

Microsoft  
**Excel**®  
**VBAプログラミング実践**  
Office 2021/2019/2016/Microsoft 365 対応

**練習問題・総合問題  
標準解答**

練習問題 標準解答 .....	1
総合問題 標準解答 .....	4

# 練習問題 標準解答

## 第3章 練習問題

- ① VBEに切り替える
- ② 《挿入》をクリック
- ③ 《標準モジュール》をクリック
- ④ モジュール「Module5」に次のプロシージャを入力

### ■「練習」プロシージャ

```
1. Sub 練習()  
2.   Dim Myrange As Range  
3.   Dim Nen As Integer  
4.   Dim Tuki As Integer  
5.   Dim Hi As Integer  
6.   Dim Hiduke As Date  
7.   Set Myrange = Range("B4")  
8.   Range("B4:D34").ClearContents  
9.   Range("D4:D34").Interior.ColorIndex = xlNone  
10.  Nen = Range("年").Value  
11.  Tuki = Range("月").Value  
12.  For Hi = 1 To 31  
13.      Hiduke = DateSerial(Nen, Tuki, Hi)  
14.      If Tuki <> Month(Hiduke) Then Exit For  
15.      Myrange.Value = Hiduke  
16.      Myrange.Offset(, 1).Value = Format(Hiduke, "aaa")  
17.      If Hiduke = Date Then Myrange.Offset(, 2).Interior.Color = vbMagenta  
18.      Set Myrange = Myrange.Offset(1)  
19.  Next Hi  
20.  Set Myrange = Nothing  
21. End Sub
```

## ■プロシージャの意味

1. 「練習」プロシージャ開始
2. Range型のオブジェクト変数Myrangeを使用することを宣言
3. 整数型の変数Nenを使用することを宣言
4. 整数型の変数Tukiを使用することを宣言
5. 整数型の変数Hiを使用することを宣言
6. 日付型の変数Hidukeを使用することを宣言
7. オブジェクト変数Myrangeにセル【B4】への参照を代入
8. セル範囲【B4:D34】の値をクリア
9. セル範囲【D4:D34】の塗りつぶしをなしにする
10. 変数Nenにセル【年】の値を代入
11. 変数Tukiにセル【月】の値を代入
12. 変数Hiが1から31になるまで処理を繰り返す
13. 変数Hidukeに、年が変数Nen、月が変数Tuki、日が変数Hiの日付を代入
14. 変数Tukiの値と変数Hidukeの月が等しくない場合(変数Hidukeの月が変わった場合)は繰り返し処理を抜け出す
15. オブジェクト変数Myrangeのセルに変数Hidukeの値を入力
16. オブジェクト変数Myrangeのセルの1列右のセルに、変数Hidukeの値を表示形式「aaa」に変換して入力
17. 変数Hidukeの値が現在の日付と等しい場合はオブジェクト変数Myrangeのセルの2列右のセルの背景色をピンクに設定
18. オブジェクト変数Myrangeに1行下のセルへの参照を代入
19. 変数HiにHi+1の結果を代入し、12行目に戻る
20. オブジェクト変数Myrangeを初期化
21. プロシージャ終了

※コンパイルし、上書き保存しておきましょう。

### ⑤ Excellに切り替える

### ⑥ セル範囲【F10:F11】、セル範囲【H10:H11】の右側にある上下のスピンボタンをクリック

※ボタンには、作成したプロシージャが実行されるように登録してあります。

※「予定」に任意の内容を入力したあとに年、月を変更すると、入力した内容が削除されることを確認しておきましょう。

※今月のカレンダーを表示し、本日の「予定」のセルがピンクになっていることを確認しておきましょう。

## 第7章 練習問題

- ① VBEに切り替える
  - ② 《ツール》をクリック
  - ③ 《参照設定》をクリック
  - ④ 《参照可能なライブラリファイル》の《Microsoft Scripting Runtime》がになっていることを確認
  - ⑤ 《OK》をクリック
  - ⑥ プロジェクトエクスプローラーの《Sheet4 (練習問題)》をダブルクリック
  - ⑦ 《オブジェクト》ボックスのをクリックし、一覧から《Worksheet》を選択
  - ⑧ 《プロシージャ》ボックスのをクリックし、一覧から《Change》を選択
- ※「Worksheet\_SelectionChange」イベントプロシージャは削除しておきましょう。
- ⑨ 次のように「Worksheet\_Change」イベントプロシージャの内容を入力

