

Staticとモジュール変数が前回の状態を持ち越す

コイ丸先輩とシャケもんが、設定変更の「つもり」をたどる

Copyright © 2026 LWP 山中 一弘

本資料は、出典を明記いただければ、商用・非商用を問わず、ご自由に複製・改変・再配布していただけます。なお、著作権表示は改変せず、そのまま記載してご利用くださいますようお願いいたします。

ストーリー

大文字も検索したくて、設定をTrueへ変えたコイ丸先輩。ところが結果は0件のままです。相談を受けたシャケもんは、今回の引数だけでなく、その前の呼び出しも並べてみます。二人とも知っているStaticが、思わぬところで話をつなぎ始めました。

VBAの手続きを分けて書く人に向けた、架空の説明用の物語です。対象はWindowsデスクトップExcel VBAとTakiLib Ver3.5.4の対象ソース。図中の件数はソース読解による予測で、今回のVBA実測値ではありません。

1 設定を変えたのに0件？ その意外な理由を探ろう

設定を変えたのに0件？ その意外な理由を探ろう

①

?

大文字も拾いたくて、
Trueに変えたんだ。
なのに**0件**？

コイ丸先輩

今回の引数と、
ひとつ前の呼出しを
見せてください。

シャケもん

毎回、同じ設定で
呼んでいるつもり
だったよ。

コイ丸先輩

2回目の呼出し

- 対象：A
- Trueを指定
- 一致：**0件**

その「毎回」が、
手掛かりかもしれません。

シャケもん

今回の引数だけでは、
条件がそろわない？

図1 | Trueを渡したことと、保存済みの設定が更新されたことを分けて確認します。

2 使える場所と、値が残る時間は別

使える場所と、値が残る時間は別

②

前の値を残すための
Staticだね。

はい。名前の届く範囲と、
値の寿命は別です。

毎回1を足すカウンター

変数	1回目	2回目	3回目
通常のDim	1	1	1
Static	1	2	3

初期状態から連続で呼ぶ例。リセット等で初期化。



名前を使える場所：
その手続き内



Staticの値：
呼出しをまたいで保持

場所が狭くても、
記憶は続くわけだ。

局所変数でも、
前回の続きになる
んです。

図2 | 初期値0のカウンターへ毎回1を足す例です。通常の非Static手続き内のDimと、手続き内Staticを比べています。

3 Trueを渡しても、設定は更新されない



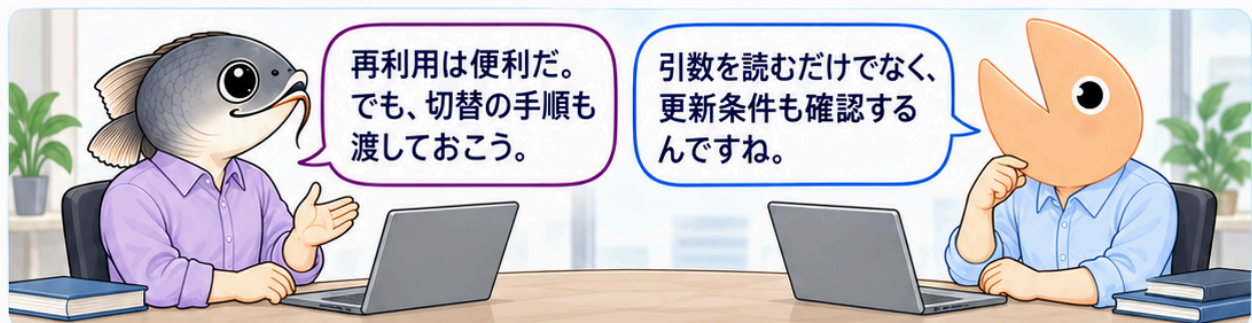
図3 | 設定を代入する処理は、パターンを指定した場合の分岐の中にあります。

4 パターンを指定すると、設定も更新される

パターンを指定すると、設定も更新される ④



同じ実行状態で、この順に呼ぶ

1回目
0件2回目
0件3回目
1件

対象版regExtractのソース読解による予測

図4 | パターンも明示すると再生成・再設定の分岐を通ります。この更新条件は、対象版regExtractの実装によるものです。

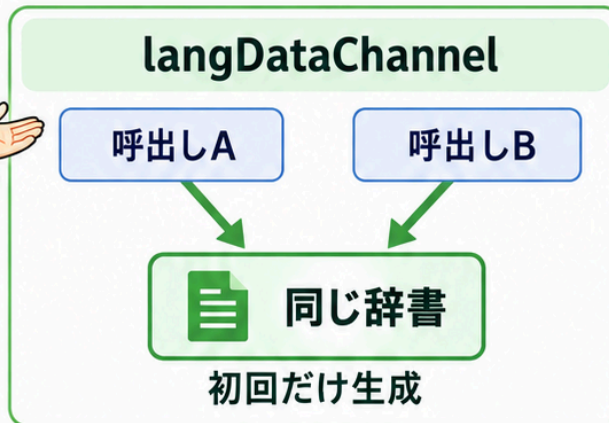
5 毎回受け取っても、同じ辞書

5

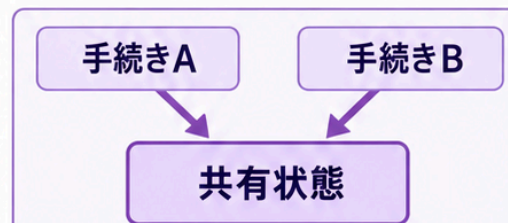
毎回受け取っても、同じ辞書

引き継ぎ用の共有箱も、
毎回作り直すわけでは
ないね。

langDataChannelは、
同じ辞書への参照を
返します。



標準モジュールのPrivate変数



一時置き場のつもりか、
共有箱のつもりか。
そこを合わせよう。



上の辞書と下の変数は、別の例・別の状態



図5 | 同じ辞書の再利用と、モジュール変数の共有は別の例です。共有したい範囲と、状態を残す目的を確かめます。

6 引き継ぐのは、コードと状態の約束

引き継ぐのは、コードと状態の約束 ⑥



図6 | どの値を持ち越し、いつ設定し直すのかまで分かれば、次の人が同じ条件を再現できます。

まとめ

コイ丸先輩が変えたのは、2回目の呼び出しに渡す引数でした。ところが対象版のregExtractは、パターンを省略すると、前回保存した正規表現オブジェクトをそのまま使います。シャケもんが追ったのは、値を代入する分岐と、その分岐を通る条件です。

Staticやモジュール変数による状態の保持は、再利用や共有に役立ちます。保守では、今回の引数と前回までの状態と一緒に確認します。引き継ぎには、初期化・呼び出し順・再設定・リセットの条件を残しましょう。

今回の仕掛け

Staticで宣言した手続き内変数は、名前を使える範囲がその手続き内でも、呼び出しをまたいで値を保持します。標準モジュールのPrivate変数は、そのモジュール内の複数手続きから扱えます。Publicは公開範囲を広げる指定であり、値の寿命を長くする指定ではありません。

