

VBAの 実用サンプル

書籍の p.407 ～ p.422 にて紹介した実用サンプルについて、紙面に入りきらなかった詳細な解説を記載します。

01

データの入力フォーム

データ入力用フォームの概要

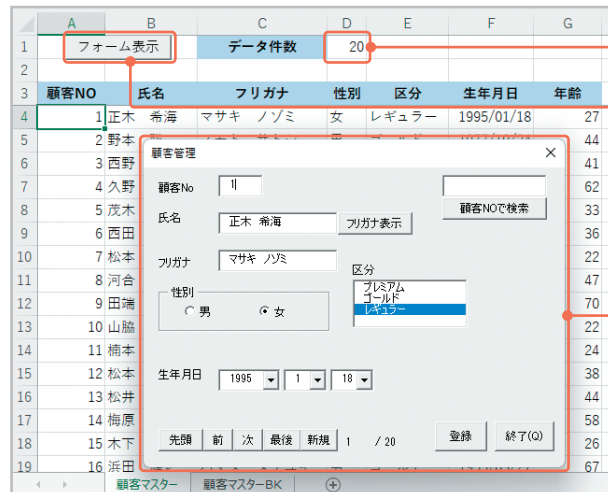
ここでは、ワークシート上のレコードをユーザーフォームに読み込んで、ユーザーフォーム上で表示、編集、追加するサンプルを紹介します。

対象となるワークシートのセル D1 には、「=COUNTA(A4:A1048576)」のように COUNTA 関数を設定してデータ件数を求めています。また、データ件数を表示するセル D1 に名前「件数」を設定しています。

Memo

COUNTA 関数は引数に指定したセル範囲のデータの個数を求めます。「=COUNTA(A4:A1048576)」は、セル A4 から一番下までのデータの個数を求めます。

作成するユーザーフォーム



名前 [件数]

① [顧客マスター] シートの [フォーム表示] ボタンをクリックする。

② [顧客管理] フォームを表示する。画面上でデータの表示 / 修正 / 追加 / 検索などの操作ができる。

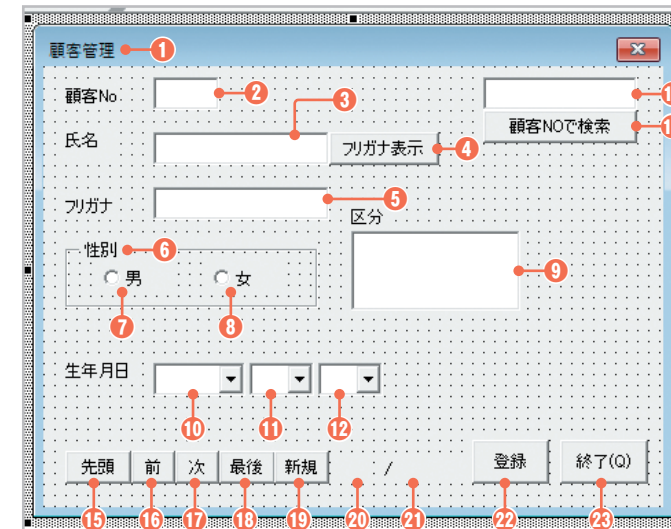
①でボタンに割り当てられている [フォーム表示] プロシージャ（作成場所：標準モジュール）が実行され、②のフォームが開きます。

Sample 【フォーム表示】 ボタンがクリックされた際の処理

```
Sub フォーム表示 ()
    UserForm1.Show vbModeless 'ユーザーフォームを開く
End Sub
```

ユーザーフォームの作成

次のようなユーザーフォームを作成します（作成手順は p.364 を参照してください）。なお、文字を表示するだけのラベルの設定は省略しています。



プロパティウィンドウでの設定内容

No.	ユーザーフォーム / コントロール名	プロパティ	設定値
①	ユーザーフォーム	オブジェクト名	UserForm1
		Caption	顧客管理
②	テキストボックス	オブジェクト名	txtNo
③	テキストボックス	オブジェクト名	txtName
④	コマンドボタン	オブジェクト名	btnKana
		Caption	フリガナ表示

No.	ユーザーフォーム / コントロール名	プロパティ	設定値
5	テキストボックス	オブジェクト名	txtKana
6	フレーム	Caption	性別
7	オプションボタン	オブジェクト名	optMan
		Caption	男
8	オプションボタン	オブジェクト名	optWoman
		Caption	女
9	リストボックス	オブジェクト名	lstKubun
10	コンボボックス	オブジェクト名	cmbYear
11	コンボボックス	オブジェクト名	cmbMonth
12	コンボボックス	オブジェクト名	cmbDay
13	テキストボックス	オブジェクト名	txtKensaku
14	コマンドボタン	オブジェクト名	btnKensaku
		Caption	顧客 No で検索
15	コマンドボタン	オブジェクト名	btnTop
		Caption	先頭
16	コマンドボタン	オブジェクト名	btnPre
		Caption	前
17	コマンドボタン	オブジェクト名	btnNext
		Caption	次
18	コマンドボタン	オブジェクト名	btnLast
		Caption	最後
19	コマンドボタン	オブジェクト名	btnNew
		Caption	新規
20	ラベル	オブジェクト名	lblRec
21	ラベル	オブジェクト名	lblAll
22	コマンドボタン	オブジェクト名	btnSave
		Caption	登録
23	コマンドボタン	オブジェクト名	btnQuit
		Caption	終了 (Q)
		Accelerator	Q