### ■「Worksheet\_Change」イベントプロシージャ

```
1. Private Sub Worksheet_Change(ByVal Target As Range)
2.     Dim MyFSO As New FileSystemObject
3.     Dim MyTXT As TextStream
4.     Dim Filename As String
5.     Dim Rireki(2) As String
6.     Filename = ThisWorkbook.Path & "¥変更履歴.csv"
7.     Set MyTXT = MyFSO.OpenTextFile(Filename, ForAppending, True)
8.     Rireki(0) = Now
9.     Rireki(1) = Target.Address(False, False)
10.    Rireki(2) = Target.Value
11.    MyTXT.WriteLine Text:=Join(Rireki, ",")
12.    MyTXT.Close
13.    Set MyFSO = Nothing
14.    Set MyTXT = Nothing
15. End Sub
```

### ■プロシージャの意味

1. 「Worksheet\_Change (Range型の引数Targetは変更したセル)」イベントプロシージャ開始
2. FileSystemObject型のオブジェクト変数MyFSOの宣言とインスタンスの生成
3. TextStream型のオブジェクト変数MyTXTを使用することを宣言
4. 文字列型の変数Filenameを使用することを宣言
5. 文字列型の配列変数Rirekiを3要素使用することを宣言
6. 変数Filenameに実行中のプロシージャが記述されたブックが保存されているフォルダーの絶対パスと「¥変更履歴.csv」を連結して代入
7. テキストファイルを開いてオブジェクト変数MyTXTに代入(変数Filenameのファイルを開く、追記モードで開く、ファイルがない場合作成する)
8. 配列変数Rireki(0)に現在の日時を代入
9. 配列変数Rireki(1)に変更されたセル番地を代入
10. 配列変数Rireki(2)に変更されたセルの値を代入
11. 配列変数Rirekiの各要素を区切り文字「,」で結合した文字列と改行を書き込む
12. テキストファイルを閉じる
13. オブジェクト変数MyFSOの初期化
14. オブジェクト変数MyTXTの初期化
15. プロシージャ終了

※コンパイルし、上書き保存しておきましょう。

- ⑩ Excellに切り替える
- ⑪ 任意のセルに値を入力

※テキストファイル「変更履歴.csv」をメモ帳で開いて、文字列が追加されたことを確認しましょう。確認後、ファイルは閉じておきましょう。

# 総合問題 標準解答

## 総合問題 1

①

- ① VBEに切り替える
- ② プロジェクトエクスプローラーの《Sheet2 (マスタ登録)》をダブルクリック
- ③ 《オブジェクト》ボックスの  をクリックし、一覧から《Worksheet》を選択
- ④ 《プロシージャ》ボックスの  をクリックし、一覧から《BeforeDoubleClick》を選択
- ⑤ 次のように「Worksheet\_BeforeDoubleClick」イベントプロシージャの内容を入力

### ■「Worksheet\_BeforeDoubleClick」イベントプロシージャ

```
1. Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)
2.     Dim Oldbunrui As String
3.     Dim Newbunrui As String
4.     If Application.Intersect(Target, Range("分類リスト")) Is Nothing Then Exit Sub
5.     Newbunrui = InputBox("新しい「分類」を入力してください。")
6.     If Newbunrui <> "" Then
7.         サブ_前処理
8.         Oldbunrui = Target.Value
9.         Target.Value = Newbunrui
10.        Columns("F").Replace _
11.            What:=Oldbunrui, Replacement:=Newbunrui, _
12.            LookAt:=xlWhole, MatchByte:=False
13.        With Worksheets("売上データ")
14.            .Unprotect Mypass
15.            .Columns("E").Replace _
16.                What:=Oldbunrui, Replacement:=Newbunrui, _
17.                LookAt:=xlWhole, MatchByte:=False
18.            .Protect Mypass
19.        End With
20.        サブ_後処理
21.    End If
22.    Cancel = True
23. End Sub
```

※10行目、11行目、15行目、16行目はコードが長いので、行継続文字「\_(半角スペース+半角アンダースコア)」を使って行を複数に分割しています。行継続文字を使わずに1行で記述してもかまいません。

