述するためのものである。

「標準モジュール」とは、Excel ウィンドウの“裏”で計算したり何らかの処理を実施し、Excel ウィンドウ上に結果を表示させるためのマクロ（プログラム）を記述するためのものである。「標準モジュール」はマクロ（プログラム）のみを記述するため、“マクロ”と呼んでも過言ではないものである。このため、Excel ウィンドウのメニューの「ツール」をクリックし、開かれるプルダウンメニューで「マクロ／マクロ」をクリックすれば、「標準モジュール」に対して操作を行うためのダイアログボックスが表示される。「標準モジュール」には、複数の「モジュール」を含めることも可能であるが、普通は1つの「モジュール」で構成する。各「モジュール」は1つの「コード」で構成される。

「フォーム」の「ダイアログボックス」と「コード」及び「標準モジュール」の「コード」は、それぞれ独立したウィンドウで、図3のVBA ウィンドウの右下のグレーの部分に表示される。したがって、それぞれのウィンドウに必要事項を記入するためには、表示するウィンドウを切り替えて入力しなければならない。これら3種類のウィンドウは、完成してしまえば「編集」する必要がなく、表示する必要もなくなり、Excel ウィンドウの背後に隠れていてかまわないものである。

「ダイアログボックス」の編集は、「コントロール」と呼ばれる“部品”を多数貼り付けることで行う。「フォーム」の「コード」と「標準モジュール」の「コード」は、多数の「Sub プロシージャ」を含むものであり、直接キー入力により記入しなければならない。

各「Sub プロシージャ」は、それぞれが独立したマクロ（プログラム）であり、Excel ウィンドウからの指示、Excel ウィンドウ上に表示したダイアログボックスからの指示で起動するものである。また、起動した「Sub プロシージャ」からの指示で起動することもある。Excel マクロは、Visual Basic というプログラミング言語で記述しなければならない。

Excel ウィンドウ上に“UserForm1”という名前で作成されている「ダイアログボックス」を表示するためのマクロは

UserForm1.Show

であり、この命令は「標準モジュール」の「コード」の「Sub プロシージャ」に記述しなければならない。Excel ウィンドウ上に表示した「ダイアログボックス」を閉じるための命令は

Unload Me

であり、この命令は「フォーム」の「コード」の「Sub プロシージャ」に記述しなければならない。

2. Excel マクロのセキュリティ

マクロ（プログラム）は、かなり高度なことも行うことができ、悪意のある者がパソコンに害をなすマクロを作成し、電子メールに添付して送り付ける事件が後を絶たない。このため、マクロの使用に関してはセキュリティ対策が講じられている。

第1のセキュリティ対策は、マクロを有効にするか無効にするかのものであり、Excel ウィンドウのメニューの「ツール」をクリックし、開かれるプルダウンメニューで「セキュリティ」をクリックしたときに表示されるダイアログボックスで選択するものである。ここで、“セキュリティレベル”を“高”にすると、基本的にマクロは実行不可になり、パソコンは悪意あるマクロに対して安全となる。しかし、この場合は自分で作成したマクロも実行不可になってしまう。マクロを使用するときは“セキュリティレベル”を“中”にしなければならない。面倒であるかもしれないが、“セキュリティレベル”を状況に応じて切り替えるべきであろう。

“セキュリティレベル”を“中”にした状態で、「Visual Basic の世界」を含んだファイルを読み込もうとすると、“マクロを無効にする”か“マクロを有効にする”かが問われる。これが第2のセキュリティ対策であり、思い当たりのない限り“マクロを無効にする”を選択すべきであろう。マクロを使用する場合は当然“マクロを有効にする”を選択しなければならない。

3. ユーザーフォームの表示と終了

ここでは、最も基本的なユーザーフォームを作成し、Excel シート上に表示させるためのマクロを作成するための方法を説明する。

(1) ダイアログボックスを作成する

【手順1-1】 Excel ウィンドウのメニューの「ツール」をクリックし、開かれるプルダウンメニューで「マクロ/Visual Basic Editor」をクリックする。Excel ウィンドウとは別に、図3のVBA ウィンドウが開く。