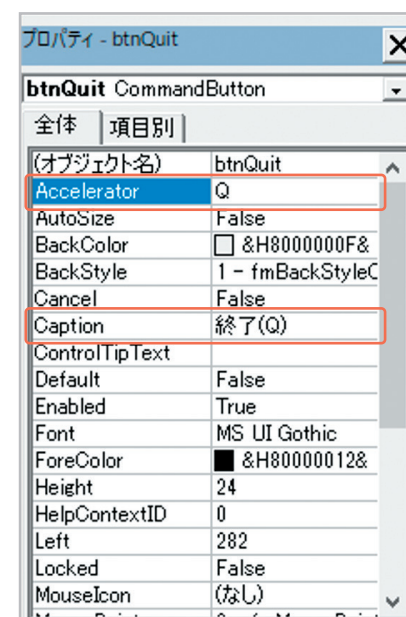
※ここでは男・女のオプションボタンをフレーム内に配置しています。先にフレームを追加した後、その中にオプションボタン（7、8）を追加します。

Column

コマンドボタンにアクセスキーを設定する

アクセスキーとは、コマンドボタンなどのコントロールを選択するための1文字で、**[Alt]**キーを押しながらアクセスキーを押すと、そのコントロールが選択されたり、コマンドを実行したりできます。コマンドボタンであれば、クリックと同じ操作になります。

アクセスキーを設定するには、[プロパティウィンドウ]の[Accelerator]プロパティでアクセスキーに割り当てたいアルファベットを半角文字で入力し、[Caption]プロパティで[終了 (Q)]のように割り当てたアクセスキーを入力します。



開く / 閉じるときのプロシージャを作成

フォームが開く直前に実行するユーザーフォームのInitializeイベントプロシージャで初期設定をします。

ここでは、リストボックスとコンボボックスの選択肢となる値を追加します。[終了]ボタン(btnQuit)をクリックしたときに実行するClickイベントプロシージャでフォームを閉じる処理を行います。

Sample Initialize イベントプロシージャ

```
Private Sub UserForm_Initialize()
    Dim i As Long
    'リストボックスに選択肢を追加
    lstKubun.AddItem "プレミアム"
    lstKubun.AddItem "ゴールド"
    lstKubun.AddItem "レギュラー"
    'コンボボックスに年の選択肢を追加
    For i = 1930 To 2022
        cmbYear.AddItem i
    Next
    'コンボボックスに月の選択肢を追加
    For i = 1 To 12
        cmbMonth.AddItem i
    Next
    'コンボボックスに日の選択肢を追加
    For i = 1 To 31
        cmbDay.AddItem i
    Next
End Sub
```

※データが1件もない場合の処理を Appendix の p.16 で追加しています。

Sample [終了] ボタン (btnQuit) がクリックされた際の処理

```
Private Sub btnQuit_Click()
    Unload Me 'ユーザーフォームを閉じる
End Sub
```

なお、この章で紹介するプロシージャ内には、エラー処理コードは記述していません。本編 p.344 を参考に、必要に応じてエラー処理コードを追加してください。

● ユーザーフォーム

リストボックス、コンボボックスに選択肢が追加されている

タブオーダーの設定

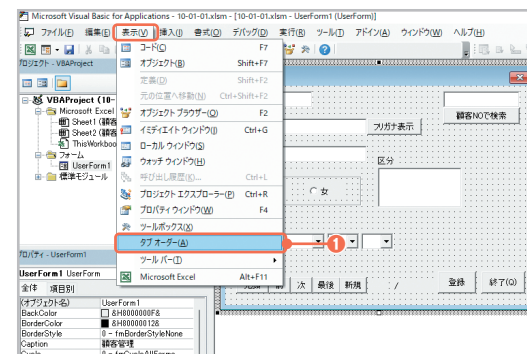
ユーザーフォームを開いて、**[Tab]**キーを押すと、**コントロールを配置した順番にフォーカスが移動**します。この移動順のことを**タブオーダー**といいます。

フォーカスとは、文字の入力などキーボード操作を受け取ることができる状態で、テキストボックスではカーソルが表示され、オプションボタンやコマンドボタンでは表示されている文字列が点線で囲まれ、操作対象となっている状態です。タブオーダーをデータの入力順に設定しておけば入力がスムーズです。