## ■プロシージャの意味

1. 「Worksheet\_BeforeDoubleClick(Range型の引数Targetはダブルクリックしたセル、ブール型の引数Cancel)」イベントプロシージャ開始
2. 文字列型の変数Oldbunruiを使用することを宣言
3. 文字列型の変数Newbunruiを使用することを宣言
4. ダブルクリックしたセルとセル範囲【分類リスト】の共有セルがない場合はプロシージャを抜け出す
5. 変数NewbunruiにInputBoxに入力された値を代入
6. 変数Newbunruiが空文字(「」)でない場合は
7. サブルーチン「サブ\_前処理」を呼び出す
8. 変数Oldbunruiに引数Targetのセルの値を代入
9. 引数Targetのセルの値に変数Newbunruiを入力
10. F列に対して置換を実行
11. 検索する文字列は変数Oldbunrui、置換後の文字列は変数Newbunrui、
12. 完全に一致するセルを検索、半角・全角は区別しない
13. ワークシート「売上データ」の
14. 保護を解除(パスワードに定数Mypassを指定)
15. E列に対して置換を実行
16. 検索する文字列は変数Oldbunrui、置換後の文字列は変数Newbunrui、
17. 完全に一致するセルを検索、半角・全角は区別しない
18. 保護(パスワードに定数Mypassを設定)
19. Withステートメント終了
20. サブルーチン「サブ\_後処理」を呼び出す
21. Ifステートメント終了
22. 引数CancelにTrueを代入(編集モードをキャンセル)
23. プロシージャ終了

※コンパイルし、上書き保存しておきましょう。

## ②

- ① Excellに切り替える
- ② 分類マスタの「麺類」と入力されたセルをダブルクリック
- ③ 分類を入力するダイアログボックスに「インスタント類」と入力
- ④ 《OK》をクリック
- ⑤ 分類マスタ、商品マスタ、売上データの「分類」の「麺類」が「インスタント類」に変更されていることを確認

## 総合問題 2

①

- ① VBEに切り替える
- ② 《ツール》をクリック
- ③ 《参照設定》をクリック
- ④ 《参照可能なライブラリファイル》の《Microsoft Scripting Runtime》を  にする
- ⑤ 《OK》をクリック

②

- ① 《挿入》をクリック
- ② 《標準モジュール》をクリック
- ③ モジュール「Module1」に次のように「売上データインポート」プロシージャを入力

### ■「売上データインポート」プロシージャ

```
1. Sub 売上データインポート()  
2.     Dim MyFSO As New FileSystemObject  
3.     Dim MyTXT As TextStream  
4.     Dim Myrange As Range  
5.     Dim Filename As String  
6.     Dim Uriage As Variant  
7.     If MsgBox("売上データをインポートします。", vbOKCancel) = vbCancel Then Exit Sub  
8.     Filename = ThisWorkbook.Path & "¥売上データ.csv"  
9.     Set MyTXT = MyFSO.OpenTextFile(Filename, ForReading)  
10.    ActiveSheet.Unprotect Mypass  
11.    Set Myrange = Cells(Rows.Count, 2).End(xlUp).Offset(1)  
12.    Do Until MyTXT.AtEndOfStream = True  
13.        With Myrange  
14.            .Value = Range("売上No.セル").Value  
15.            Uriage = Split(MyTXT.ReadLine, ",")  
16.            .Offset(, 1).Value = Uriage(0)  
17.            .Offset(, 2).Value = Uriage(1)  
18.            .Offset(, 3).Value = Uriage(2)  
19.            .Offset(, 4).Value = Uriage(3)  
20.            .Offset(, 5).Value = Uriage(4)  
21.            .Offset(, 6).Value = Uriage(5)  
22.            .Offset(, 7).Value = Uriage(6)  
23.            Range("売上No.セル").Value = Range("売上No.セル").Value + 1  
24.            Set Myrange = .Offset(1)  
25.        End With  
26.    Loop  
27.    MyTXT.Close  
28.    Range("書式設定セル").Copy  
29.    Set Myrange = Range("B5").CurrentRegion.Offset(1)  
30.    Myrange.Resize(Myrange.Rows.Count - 1).PasteSpecial Paste:=xlPasteFormats  
31.    Application.CutCopyMode = False  
32.    Range("B1").Select  
33.    ActiveSheet.Protect Mypass  
34.    MsgBox "売上データをインポートしました。"  
35.    Set MyFSO = Nothing  
36.    Set MyTXT = Nothing  
37.    Set Myrange = Nothing  
38. End Sub
```