【手順1-2】 VBA ウィンドウのメニューの「挿入」をクリックし、開かれるプルダウンメニューで「ユーザーホーム」をクリックする。図4と図5の2つのウィンドウが開く。ただし、図4はVBA ウィンドウのグレーの部分の左上に表示され、図5はディスプレイ上に表示されVBA ウィンドウの外に移動することもできる。

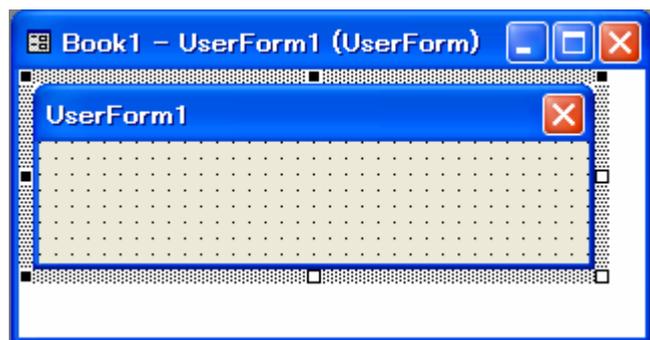


図4 ダイアログボックス作成シート



図5 ツールボックス

【手順1-3】 「ツールボックス」で1番右の列で上から2つ目のアイコン（コマンド ボタン）をクリックする。次に、図4のシートのグレーの部分でクリックする。すると、図6のように「コントロール」を「ダイアログボックス」に貼り付けることを意味する。

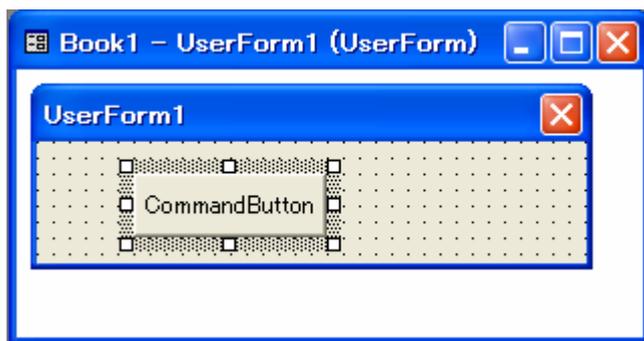


図6 コマンドボタンの貼り付け

同様な操作を繰り返すことで「ダイアログボックス」を完成させることができるが、ここではここまでとする。

図4と図6を比較したとき、図4ではシート全体が網掛け枠線で囲まれているが、図6ではボタンが網掛け枠線で囲まれている。図6で、ボタン以外の部分をクリックすれば、図4のようにシート全体が網掛け枠線で囲まれる。この網掛け枠線は、その段階でアクティブなものを示すものであり、何がアクティブであるかが重要な意味をもつことになる。

(2) コントロールを編集する

VBA ウィンドウの左側に2つのウィンドウが表示されるが、これまで説明してこなかった。1つは「プロジェクト」という名前のウィンドウであり、図1によく似たファイル全体の構成を表示するものである。もう1つは「プロパティ」という名前のウィンドウであり、ダイアログボックス作成用シート上で「アクティブ」なもののプロパティ（属性）を表示するためのものである。ダイアログボックスに貼り付けたコントロールをクリックすることで「アクティブ」なものを変更すると、「プロパティ」が変更されることを確認しなさい。シート全体あるいはシートに貼り付けたコントロールを編集するために、この「プロパティ」を使用する。