タブオーダーを設定する手順は次の通りです。

Step 1

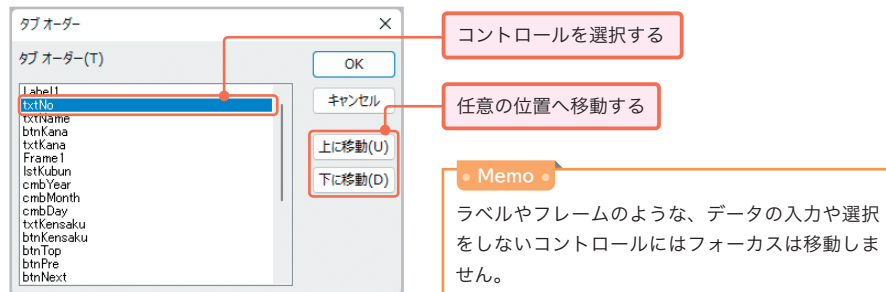
ユーザーフォームを表示し、[表示] メニューの [タブオーダー] をクリック①



● Memo ●
ユーザーフォームが表示されていない場合は、[プロジェクトエクスプローラー] で開きたいユーザーフォーム名をダブルクリックします。

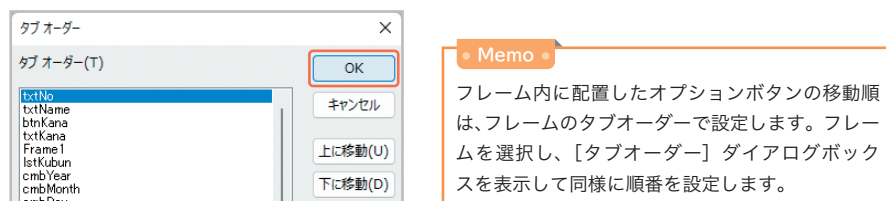
Step 2

[タブオーダー] ダイアログで順番を移動したいコントロールを選択し、[上/下に移動] ボタンをクリックして、コントロールを任意の位置まで移動



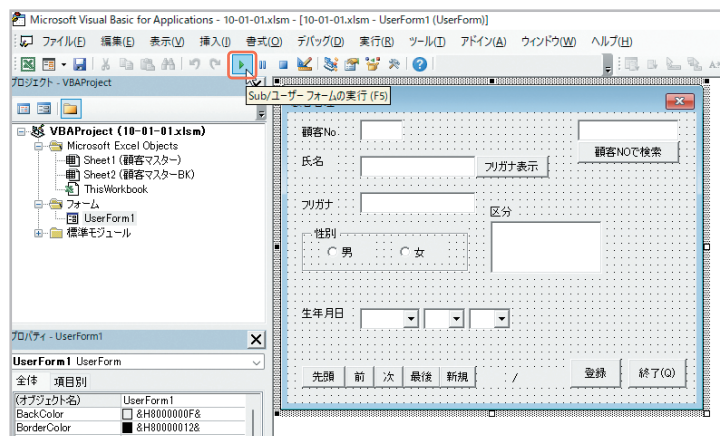
Step 3

目的の順番に変更できたら [OK] ボタンで確定



Step 4

[Sub/ユーザーフォームの実行] ボタンでユーザーフォームを開く



Step 5

[Tab] キーを押して、タブオーダーで設定した順番にフォーカスが移動することを確認する。



• Memo •

逆方向に移動するには、[Shift] キーを押しながら [Tab] キーを押します。

GetPhonetic メソッドでフリガナを取得する

漢字の氏名を入力したら、そのフリガナを自動で表示できるようにすると入力の手間が省けます。VBA で入力された文字のフリガナを取得するには Application オブジェクトの **GetPhonetic メソッド** を使います。

書式 >> GetPhonetic

Application.GetPhonetic ([Text])

引数 Text : フリガナに変換する文字列を指定。
省略時は以前に指定した引数 [Text] のフリガナ候補の文字列を返す。
候補となる文字列がない場合は空の文字列 [""] を返す。

以下のコードは [フリガナ表示] ボタンをクリックしたときに実行する Click イベントプロシージャで、テキストボックス txtName に入力された文字列のフリガナをテキストボックス txtKana に表示するコードは以下のようになります。

Sample [フリガナ表示] ボタン (btnKana) がクリックされた際の処理

```
Private Sub btnKana_Click()  
    txt3Kana.Value = Application.GetPhonetic(txtName.Value)  
End Sub
```

- ① 氏名を入力して [フリガナ表示] ボタンをクリックする。
- ② [フリガナ] 欄にフリガナが表示される。

● ユーザーフォームの実行結果

- 1 氏名を入力し、[フリガナ表示] ボタンをクリックすると、
- 2 [フリガナ] 欄にフリガナが表示される。

● Memo ●

正しいフリガナが表示されなかった場合は、[フリガナ] 欄で直接修正してください。

Column

フリガナをひらがなで表示する

GetPhonetic メソッドで表示されるフリガナをひらがなにしたい場合は、文字種を変換する StrConv 関数を使用して以下のように記述します。

Sample

```
txtKana.Value = StrConv(Application.GetPhonetic(txtName.Value), vbHiragana)
```