## ■プロシージャの意味

1. 「売上データインポート」プロシージャ開始
2. FileSystemObject型のオブジェクト変数MyFSOを使用することを宣言してインスタンスを生成
3. TextStream型のオブジェクト変数MyTXTを使用することを宣言
4. Range型のオブジェクト変数Myrangeを使用することを宣言
5. 文字列型の変数Filenameを使用することを宣言
6. バリエーション型の変数Uriageを使用することを宣言
7. 《OK》《キャンセル》ボタンを持つメッセージボックスにメッセージ「売上データをインポートします。」を表示し、《キャンセル》がクリックされた場合はプロシージャを抜け出す
8. 変数Filenameに実行中のプロシージャが記述されたブックが保存されているフォルダーのパスと「¥売上データ.csv」を連結して代入
9. 変数Filenameのファイルを読み込みモードで開いてオブジェクト変数MyTXTに代入
10. パスワードに定数Mypassを指定してアクティブシートの保護を解除
11. オブジェクト変数Myrangeに、セル【B1】を起点としたワークシートの最終行セルの、上端セルの1行下のセルへの参照を代入
12. 読み込み位置がテキストファイルの末尾になるまで処理を繰り返す
13. オブジェクト変数Myrangeの
14. 値にセル【売上No.セル】の値を入力
15. 変数Uriageに1行分の文字列を区切り文字「,」で分割し、配列として代入
16. 1列右のセルに配列変数Uriage(0)の値を入力
17. 2列右のセルに配列変数Uriage(1)の値を入力
18. 3列右のセルに配列変数Uriage(2)の値を入力
19. 4列右のセルに配列変数Uriage(3)の値を入力
20. 5列右のセルに配列変数Uriage(4)の値を入力
21. 6列右のセルに配列変数Uriage(5)の値を入力
22. 7列右のセルに配列変数Uriage(6)の値を入力
23. セル【売上No.セル】の値にセル【売上No.セル】の値+1の結果を入力
24. オブジェクト変数Myrangeに1行下のセルへの参照を代入
25. Withステートメント終了
26. 12行目に戻る
27. テキストファイルを閉じる
28. セル範囲【書式設定セル】をコピー
29. セル【B5】を含む連続するセル範囲全体を1行下に移動させ、そのセル範囲への参照をオブジェクト変数Myrangeに代入(項目名の1行目をセル範囲から外すために、セル範囲全体を1行下にずらす)
30. オブジェクト変数Myrangeから下側の1行を縮めたセル範囲に書式だけを貼り付け(セル範囲全体を1行下にずらした分、セル範囲の下側を1行縮める)
31. コピーモードを解除
32. セル【B1】を選択
33. パスワードに定数Mypassを設定してアクティブシートを保護
34. メッセージ「売上データをインポートしました。」を表示
35. オブジェクト変数MyFSOを初期化
36. オブジェクト変数MyTXTを初期化
37. オブジェクト変数Myrangeを初期化
38. プロシージャ終了

※コンパイルし、上書き保存しておきましょう。

### ③

- ① Excellに切り替える
- ② 「売上データインポート」ボタンをクリック
- ③ 《OK》をクリック
- ④ 《OK》をクリック

## 4

- ① **2021/365** 《校閲》タブ→《保護》グループの  (シート保護の解除) をクリック
- 2019/2016** 《校閲》タブ→《変更》グループの  (シート保護の解除) をクリック
- ② 《パスワード》に「VBA」と入力  
※大文字で入力します。
- ③ 《OK》をクリック
- ④ 3～4行目を選択
- ⑤ 《ホーム》タブ→《セル》グループの  (書式) →《表示設定》の《非表示/再表示》→《行を表示しない》をクリック
- ⑥ **2021/365** 《校閲》タブ→《保護》グループの  (シートの保護) をクリック
- 2019/2016** 《校閲》タブ→《変更》グループの  (シートの保護) をクリック
- ⑦ 《シートの保護を解除するためのパスワード》に「VBA」と入力  
※大文字で入力します。
- ⑧ 《OK》をクリック
- ⑨ 《パスワードをもう一度入力してください。》に「VBA」と入力  
※大文字で入力します。
- ⑩ 《OK》をクリック