【手順2-1】 手順1-3で貼り付けたコントロール（コマンドボタン）をアクティブにする。このとき「プロパティ」は図7のようになる。

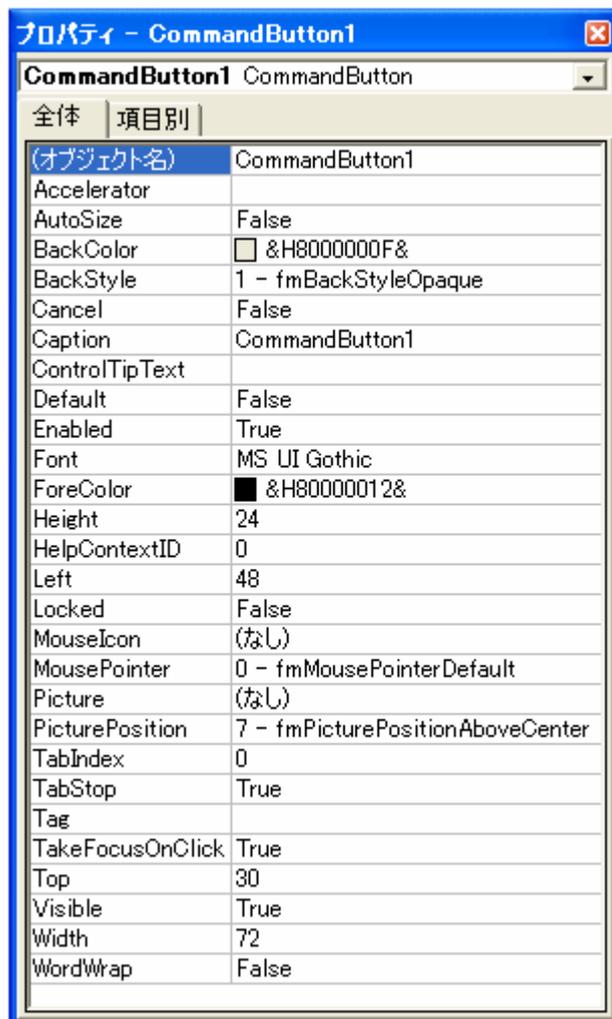


図7 コマンドボタンのプロパティ

【手順2-2】 「プロパティ」は2列になっており、左側が項目名であり、右側がその値である。

Caption という項目の値は「Command Button1」となっている。これは、図6のようにシート上に貼り付けたボタンの見出しを意味している。この「CommandButton1」の部分を「終了」に変更すると、シートに貼り付けたボタンの見出しが図8のように「終了」に変更される。

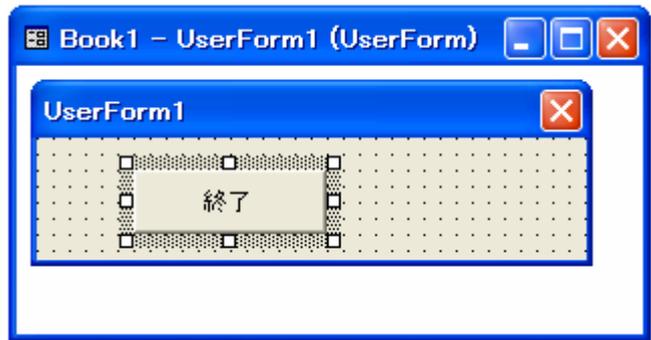


図8 見出しの変更

(3) オプションボタンに Sub プロシージャを関係付ける

【手順3-1】 コマンドボタンをアクティブにした状態（図8の状態）で、このボタンをダブルクリックする。すると、図9の「UserForm1」の「コード」を表すウィンドウが表示される。さらに、中身のない Sub プロシージャが記述される。



図9 「UserForm1」の「コード」を表すウィンドウ

【手順3-2】 図10のように「Private Sub …」行と「End Sub」行の間に

Unload Me

という行を入力する。



図10 Sub プロシージャの作成

この例のように「Sub」行で始まり

(Private と())の意味は当面无視する。)「End Sub」行で終わる部分が「Sub プロシージャ」であり、「Sub」に続く文字列が Sub プロシージャの名称である。