Column

日本語入力モードを設定する

テキストボックスに入力するデータの種別にあわせて日本語入力モードを設定できます。テキストボックスの [プロパティウィンドウ] の [IMEMode] プロパティで文字種を一覧から選択します。テキストボックスだけでなくコンボボックスでも設定できます。

● IMEMode プロパティの主な設定値

設定値	内容
mIMEModeNoControl	変更なし (既定値)
fmIMEModeOn	IME オン
fmIMEModeOff	IME オフ (英語モード)
fmIMEModeDisable	IME オフ (使用不可)
fmIMEModeHiragana	全角ひらがな
fmIMEModeKatakana	全角カタカナ
fmIMEModeKatakanaHalf	半角カタカナ
fmIMEModeAlphaFull	全角英数字
fmIMEModeAlpha	半角英数字

ユーザーフォームのコントロールにセルの値を表示する

テキストボックスなどユーザーフォーム上のコントロールにセルの値を表示するには、コントロールの Value プロパティか Text プロパティにセルの Value プロパティの値を代入します。オプションボタンの場合は、True / False の値とセルの値を対応させるための処理が必要になります。

● 顧客マスター表のデータとユーザーフォームの対応

顧客マスターの表のデータをユーザーフォームに表示する

セルの値を読み込むプロシージャを作成する

フォームを開いたときは1レコード目、[次] ボタンをクリックしたら次のレコード……というように、場合によって表示するレコードは異なります。どのレコードを表示するかは、[顧客 NO] 列のように固有の値をもつ列の値を引数に使って、レコードを読み込む Sub プロシージャを作成し、それを呼び出して使用します。ここでは、[顧客 NO] の値を代入する引数 cNo を受け取って、レコードをフォームに読み込む Sub プロシージャ [レコード読み込み] を作成します。このプロシージャは UserForm1 のコードウィンドウに記述します。

Sample [UserForm1] ユーザーフォームのコードウィンドウに記述する処理

```

'A: [レコード読み込み] プロシージャ、変数宣言と値の代入
Dim rRng As Range
Const S_ROW As Long = 3 '表の見出し行の行番号

Sub レコード読み込み (cNo As Long) 'cNo...顧客 NOの値
    Dim r As Long '顧客 Noのセルの行番号代入用
    Set rRng = Range("A3").CurrentRegion.Columns(1).Find( _
        What:=cNo, Lookat:=xlWhole)
    If rRng Is Nothing Then
        MsgBox "データが見つかりません"
        Exit Sub
    End
    r = rRng.Row '見つかったセルの行番号を取得
'B: テキストボックスに値を表示
    txtNo.Text = Cells(r, 1).Value '顧客 No
    txtName.Text = Cells(r, 2).Value '氏名
    txtKana.Text = Cells(r, 3).Value 'フリガナ
'C: オプションボタンに値を表示(性別)
    Select Case Cells(r, 4).Value
        Case "男"
            optMan.Value = True
        Case "女"
            optWoman.Value = True
        Case Else
            optMan.Value = False
            optWoman.Value = False
    End Select
'D: リストボックスに値を表示(区分)
    If Cells(r, 5).Value <> "" Then
        lstKubun.Text = Cells(r, 5).Value
    Else
        lstKubun.ListIndex = -1
    End If
'E: コンボボックスに値を表示(生年月日)
    If Cells(r, 6).Value <> "" Then
        cmbYear.Text = Year(Cells(r, 6).Value)
        cmbMonth.Text = Month(Cells(r, 6).Value)
        cmbDay.Text = Day(Cells(r, 6).Value)
    Else
        cmbYear.Text = ""
        cmbMonth.Text = ""
        cmbDay.Text = ""
    End If
End Sub

```

```

'F: ラベルに現在のレコード番号とレコード数を表示
lblRec.Caption = r - S_ROW
lblAll.Caption = "/" & Range("件数").Value
End Sub

```

☑ブロック A

「レコード読み込み」プロシージャ、変数宣言と値の代入を行います。

```

'A: [レコード読み込み] プロシージャ、変数宣言と値の代入
Dim rRng As Range
Const S_ROW As Long = 3 '表の見出し行の行番号

Sub レコード読み込み (cNo As Long) 'cNo...顧客 NOの値
    Dim r As Long '顧客 Noのセルの行番号代入用
    Set rRng = Range("A3").CurrentRegion.Columns(1).Find( _
        What:=cNo, Lookat:=xlWhole)
    If rRng Is Nothing Then
        MsgBox "データが見つかりません"
        Exit Sub
    End
    r = rRng.Row '見つかったセルの行番号を取得