保持される値は永続保存ではありません。VBEでのリセット等で初期化されます。クラスモジュールではインスタンスの寿命にも関係するため、「Excelを閉じるまで必ず残る」と一律に説明しないようにします。

langDataChannelは、Staticの辞書参照がNothingの場合だけ辞書を生成します。複数回取得しても、独立した新しい辞書にはなりません。ある変数へNothingを代入して参照を外しても、ほかの変数が同じ辞書を参照していれば、辞書本体まで直ちに消えるとは限りません。

図5下段は共有範囲の概念図です。対象ソースにはCSV区切り文字を扱うモジュール変数TK3_CHAR_COMMAもありますが、それだけでCSV処理の不具合や並行実行時の競合を断定するものではありません。

今回のソースコードのサンプル

サンプルコード 同じ実行の中で、3回の設定を比べる

```
01 Public Sub DemoRegExtractState()
02     Call regExtract("A", "a", False, True)
03     Call regExtract("A", a_ignorcase:=True)
04     Call regExtract("A", "a", True, True)
05 End Sub
```

関数の説明

regExtractは、正規表現の一致件数・位置・内容などを出力するSubです。戻り値を返すFunctionではありません。a_ignorcaseは対象ソースの引数名の綴りです。Trueなら大文字・小文字を区別しない設定にします。

コードの説明

最初はパターンaを大文字・小文字を区別する設定で保存します。次はパターンを省略してTrueだけを渡します。最後はパターンも明示して設定し直します。対象文字列Aは3回とも同じです。

入力・概要

TakiLib Ver3.5.4の対象ソースを同じVBAプロジェクトへ導入した検証用ブックの標準モジュールへ、上のSub全体を貼り付けます。DemoRegExtractStateを1回実行し、既定のDebug出力先を使う環境ではイミディエイトウィンドウのCountを順番に読みます。出力先を変更している環境では、その設定も確認してください。

結果

対象ソースから予測する一致件数は、0、0、1です。今回このコードをExcelで実行した実測値ではありません。初回からパターンを省略すると、正規表現オブジェクトがまだ生成されていないため、この例では最初の呼び出しで必ずパターンを指定します。

読むポイント

引数を省略した場合に「既定値へ戻す」のか「前の設定を使う」のかは、APIの実装・契約によります。Staticという宣言だけでは、どの引数で設定が更新されるかまで決まりません。

対象ソースの保持・更新部分

サンプルコード regExtractの内部抜粋

```
01 Static xreg As Object
02 If Not valuesMissing(a_pattern) Then
03     Set xreg = langRegFactory
04     With xreg
```

```

05      .Pattern = a_pattern
06      .IgnoreCase = a_ignorecase
07      .Global = a_global
08  End With
09 End If

```

コードの説明

xregはStaticで保持されます。valuesMissingでパターン指定の有無を判定し、指定がある場合だけ langRegFactoryでオブジェクトを作り直して、3つの設定を代入します。省略時はこのIfを通りません。

入力・概要

TakiLibのregExtract内部から、保持・更新に直接関係する部分だけを抜粋しています。引数や補助関数を含む手続き全体ではないため、抜粋単体では実行できません。上の呼び出しサンプルと、ソース読解用の抜粋を区別してください。

出典メモ

VBAのStaticと値の寿命は、[Microsoft Learn: Static statement](#)を参照しています。

TakiLib Ver3.5.4 (2026年9月6日出力版) のregExtract、langDataChannel、TK3_CHAR_COMMAを参照しました。図3・4の予測は、この版のregExtractにあるパターン指定時の分岐に基づきます。別版では実装を確認してください。速度向上や並行実行の安全性を保証する検証は行っていません。

コード抜粋の出典: Copyright (c) 2021 Kazuhiro Yamanaka. TakiLibコードの利用には、次のMIT Licenseが適用されます。

TakiLibのコードライセンス

The MIT License

Copyright (c) 2021 Kazuhiro Yamanaka

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.