この Sub プロシージャは、名称に「_Click」が付いているので特殊な機能を果たすものとなり、クリックされたときに実施される Sub プロシージャを意味する。名称が「CommandButton1_Click」であるので、「CommandButton1」がクリックされたときに実施される Sub プロシージャになる。

すなわち、図8のダイアログボックスで見出しが「終了」であるコマンドボタンがクリックされたとき、ダイアログボックスを閉じることを意味する。

VBA ウィンドウのメニューの「表示」をクリックし、開かれるプルダウンメニューで「オブジェクト」をクリックすれば、図8の「ダイアログボックス」ウィンドウが表示され、「コード」をクリックすれば、図10の「コード」ウィンドウが表示される。

(4) UserForm1 を表示するマクロを作成する

【手順 4-1】 VBA ウィンドウのメニューの「挿入」をクリックし、開かれるプルダウンメニューで「標準モジュール」をクリックすれば、図 11 のような標準モジュールの「コード」を表すウィンドウが表示される。

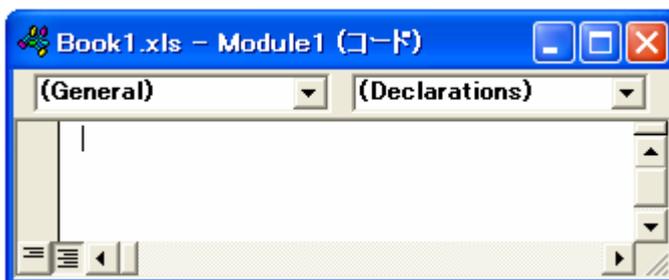


図 11 標準モジュールの「コード」を表すウィンドウ

【手順 4-2】 「sub abc」と入力し Enter キーを押すと、図 12 のように中身の無い Sub プロシージャが作成される。

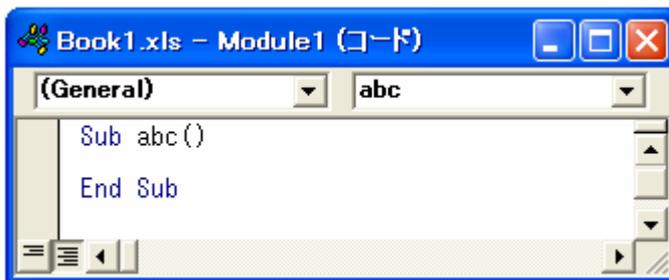


図 12 Sub プロシージャの入力 1

「()」と「End Sub」が自動的に追加され、「sub」が「Sub」と変化する。

【手順 4-3】 図 13 のように「Sub …」行と「End Sub」行の間に

UserForm1.Show

という行を入力する。

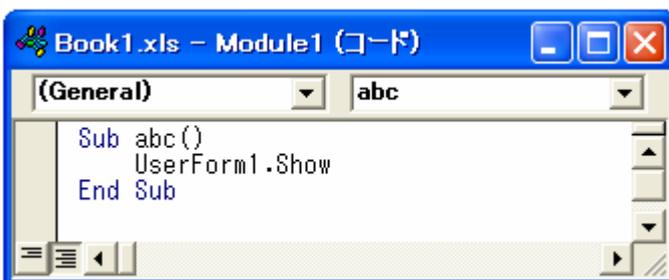


図 13 Sub プロシージャの入力 2

これで、UserForm1 という名前の付いたユーザーフォームを起動させるためのマクロ (abc という名前の付いた Sub プロシージャ) が完成したことになる。

4. 「プロジェクト」ウィンドウ

図 10 と図 13 を比較したとき、その違いはタイトルバーが「UserForm1 (コード)」と「Module1 (コード)」となっているだけである。特に、“最大表示”してあるときにはそれらが区別しにくい。