## 総合問題 3

①

- ① VBEに切り替える
- ② プロジェクトエクスプローラーの標準モジュール「Module1」をダブルクリック
- ③ 次のように「売上集計ブック出力」プロシージャを入力

### ■「売上集計ブック出力」プロシージャ

```
1. Sub 売上集計ブック出力()  
2.     Dim Mybook As Workbook  
3.     Dim Mysheet As Worksheet  
4.     Dim Myrange As Range  
5.     Dim Torihiki As String  
6.     Dim i As Integer  
7.     If MsgBox("取引先別集計をブックで出力します。", vbOKCancel) = vbCancel Then Exit Sub  
8.     Set Myrange = Range("取引先リスト")  
9.     Set Mysheet = Worksheets("売上データ")  
10.    Set Mybook = Workbooks.Add  
11.    Application.ScreenUpdating = False  
12.    Mysheet.Unprotect Mypass  
13.    For i = 1 To Myrange.Rows.Count  
14.        On Error GoTo ErrorSheet  
15.        Mybook.Worksheets(i).Select  
16.        On Error GoTo 0  
17.        Torihiki = Myrange.Cells(i, 1).Value  
18.        ActiveSheet.Name = i & "_" & Torihiki  
19.        With Mysheet  
20.            .Range("K6").Value = Torihiki  
21.            .Range("B5").CurrentRegion.AdvancedFilter _  
22.                Action:=xlFilterCopy, CriteriaRange:=.Range("K5:K6"), _  
23.                CopyToRange:=Range("A1")  
24.        End With  
25.        Columns("C").Delete  
26.        Columns("A:G").AutoFit  
27.        ActiveWindow.DisplayGridlines = False  
28.    Next i  
29.    Mybook.Worksheets(1).Select  
30.    Mysheet.Protect Mypass  
31.    Application.ScreenUpdating = True  
32.    On Error Resume Next  
33.    Workbooks("取引先別売上.xlsx").Close SaveChanges:=False  
34.    On Error GoTo 0  
35.    Application.DisplayAlerts = False  
36.    Mybook.SaveAs Filename:=ThisWorkbook.Path & "¥取引先別売上.xlsx"  
37.    Application.DisplayAlerts = True  
38.    Mybook.Close  
39.    Set Mybook = Nothing  
40.    Set Mysheet = Nothing  
41.    Set Myrange = Nothing  
42.    MsgBox "ブック形式での出力が終了しました."  
43.    Exit Sub  
44. ErrorSheet:  
45.     Mybook.Worksheets.Add After:=Mybook.Worksheets(Mybook.Worksheets.Count)  
46.     Resume  
47. End Sub
```

※21行目、22行目はコードが長いので、行継続文字「\_ (半角スペース+半角アンダースコア)」を使って行を複数に分割しています。行継続文字を使わずに1行で記述してもかまいません。