```

- ① 表示する「顧客 NO」のセルを代入するために、Range 型の変数 **rRng** をモジュールレベルで宣言する。
- ② 表の見出し行の行番号 3 を代入した定数 **S_ROW** を宣言する。
- ③ 引数 **cNo**（顧客 No の値）を受け取って実行する「レコード読み込み」というプロシージャを作成する。
- ④ セル **A3** を含む表の 1 列目（「顧客 NO」の列）から引数 **cNo** の値を持つセルを検索し、見つかったら変数 **rRng** に代入する。
- ⑤ 見つからなかった場合は、「データが見つかりません」とメッセージ表示して終了する。
- ⑥ 変数 **rRng**（顧客 No のセル）の行番号を変数 **r** に代入する。

☑ブロック B

テキストボックスに値を表示します。

```

'B: テキストボックスに値を表示
txtNo.Text = Cells(r, 1).Value '顧客 No
txtName.Text = Cells(r, 2).Value '氏名
txtKana.Text = Cells(r, 3).Value 'フリガナ

```

- ⑦ テキストボックスにそれぞれ、
 - txtNo** → **r** 行 1 列目（顧客 No）の値を表示する。
 - txtName** → **r** 行 2 列目（氏名）の値を表示する。
 - txtKana** → **r** 行 3 列目（フリガナ）の値を表示する。

✓ ブロック C

オプションボタンの選択値を設定します。

'C: オプションボタンに値を表示(性別)

```
Select Case Cells(r, 4).Value
Case "男"
    optMan.Value = True
Case "女"
    optWoman.Value = True
Case Else
    optMan.Value = False
    optWoman.Value = False
End Select
```

- ⑧ r 行 4 列目のセルの値（性別）が、
 男の場合 → オプションボタン **optMan** をオンにする。
 女の場合 → オプションボタン **optWoman** をオンにする。
 どちらもでない場合 → **optMan**、**optWoman** どちらもオフにする。

✓ ブロック D

リストボックスの選択値を設定します。

'D: リストボックスに値を表示(区分)

```
If Cells(r, 5).Value <> "" Then
    lstKubun.Text = Cells(r, 5).Value
Else
    lstKubun.ListIndex = -1
End If
```

- ⑨ r 行 5 列目のセルの値（区分）が
 空欄でない場合 → r 行 5 列目の値をリストボックス **lstKubun** に表示する。
 空欄の場合 → **lstKubun** は未選択の状態にする。

• Memo •

セルの値をリストボックスの Text プロパティに代入することで、リストボックスの項目を選択できます。このときセルの値がリストボックスに追加されている必要があります。追加されていなかった場合はエラーになります。

✓ ブロック E

コンボボックスに値を表示します。

'E: コンボボックスに値を表示(生年月日)

```
If Cells(r, 6).Value <> "" Then
    cmbYear.Text = Year(Cells(r, 6).Value)
    cmbMonth.Text = Month(Cells(r, 6).Value)
    cmbDay.Text = Day(Cells(r, 6).Value)
Else
    cmbYear.Text = ""
    cmbMonth.Text = ""
    cmbDay.Text = ""
End If
```

- ⑩ r 行 6 列目のセルの値（生年月日）が
 空欄でない場合 → r 行 6 列目のセルの値から年 / 月 / 日を取得して
 各コンボボックスに表示する。
 空欄の場合 → それぞれのコンボボックスを空欄にする。

• Memo •

コンボボックスを空欄にするのに、Text プロパティに「」を代入する代わりに、「cmbYear.ListIndex=-1」と記述することもできます。

✓ ブロック F

フォームに表示するデータのレコード番号は表のデータの行番号から見出し行の行番号 (**S_ROW**) を引いた数となります。これを使ってラベルに現在のレコード番号とレコード数を表示します。

'F: ラベルに現在のレコード番号とレコード数を表示

```
lblRec.Caption = r - S_ROW
lblAll.Caption = "/" & Range("件数").Value
```

- ⑪ ラベルに現在のレコード番号、レコード数を表示
 ラベル **lblRec** (レコード番号)
 → 変数 **rRng** の行番号 **r** から定数 **S_ROW(3)** を引いた値を表示する。
 ラベル **lblAll** (レコード数)
 → 「/」と、名前「件数」のセルの値を表示する。
 レコード番号は、表示データの行番号から定数 **S_ROW** を引いた値になる。

ListIndex プロパティ

ListIndex プロパティは、リストボックスやコンボボックスで選択されている項目を上から 0、1、2…の数値で返します。何も選択されていない場合は ListIndex は -1 を返します。

ここでは、[区分] 列のセルが空欄だった場合は、「リストボックスで何も選択しない」という意味で ListIndex に -1 を代入しています。

フォームを開いたときに 1 レコード目を表示する

フォームを開いたときに表の 1 レコード目（1 件目）を表示するには、フォームの Initialize イベントプロシージャの最後に次のコードを追加します。表の 1 件目の顧客 No の値を引数に設定した、[レコード読み込み] プロシージャを呼び出します。

