

## 本書をご利用いただく前に

本書について …… 008

VBA エキスパートの概要 …… 009

学習環境 …… 013

1. 環境について …… 013
2. Excel 2007 の環境設定 …… 014
3. Excel 2003 の環境設定 …… 017
4. その他の設定 …… 021
5. 学習用データのダウンロードサービス …… 022

## 1 VBAの基礎

1-1 VBA の基本構文 …… 2

1-2 変数 …… 3

1-3 セルの操作 …… 4

1-4 ステートメント …… 5

1-5 シートとブックの操作 …… 7

1-6 デバッグ …… 9

## 2 Visual Basic Editor (VBE) の操作

2-1 イミディエイトウィンドウ …… 12

2-2 ウォッチウィンドウ …… 19

2-3 呼び出し履歴 …… 24

## 3 プロシージャ

3-1 他のプロシージャを呼び出す …… 28

3-2 Function プロシージャ …… 29

3-3 プロシージャの引数 …… 32

参照渡しと値渡し …… 33

省略可能な引数を定義する …… 37

## 4 変数と配列

4-1 配列とは …… 42

4-2 動的配列 …… 45

4-3 静的変数 …… 48

4-4 オブジェクト変数 …… 50

4-5 ユーザー定義型 …… 53

## 5 イベント

5-1 イベントとは …… 60

イベントとは何か …… 60

イベントプロシージャの挿入方法 …… 60

イベントを利用する際の注意 …… 64

5-2 ブックのイベント …… 67

5-3 シートのイベント …… 72

## 6 ステートメント

- 6-1 Select Case ステートメント ..... 80
- 6-2 Do...Loop ステートメント ..... 85
- 6-3 For Each...Next ステートメント ..... 91
- 6-4 Exit ステートメント ..... 94

## 7 関数

- 7-1 配列を操作する関数 ..... 98
- 7-2 データを判定する関数 ..... 100
- 7-3 日付を操作する関数 ..... 103

## 8 エラーへの対応

- 8-1 エラー処理 ..... 106
  - 文法エラー ..... 106
  - 論理エラー ..... 107
- 8-2 エラーへの対応 ..... 112
  - エラーが発生したら別の処理にジャンプする ..... 112
  - どんなエラーが発生したか調べる ..... 114
  - 発生したエラーを無視する ..... 118
- 8-3 エラー対策のポイント ..... 121

## 9 UserForm

- 9-1 UserForm とは ..... 124
  - 設計 ..... 124
  - コーディング ..... 128
  - 表示 ..... 131
- 9-2 よく使うコントロール ..... 133
  - ラベル ..... 133
  - コマンドボタン ..... 135
  - テキストボックス ..... 137
  - リストボックス ..... 140
  - コンボボックス ..... 142
  - チェックボックス ..... 144
  - オプションボタン ..... 147
  - イメージ ..... 149
- 9-3 UserForm のイベント ..... 154
  - Initialize イベント ..... 154
  - QueryClose イベント ..... 157

## 10 メニューの操作

- 10-1 コンテキストメニューとは ..... 162
- 10-2 コマンドを登録する ..... 165
  - 区切り線を付ける ..... 167
  - 追加する位置を指定する ..... 171
  - コマンドにアイコンを表示する ..... 172
  - コマンドにチェックマークを付ける ..... 173

- コマンドに説明を付ける …… 175
- 10-3 コマンドを削除する …… 176

## 11 Windowsの機能を利用する

- 11-1 OLE オートメーションとは …… 182
  - OLE オートメーションの使い方 …… 182
  - OLE オートメーションの注意 …… 186
  - 事例 1：Internet Explorer を操作する …… 186
  - 事例 2：Windows Script Host を利用する …… 190
- 11-2 API とは …… 191
  - API の概念 …… 191
  - API の使い方 …… 191
  - API の注意点 …… 193
  - 事例 1：ゴミ箱に移動する …… 197

## 12 レジストリの操作

- 12-1 レジストリ の概念 …… 200
  - レジストリエディタの起動 …… 200
- 12-2 操作できる場所 …… 202
- 12-3 レジストリに登録する …… 203
- 12-4 レジストリを取得する …… 204
- 12-5 レジストリを削除する …… 206
- 12-6 レジストリを利用したマクロ例 …… 207
  - 前回の表示位置を再現する …… 207

- 前回の設定を再現する …… 209
- 同じファイルを処理しない …… 212

## 13 ファイルの操作

- 13-1 テキストファイルの操作 …… 216
  - ファイルを開くという意味 …… 216
  - テキストファイルを開く …… 217
  - テキストファイルを閉じる …… 218
  - テキストファイルから 1 行読み込む …… 219
  - ファイルに書き込む …… 221
  - ファイルをすべて読み込む …… 223
- 13-2 ファイルのコピーと移動 …… 225
- 13-3 ファイルの削除 …… 228
- 13-4 ファイルの存在確認 …… 229
- 13-5 カレントフォルダ …… 232
- 13-6 フォルダの操作 …… 235

索引 …… 237

## 3-1 他のプロシージャを呼び出す

モジュール内に存在する別のプロシージャを呼び出すには、**Call** という命令を使います。次のマクロ「Sample1」は、セル A1 に「100」を入力した後で、プロシージャ「Sample2」を呼び出します。

```
Sub Sample1 ()
    Range("A1").Value = 100
    Call Sample2
End Sub

Sub Sample2 ()
    Range("B1").Value = Range("A1").Value * 2
End Sub
```

	A	B	C
1	100	200	
2			
3			

他のプロシージャを呼び出す Call は、省略することも可能です。

```
Sub Sample1 ()
    Range("A1").Value = 100
    Sample2
End Sub

Sub Sample2 ()
    Range("B1").Value = Range("A1").Value * 2
End Sub
```

しかし、ただプロシージャ名を記述するのではなく「別のプロシージャを呼び出している」ということを分かりやすく表すためにも、Call は省略しない方がいいでしょう。なお、本書では、Sub プロシージャを呼び出すときは Call を付け、Function プロシージャを呼び出すときは Call を省略して解説します。

## 3-2 Functionプロシージャ

「Sub マクロ名 ()」で始まり、「End Sub」で終わるプロシージャを Sub プロシージャと呼びます。Sub プロシージャを実行すると、プロシージャに記述された VBA の命令が実行されますが、「End Sub」まで実行すると終了します。実行した結果を、自動的に何かすることはありません。それに対して、「Function マクロ名 ()」で始まり、「End Function」で終わる Function プロシージャは、**実行の結果を呼び出し元に返す**働きをします。

図のように、ワークシートの A 列に税抜き金額が入力されていて、消費税込みの金額を B 列に入力するようなマクロでは、税込み金額の計算部分だけを Function プロシージャとして独立させると、税率の変化や計算方法の変化などに対応しやすくなります。

	A	B	C
1	税抜き金額	税込み金額	
2	1,000		
3	2,000		
4	3,000		
5	4,000		
6			
7			
8			

計算の元になる数値を 1.05 倍して返す Function プロシージャを作ってみましょう。

Function プロシージャの書式は次の通りです。

**Function プロシージャ名 (受け取る値) As 返すデータ型**

プロシージャ名は、Sub プロシージャと同じように、意味のある名前を付けてください。ここでは「Tax」とします。

**Function Tax (受け取る値) As 返すデータ型**

この Function プロシージャが返すのは税込みの金額です。単位は円で、小数点がつくことはありません。そこで、Function プロシージャが返す値は「長整数型 (Long)」とします。なお、計算の結果に