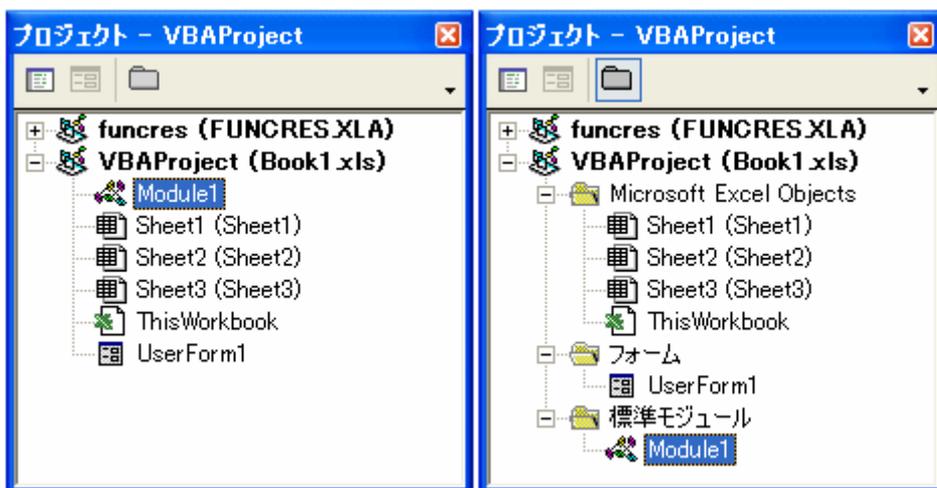


図 14a 「プロジェクト」ウィンドウ 1

図 14b 「プロジェクト」ウィンドウ 2

これらを区別するためには、VBA ウィンドウの左上に表示される図 14 の「プロジェクト」ウィンドウを利用する。

「プロジェクト」ウィンドウには「コードの表示」「オブジェクトの表示」「フォルダの切り替え」の3つのボタンがあり、「フォルダの切り替え」ボタンを押すことで図 14a と図 14b が切り替えられる。項目をリストアップしただけの図 14a と、フォルダ形式で表示した図 14b があるが、その差は見た目だけである。

図 14 のように「標準モジュール」に含まれる Module1 が選択（クリックすればよい）されているときは、「オブジェクトの表示」ボタンが使用不可になる。その他の UserForm1 や Sheet1 等が選択されているときは「オブジェクトの表示」ボタンが有効である。UserForm1 や Sheet1 をダブルクリックするか、クリックして選択しておいて「オブジェクトの表示」ボタンをクリックすれば、ダイアログボックスや Excel シートを表示することができる。なお、Module1 をダブルクリックすると Module1 の「コード」が表示される。

UserForm1 や Module1 を選択し、「コードの表示」ボタンをクリックすることで、それぞれの「コード」を表示することができる。表示されている「コード」が何に対するコードであるかを知るためには、「プロジェクト」ウィンドウで何が選択されているかを見ればよい。

5. Sub プロシージャの起動

(1) 標準的な Sub プロシージャの起動方法

図 13 のように「標準モジュール」上に作成した Sub プロシージャを起動させるための方法は、複数用意されている。

標準的な起動方法は、Excel ウィンドウで、メニューの「ツール」をクリックし「マクロ / マクロ」をクリックする。

そのとき、図 15 のダイアログボックスが表示される。

マクロ名欄には、操作中の Excel ファイルに含まれている「標準モジュール」の Sub

プロシージャ名リストが表示される。起動すべき Sub プロシージャ名を選択し、右側にある「実行」ボタンをクリックすれば、その Sub プロシージャを起動することができる。

図 15 のダイアログボックスで、「編集」ボタンをクリックすれば図 13 のように「標準モジュール」の「コード」が表示され、Sub プロシージャの編集が可能となる。