なお、データが 1 件も入力されていない場合は [顧客 No] に 1 を表示します。

Sample フォームを開くときに実行する Initialize イベントプロシージャ

```
Private Sub UserForm_Initialize()
    Dim i As Long
    : (省略)
    If Range("A4").Value = "" Then ①
        txtNo.Text = 1
    Else ②
        Call レコード読み込み (Range("A4").Value)
    End If
End Sub
```

セル A4（1 レコード目のセル）に

何も入力されていない場合

→テキストボックス「txtNo」（顧客 No）に 1 を表示し、変数 rRng にセル A4 を代入する。

データが入力されていた場合

→ A4 の値を引数にして [レコード読み込み] プロシージャを呼び出す。

lblRec.Caption: レコード番号: セル A4 の行番号から定数 S_ROW (3) を引いた値になる

lblAll.Caption: 全レコード数: セル D1 (名前 [件数]) の値

[次] [前] ボタンでデータを切り替える

[次] ボタンでは、現在表示しているレコードの 1 行下のレコードを表示し、[前] ボタンでは、1 行上のレコードを表示します。現在表示しているレコードが先頭の場合 / 最後の場合で処理を分けた上で引数 cNo の値を取得し、[レコード読み込み] プロシージャを呼び出します。

Sample [次] ボタン (btnNext) がクリックされた際の処理

```
Private Sub btnNext_Click()
    If rRng.Row = Range("A" & Rows.Count).End(xlUp).Row Then ①
        Exit Sub
    Call レコード読み込み (rRng.Offset(1).Value) ②
End Sub
```

① 現在表示しているデータの行番号が表の最下行のセルの行番号と同じ場合（最後の場合）は、処理を終了する。

② 現在表示しているデータの 1 つ下の行のセルの値を引数にして [レコード読み込み] プロシージャを呼び出す。

●【先頭】 ボタンクリック時

【新規】 ボタンでコントロールを初期化する

【新規】 ボタンを押したら、新規入力用にフォームに表示されているデータをクリアし、テキストボックス txt1No に最終レコードの顧客 NO に 1 を加算した値を設定します。コントロールの値をリセットするために For Each ステートメントを使い、TypeName 関数でコントロールの種類を調べ、コントロールの種類によってリセットの方法を指定しています。

Sample 【新規】 ボタン (btnNew) がクリックされた際の処理

```
Private Sub btnNew_Click()
    Dim lstRng As Range '表の最終レコードのセル代入用
    Dim obj As Control

    'A: コントロールの値をリセット
    For Each obj In UserForm1.Controls
        Select Case TypeName(obj)
            Case "TextBox"
                obj.Value = ""
            Case "ListBox", "ComboBox"
                obj.ListIndex = -1
            Case "OptionButton"
                obj.Value = False
        End Select
    Next
```

'B: 最終セルと新規データ用セルの取得

```
Set lstRng = Range("A" & Rows.Count).End(xlUp)
Set rRng = lstRng.Offset(1)
'ラベルに[新規レコード] と表示
lblRec.Caption = ""
lblAll.Caption = "新規レコード"
txtNo.Value = lstRng.Value + 1
End Sub
```

☑ ブロック A

コントロールの値を初期化します。

'A: コントロールの値をリセット

```
For Each obj In UserForm1.Controls
    Select Case TypeName(obj)
        Case "TextBox"
            obj.Value = ""
        Case "ListBox", "ComboBox"
            obj.ListIndex = -1
        Case "OptionButton"
            obj.Value = False
    End Select
Next
```

- ① オブジェクト型の変数 obj を UserForm1 上のすべてのコントロールに順番に代入しながら以下の処理を実行する。
- ② 変数 obj に代入されたコントロールの種類が
 テキストボックス → Value プロパティに "" を代入する。
 リストボックスまたはコンボボックス → ListIndex プロパティに -1 を代入する。
 オプションボタン → オフに設定する (False)。

☑ ブロック B

最終セルと新規データ用のセルを取得します。

'B: 最終セルと新規データ用セルの取得

```
Set lstRng = Range("A" & Rows.Count).End(xlUp)
Set rRng = lstRng.Offset(1)
'ラベルに[新規レコード] と表示
lblRec.Caption = ""
lblAll.Caption = "新規レコード"
txtNo.Value = lstRng.Value + 1
```



```
Cells(r, 6).NumberFormatLocal = "yyyy/mm/dd" ①⑦
Cells(r, 7).Formula = "=IF(" & Cells(r, 6).Address & _
    "=""", DATEDIF(" & Cells(r, 6).Address & _
    ",TODAY(),""Y""))" ①⑧
'ラベルに現在のレコード番号とレコード数を表示
lblRec.Caption = rRng.Row - S_ROW ①⑨
lblAll.Caption = "/" & Range("件数").Value ①⑩
End Sub
```

