

● 1回作ったプロシーチャをどう呼び出せるか（どう使いまわせるか？・再利用できるか？どう人間がサボれるか？）の関連図

この色の矢印は、
デフォルト（既定）
もくそもなく、
呼び出しOKです。→

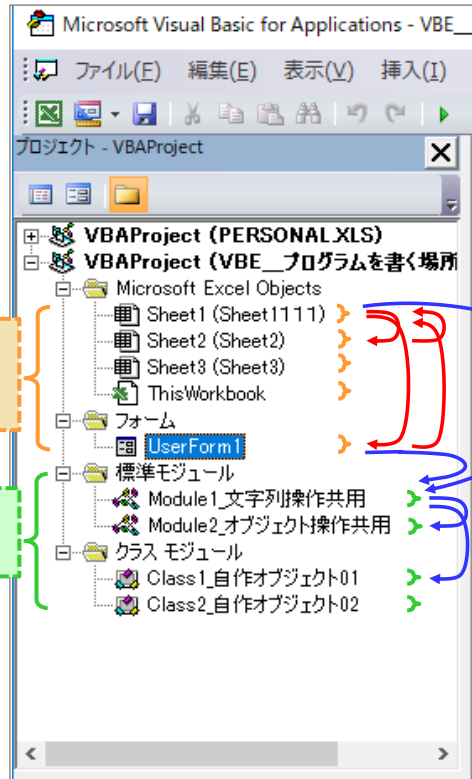
この色の矢印は、
デフォルト
（既定）では、
呼び出しNGです。→

※ただし、ある細工をすると呼び出せます。
（クラスモジュールからだけはそれでも不可です。また、イベントプロシーチャを呼び出す場合は、引数が必要な場合はタミーの引数を指定すると動く場合があります。が、それでもどうにもエラーになることもあります。）

大きなくくりとしてのイメージ。→

「クラスオブジェクト」のモジュール（イベントプロシーチャが書ける。）

その他のモジュール（イベントプロシーチャが書けない。）



モジュールやプロシーチャ個々の小さなくくりとしてのイメージ。←

● 「マクロの自動記録」が「限定的」で、汎用的にならない理由
マクロの自動記録機能によって自動的に書き込まれたSubプロシーチャは、確かに「VBA」で書かれるには違いないのですが、ただし、そのうちの「オブジェクトとプロパティ、メソッド、列挙体、定数」あたりしか自動で書き込まれません。一般的なプログラムに必要な、「変数、繰り返し文、分岐の文、関数、その他、いろいろなもの」が抜け落ちていきます。通常、汎用的なプログラムを書くにはそのすべてが必要ですが、「マクロの自動記録」では、それが抜け落ちていきます。なので、「限定的」となります。ただし、だからといって役に立たないものではありません。使い方次第で十分に戦力になります。

● ExcelやWordの「マクロの記録機能」と「標準モジュール」・「プロシーチャ」との関係についての補足

ExcelやWordでは、「マクロの記録機能」でマクロを作ると、自動的に標準モジュールが作成（＝挿入）されて、その中に記録内容に沿った「Subプロシーチャ」が自動的に記述されます。そしてそれが「マクロ」になります。

逆に言うと、「手作業で」、「標準モジュールを作って（＝挿入して）その中にSubプロシーチャを書けば」、それもいたるところで「マクロ」として認識され、実行する＝呼び出すことができるようになります。
※Accessの場合は、少し事情が異なります。

※Subプロシーチャの「Sub」の前に「Private」を付加すると「マクロ」として認識することはできなくなります。なお、Functionプロシーチャも「マクロ」としては認識されません。ExcelやWordの中で「マクロ」と認識されるのは「Sub」で始まって「End Sub」で終わるプロシーチャだけです。

基本、標準モジュールに「文字ベースの値を返すFunctionプロシーチャ」を作ると、ワークシートの数式から、それを呼び出して、あたかもワークシート関数のように数式内で計算や文字操作をさせることができます。

そしてそれを、Personal.xlsb（またはxls）の標準モジュールに作るとすべてのExcelファイルから、ワークシートの数式から呼び出せるようになります。

ただし、Personal.xlsb（またはxls）に作った、「値を返すFunctionプロシーチャ」を、呼び出す場合は、Functionプロシーチャ名だけ書いたのではセルに「#NAME?」（名前が持つかりませんよ？的な意味です）と表示されてしまって正常に呼び出せません。

その場合は、例えばPersonal.xlsbの場合なら、「=Personal.xlsb!Functionプロシーチャ名(引数・・・)」という形で、「Personal.xlsb!」を付加して数式に書いて呼び出す正常に呼び出せます。

標準モジュールに書いた「Subプロシーチャ」や「Functionプロシーチャ」は、以下の場所からも呼び出せます。（数式バー以外は全部Subです。）

- リボンの自作ボタン
- クイックツールバーの自作ボタン
- 数式バー（値を返すFunctionのみ）
- シートやユーザーフォーム上に作った自作のコマンドボタン
- 「開発」タブの「マクロ」ボタン