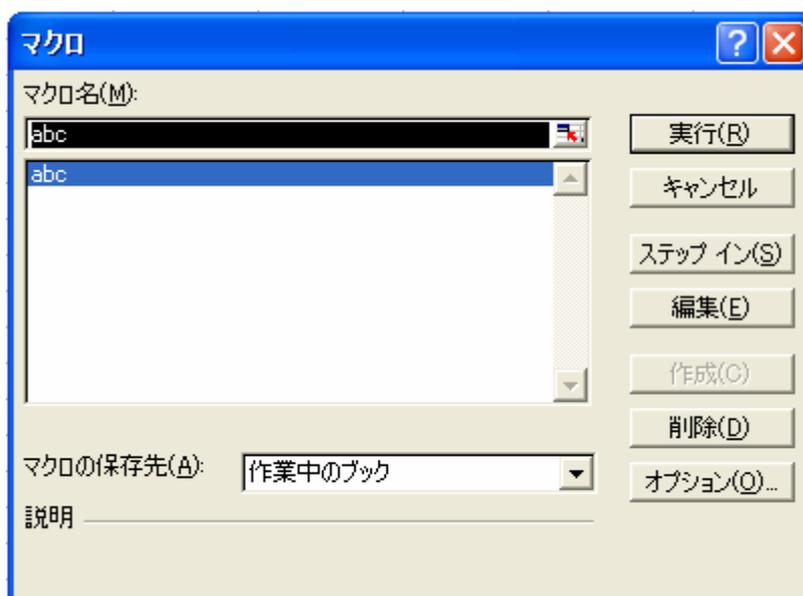


図 15 マクロ操作ダイアログボックス

(2) ショートカットキーによる Sub プロシージャの起動方法

図 15 のダイアログボックスで、「オプション」ボタンをクリックすれば、図 16 のような「マクロ オプション」ダイアログボックスが表示される。これは、図 15 で選択された Sub プロシージャを起動させるための「ショートカットキー」を設定するためのものである。

「ショートカットキー」とは、キー操作により Excel に何らかの処理をさせるためのものであり、コントロール (Ctrl) キーを押しながら半角のキーを押すものである。この操作は「Ctrl + c」のように略記される。

代表的なデフォルトショートカットキーとしては次のものがあり、これらは最初から Excel で定義されている。

Ctrl + a : すべて選択	Ctrl + c : コピー	Ctrl + v : 貼り付け	Ctrl + x : 切り取り
Ctrl + n : 新規作成	Ctrl + o : 開く	Ctrl + s : 上書き保存	Ctrl + p : 印刷

これらと同様に、起動すべき Sub プロシージャにショートカットキーを割り付けることで、(1) の標準的方法に依らずに Sub プロシージャを起動させることができる。

図 16 のダイアログボックスで、「Ctrl+」の右側にある枠内に半角英字を入力する。例えば、「e」を入力し、「OK」ボタンをクリックすれば、図 15 で選択した Sub プロシージャをショートカットキー「Ctrl + e」で起動することができる。

ここで定義したショートカットキーは、デフォルトショートカットキーに優先するが、よく使うデフォルトショートカットキーとは重複しないように設定すべきであろう。また、複数の Sub プロシージャに同一のショートカットキーを割り当てると、混乱が生じるので注意すべきである。

(3) マクロボタンによる Sub プロシージャの起動方法

Sub プロシージャを起動させるための「マクロボタン」を作成することも可能であり、以下では「マクロボタン」の作成方法を簡単に説明する。

【手順 1】 Excel ウィンドウのメニューバーの「ツール」をクリックし、開かれるプルダウンメニューの「ユーザー設定」をクリックする。「ユーザー設定」ウィザードが開くので、「ツールバー」タブをクリックする。

【手順 2】 「新規作成」ボタンをクリックし、開かれる「新しいツールバー」ウィザードで「OK」ボタンをクリックする。すると、図 17 のようにボタンを含まないツールバー「ユーザー設定 1」が表示される。ただし、「ユーザー設定」ウィザードの「ツールバー」選択欄に「ユーザー設定 1」が既にあるときは、「ユーザー設定 1」