このコードは大きく3つのブロックに分かれています。

A: 入力欄の空欄チェック、データ型チェック

B: 入力内容の重複チェック

C: レコード書き込み

以下、ブロックごとにコードの解説を行います。

☑ ブロック A

入力欄が空欄かどうか、生年月日のコンボボックスに入力された値が日付として扱えるかどうかをチェックしています。

```
'A: データチェック
birth = cmbYear.Value & "/" & cmbMonth.Value & "/" & cmbDay.Value ①
Select Case True ②
    Case txtNo.Value = "" ③
        MsgBox "顧客 NOを入力してください" ③
        Exit Sub
    Case txtName.Value = "" ④
        MsgBox "氏名を入力してください" ④
        Exit Sub
    Case optMan.Value = False And optWoman.Value = False ⑤
        MsgBox "性別を選択してください" ⑤
        Exit Sub
    Case lstKubun.ListIndex = -1 ⑥
        MsgBox "区分を選択してください" ⑥
        Exit Sub
    Case IsDate(birth) = False ⑦
        MsgBox "正しい日付を入力してください" ⑦
        Exit Sub
End Select
```

- ① コンボボックス **cmbYear**、**cmbMonth**、**cmbDay** の値を「/」でつなげて「2022/11/26」の形式で文字列にして変数 **birth** に代入する。
- ② 以下の値が **True** かどうかを判定する。
- ③ テキストボックス **txtNo** の値が「」の場合、「顧客 NOを入力してください」とメッセージを表示して処理を終了する。
- ④ テキストボックス **txtName** の値が「」の場合、「氏名を入力してください」とメッセージを表示して処理を終了する。
- ⑤ オプションボタン **optMan** と **optWoman** の値がともに **False** の場合、「性別を選択してください」とメッセージを表示して処理を終了する。
- ⑥ リストボックス **lstKubun** が何も選択されていない場合、「区分を選択してください」とメッセージを表示して処理を終了する。
- ⑦ 変数 **birth** が日付とみなされない場合、「正しい日付を入力してください」とメッセージを表示して処理を終了する。

◆ Memo ◆

IsDate 関数は、引数に指定した値が日付や時刻として扱えるかどうかを調べ、扱える場合は True、扱えない場合は False が返る (p.356 参照)。

☑ ブロック B

テキストボックス **txtNo** の値が既に入力済みのレコードと重複するかどうかをチェックしています。

```
'B: 重複チェック
Set rng = Range("A4").CurrentRegion.Columns(1).Find( _ ⑧
    What:=txtNo.Value, lookat:=xlWhole) ⑧
'同じ顧客番号が見つかった場合
If Not rng Is Nothing Then
    rng.Select ⑨
    If MsgBox("上書きしますか?", vbYesNo) = vbNo Then Exit Sub ⑩
    Set rRng = rng ⑪
End If
```

- ⑧ セル **A4** を含む表の1列目でテキストボックス **txtNo** の値（顧客 No）を検索し、見つかったら変数 **rng** に代入する。
- ⑨ 見つかったセルを選択する。
- ⑩ メッセージを表示に対して **[No]** ボタンが選択された場合、処理を終了する。
- ⑪ 変数 **rRng** に変数 **rng** の値を代入する（見つかったセルが書き換え対象のセルとなる）。

▼ ブロック C

ユーザーフォームに表示されている値を「顧客マスター」表に書き出します。

'C: 書き込み

```

r = rRng.Row
Cells(r, 1).Value = txtNo.Value
Cells(r, 2).Value = txtName.Value
Cells(r, 3).Value = txtKana.Value
If optMan.Value Then
    Cells(r, 4).Value = "男"
Else
    Cells(r, 4).Value = "女"
End If
Cells(r, 5).Value = lstKubun.Value
Cells(r, 6).Value = birth
Cells(r, 6).NumberFormatLocal = "yyyy/mm/dd"
Cells(r, 7).Formula = "=IF(" & Cells(r, 6).Address & _
    "=""", "", DATEDIF(" & Cells(r, 6).Address & _
    ", TODAY(), ""Y""))"
'ラベルに現在のレコード番号とレコード数を表示
lblRec.Caption = rRng.Row - S_ROW
lblAll.Caption = "/" & Range("件数").Value