## ■プロシージャの意味

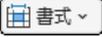
1. 「売上集計ブック出力」プロシージャ開始
2. ワークブック型のオブジェクト変数Mybookを使用することを宣言
3. ワークシート型のオブジェクト変数Mysheetを使用することを宣言
4. Range型のオブジェクト変数Myrangeを使用することを宣言
5. 文字列型の変数Torihikiを使用することを宣言
6. 整数型の変数iを使用することを宣言
7. 《OK》《キャンセル》ボタンを持つメッセージボックスにメッセージ「取引先別集計をブックで出力します。」を表示し、《キャンセル》がクリックされた場合はプロシージャを抜け出す
8. オブジェクト変数Myrangeにセル範囲【取引先リスト】への参照を代入
9. オブジェクト変数Mysheetにワークシート「売上データ」への参照を代入
10. オブジェクト変数Mybookに追加した新しいブックへの参照を代入
11. 画面の更新を無効に設定
12. パスワードに定数Mypassを指定してオブジェクト変数Mysheetの保護を解除
13. 変数iが1からオブジェクト変数Myrangeのセルの行数になるまで処理を繰り返す
14. エラーが発生した場合は行ラベルErrorSheetへ
15. オブジェクト変数Mybookのi番目のシートを選択
16. エラートラップを無効
17. 変数Torihikiにオブジェクト変数Myrangeのセル範囲のi行目1列目のセルの値を代入
18. アクティブシートの名前に変数i\_変数Torihikiを設定
19. オブジェクト変数Mysheetのワークシートの
20. セル【K6】に変数Torihikiの値を入力
21. セル【B5】を含む連続するセル範囲でフィルターオプションを実行(  
抽出結果は別の場所に抽出、検索条件はセル範囲【K5:K6】、  
抽出場所はアクティブシートのセル【A1】)
22. Withステートメント終了
23. C列を削除
24. A～G列の列幅を自動調整
25. アクティブウィンドウの枠線を非表示
26. 変数iに1を加算し、13行目に戻る
27. オブジェクト変数Mybookの1番目のワークシートを選択
28. パスワードに定数Mypassを設定してオブジェクト変数Mysheetを保護
29. 画面の更新を有効に設定
30. 発生したエラーは無視する
31. ブック「取引先別売上.xlsx」を保存せずに閉じる
32. エラートラップを無効
33. 警告のメッセージを無効
34. オブジェクト変数Mybookを実行中のプロシージャが記述されたブックが保存されているフォルダー内に「取引先別売上.xlsx」という名前で保存
35. 警告のメッセージを有効
36. オブジェクト変数Mybookのブックを閉じる
37. オブジェクト変数Mybookを初期化
38. オブジェクト変数Mysheetを初期化
39. オブジェクト変数Myrangeを初期化
40. メッセージ「ブック形式での出力が終了しました。」を表示
41. プロシージャを抜け出す
42. 行ラベルErrorSheet
43. オブジェクト変数Mybookのワークシートの最後にワークシートを追加
44. エラーが発生した行へ戻る
45. プロシージャ終了

※コンパイルし、上書き保存しておきましょう。

## ②

- ① Excelに切り替える
- ② 「売上集計ブック出力」ボタンをクリック
- ③ 《OK》をクリック
- ④ 《OK》をクリック
- ⑤ フォルダー「ExcelVBAプログラミング実践2021/2019/2016/365」のフォルダー「総合問題」に「取引先別売上.xlsx」が出力されていることを確認  
※ブック「取引先別売上.xlsx」を閉じておきましょう。

## ③

- ① **2021/365** 《校閲》タブ→《保護》グループの  (シート保護の解除) をクリック  
**2019/2016** 《校閲》タブ→《変更》グループの  (シート保護の解除) をクリック
- ② 《パスワード》に「VBA」と入力  
※大文字で入力します。
- ③ 《OK》をクリック
- ④ K列を選択
- ⑤ 《ホーム》タブ→《セル》グループの  (書式) →《表示設定》の《非表示/再表示》→《列を表示しない》をクリック
- ⑥ **2021/365** 《校閲》タブ→《保護》グループの  (シートの保護) をクリック  
**2019/2016** 《校閲》タブ→《変更》グループの  (シートの保護) をクリック
- ⑦ 《シートの保護を解除するためのパスワード》に「VBA」と入力  
※大文字で入力します。
- ⑧ 《OK》をクリック
- ⑨ 《パスワードをもう一度入力してください。》に「VBA」と入力  
※大文字で入力します。
- ⑩ 《OK》をクリック

## 総合問題 4

①

- ① VBEに切り替える
- ② プロジェクトエクスプローラーの標準モジュール「Module1」をダブルクリック
- ③ 次のように「売上集計印刷出力」プロシージャを入力

### ■「売上集計印刷出力」プロシージャ

```
1. Sub 売上集計印刷出力()  
2.     Dim Myrange As Range  
3.     Dim Bunrui As Range  
4.     If MsgBox("分類別集計を印刷で出力します。", vbOKCancel) = vbCancel Then Exit Sub  
5.     ActiveSheet.Unprotect Mypass  
6.     Set Myrange = Range("B5").CurrentRegion  
7.     Set Bunrui = Range("E6")  
8.     Myrange.Sort Key1:=Range("E5"), Order1:=xlAscending, Header:=xlYes  
9.     ActiveSheet.ResetAllPageBreaks  
10.    Do Until Bunrui.Offset(1).Value = ""  
11.        If Bunrui.Value <> Bunrui.Offset(1).Value Then  
12.            ActiveSheet.HPageBreaks.Add Before:=Bunrui.Offset(1)  
13.        End If  
14.        Set Bunrui = Bunrui.Offset(1)  
15.    Loop  
16.    With ActiveSheet.PageSetup  
17.        .PrintArea = Myrange.Address  
18.        .PrintTitleRows = Rows(5).Address  
19.    End With  
20.    On Error Resume Next  
21.    ActiveSheet.PrintOut Preview:=True  
22.    On Error GoTo 0  
23.    Myrange.Sort Key1:=Range("B5"), Order1:=xlAscending, Header:=xlYes  
24.    ActiveSheet.Protect Mypass  
25.    Set Myrange = Nothing  
26.    Set Bunrui = Nothing  
27. End Sub
```