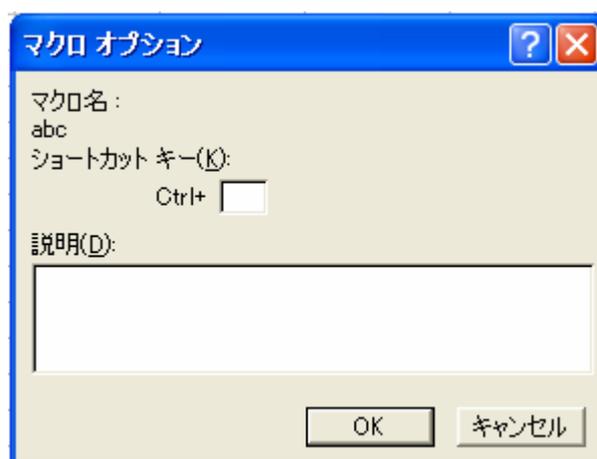


図 16 マクロ オプション

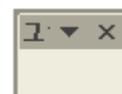


図 17

をクリックする。

【手順3】 ユーザー設定ウィザードの「コマンド」タブをクリックし、図18のように「分類」欄の「マクロ」をクリックする。右側に表示される「ユーザー設定ボタン」を、上の手順2で表示させたツールバー「ユーザー設定1」にドラッグし、図19のようにする。

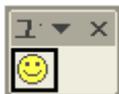


図 19

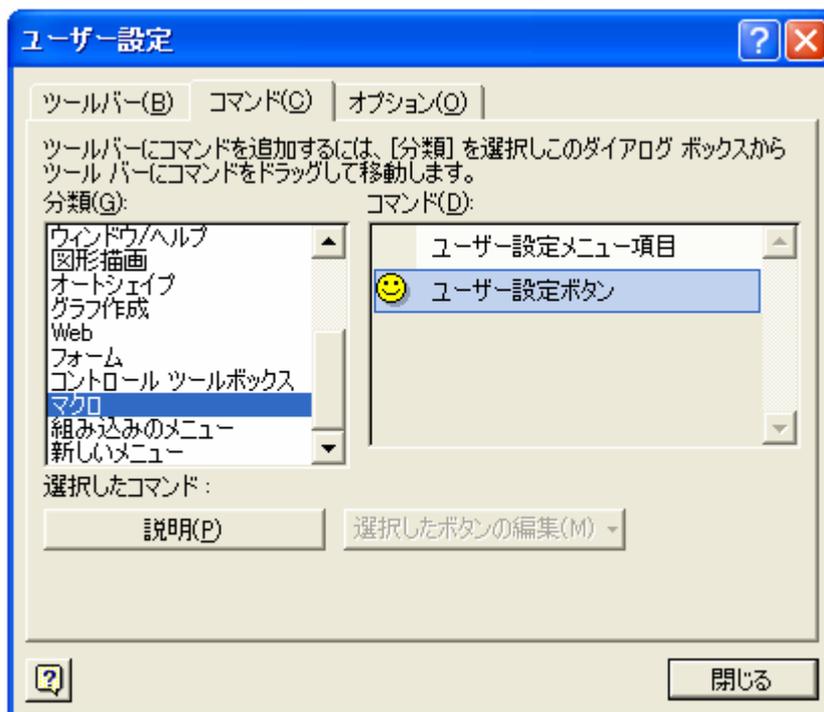


図 18 「ユーザー設定」ウィザード

【手順4】 「ユーザー設定」ウィザードの「選択したボタンの編集」ボタンをクリックし、開かれるメニューで「マクロの登録」をクリックする。

【手順5】 「マクロの登録」ウィザードで、作成したマクロボタンに登録すべき「Sub プロシージャ」を選択し、「OK」ボタンをクリックする。

【手順6】 さらに、「ユーザー設定」ウィザードの「選択したボタンの編集」ボタンをクリックし、図20のように開かれるメニューの「名前」欄にマクロボタンに付ける「名前」を記入し、「ボタンイメージの変更」にマウスを合わせたときに表示されるボタンの候補の中から好みのボタンを選択し、クリックする。

【手順7】 「ユーザー設定」ウィザードの「閉じる」ボタンをクリックする。



図 20 「選択したボタンの編集」メニュー