## ■プロシージャの意味

1. 「売上集計印刷出力」プロシージャ開始
2. Range型のオブジェクト変数Myrangeを使用することを宣言
3. Range型のオブジェクト変数Bunruiを使用することを宣言
4. 《OK》《キャンセル》ボタンを持つメッセージボックスにメッセージ「分類別集計を印刷で出力します。」を表示し、《キャンセル》がクリックされた場合はプロシージャを抜け出す
5. パスワードに定数Mypassを指定してアクティブシートの保護を解除
6. オブジェクト変数Myrangeにセル【B5】を含む連続するセル範囲への参照を代入
7. オブジェクト変数Bunruiにセル【E6】への参照を代入
8. オブジェクト変数Myrangeのセル範囲で並べ替えを実行(並べ替えフィールドはセル【E5】、昇順、先頭行を見出しとする)
9. アクティブシートのすべての改ページを解除
10. オブジェクト変数Bunruiの1行下のセルの値が空文字(「」)になるまで処理を繰り返す
11. オブジェクト変数Bunruiの値とオブジェクト変数Bunruiの1行下のセルの値が異なる場合は
12. オブジェクト変数Bunruiの1行下のセルの上に水平改ページを追加
13. Ifステートメント終了
14. オブジェクト変数Bunruiに1行下のセルへの参照を代入
15. 10行目に戻る
16. アクティブシートのページ設定の
17. 印刷範囲にオブジェクト変数Myrangeのセル範囲のセル番地を設定
18. 行タイトルに5行目を設定
19. Withステートメント終了
20. 発生したエラーは無視する
21. 印刷プレビューを表示してアクティブシートを印刷
22. エラートラップを無効
23. オブジェクト変数Myrangeのセル範囲で並べ替えを実行(並べ替えフィールドはセル【B5】、昇順、先頭行を見出しとする)
24. パスワードに定数Mypassを設定してアクティブシートを保護
25. オブジェクト変数Myrangeを初期化
26. オブジェクト変数Bunruiを初期化
27. プロシージャ終了

※コンパイルし、上書き保存しておきましょう。

## ②

- ① Excellに切り替える
- ② 「売上集計印刷出力」ボタンをクリック
- ③ 《OK》をクリック
- ④ 《印刷プレビュー》タブ→《印刷》グループの  (印刷) をクリック

よくわかる

# Microsoft® Excel® VBAプログラミング実践

Office 2021/2019/2016/Microsoft 365 対応

(FPT2308)

## 練習問題・総合問題 標準解答

2023年12月 5 日 初版発行

著作／制作：株式会社富士通ラーニングメディア

- 
- Microsoft、Excel、Microsoft 365は、マイクロソフトグループの企業の商標です。
  - その他、記載されている会社および製品などの名称は、各社の登録商標または商標です。
  - 本文中では、TMや®は省略しています。
  - 本文中のスクリーンショットは、マイクロソフトの許諾を得て使用しています。
  - 本資料は、構成・文章・プログラム・画像・データなどのすべてにおいて、著作権法上の保護を受けています。  
本資料の一部あるいは全部について、いかなる方法においても複写・複製など、著作権法上で規定された権利を侵害する行為を行うことは禁じられています。
  - 本製品に起因してご使用者に直接または間接的損害が生じても、株式会社富士通ラーニングメディアはいかなる責任も負わないものとし、一切の賠償などは行わないものとします。
  - 本資料に記載された内容などは、予告なく変更される場合があります。
  - 購入者自らが使用になる場合に限り、複製を許諾します。