```

- ⑫変数 **rRng** の行番号を変数 **r** に代入する。
- ⑬テキストボックスの値をそれぞれ、変数で指定したセルに入力する。
txtNo → **r** 行 1 列目
txtName → **r** 行 2 列目
txtKana → **r** 行 3 列目
- ⑭オプションボタン **optMan** の値が
True の場合 → **r** 行 4 列目に「男」と表示する。
 そうでない場合 → **r** 行 4 列目に「女」と表示する。
- ⑮リストボックス **lstKubun** で選択されている値を変数 **r** 行 5 列目に表示する。
- ⑯変数 **birth** の値を変数 **r** 行 6 列目に表示する。
- ⑰ **r** 行 6 列目のセルの表示形式を「yyyy/mm/dd」に設定する。
- ⑱変数 **r** 行 7 列目のセルに **DATEDIF** 関数と **IF** 関数と組み合わせて「=IF(\$F\$23="", "", DATEDIF(\$F\$23, TODAY(), "Y"))」となるように、年齢を求める数式を入力する。
- ⑲変数 **rRng** の行番号から定数 **S_ROW** を引いてレコード番号を求め、ラベル **lblRec** に表示する。
- ⑳「/」と名前「件数」のセルの値をラベル **lblAll** に表示する（これで「レコード番号 / 全レコード数」が表示される）。

Column

DATEDIF 関数で年齢を求める式をセルに入力する

ワークシート関数の DATEDIF 関数を使って年齢を求める式を入力する場合、以下の書式で設定します。

書式 >> DATEDIF 関数

=DATEDIF(生年月日, TODAY(), "Y")

- 引数**
- 第1引数：生年月日
 - 第2引数：今日の日付を求める TODAY 関数
 - 第3引数：単位「Y」

第1引数でセルを参照する場合は、Range オブジェクトの Address プロパティを使ってセル番地を取得します。Address プロパティに引数を指定しないと絶対参照となります（DATEDIF 関数は p.350、Address プロパティは p.309 を参照）。

サンプルでは、生年月日が空欄の場合を考慮し、If 関数と組み合わせています。

● 新規レコードの場合

● 既存レコードの場合

既存の顧客 NO の場合は、そのレコードのセルを選択し、上書きを確認するメッセージを表示する。
 [はい] → そのレコードに上書きする
 [いいえ] → 何もせず終了

● Memo

既存のレコードで「顧客 NO」を変更し、重複がなければそのまま書き込まれます。

【顧客 NO で検索】 ボタンで検索したデータを表示する

【顧客 NO で検索】 ボタンをクリックするとテキストボックスに入力された顧客 NO を持つレコードを検索し、見つかったらユーザーフォームに読み込みます。顧客 NO は数値なので、数値が入力されたかどうかを `IsNumeric` 関数で調べ、数値の場合に、その値を引数にして【レコード読み込み】 プロシージャを呼び出します。

Sample 【顧客 NO で検索】 ボタン (btnKensaku) がクリックされた際の処理

```
Private Sub btnKensaku_Click()
    Dim i As Long
    If IsNumeric(txtKensaku.Value) = False Then
        MsgBox "顧客 NO を入力してください"
        Exit Sub
    End If
    Call レコード読み込み (txtKensaku.Value)
End Sub
```

- ① テキストボックス `txtKensaku` に入力された値が数値として扱えない場合は「顧客 NO を入力してください」とメッセージを表示して処理を終了する。
- ② `txtKensaku` に入力された値を引数にして【レコード読み込み】 プロシージャを呼び出す。



Column

ユーザーフォームに表示されているデータを削除する

ユーザーフォームに表示されているデータを削除するコードを紹介します。ユーザーフォームに【削除】 ボタン (btnDel) を追加し、「btnDel_Click」 イベント プロシージャでコードを記述します。

ここでは、削除するデータが最終レコードかどうかを確認し、最終レコードを削除した場合は、削除後にユーザーフォームに 1 つ上のレコードを表示し、最後でない場合は、1 つ下のレコードを表示するように引数を取得し、【レコード読み込み】 を呼び出します。

Sample 【削除】 ボタン (btnDel) がクリックされた際の処理

```
Private Sub btnDel_Click()
    Dim ans As Integer 'メッセージの戻り値用
    Dim no As Long '引数となる顧客 NO 代入用
    ans = MsgBox("削除すると元に戻せません" & vbCrLf & _
        "よろしいですか?", _
        vbYesNo + vbExclamation, "削除確認")
    If ans = vbNo Then Exit Sub
    If rRng.Row = Range("A" & Rows.Count).End(xlUp).Row Then
        no = rRng.Offset(-1).Value
    Else
        no = rRng.Offset(1).Value
    End If
    Range(rRng, rRng.Offset(, 6)).Delete xlShiftUp
    Call レコード読み込み (no)
End Sub
```

- ① 【削除すると元に戻せません】 [よろしいですか?] とメッセージを改行表示する。
- ② メッセージで [いいえ] ボタンがクリックされた場合、処理を終了する。
- ③ 変数 `rRng` (ユーザーフォームに表示しているデータのセル) の行番号が最終レコードの行と同じ場合 → 変数 `no` に変数 `rRng` の 1 つ上のセルの値を変数 `no` に代入する。
そうでない場合 → 変数 `no` に変数 `rRng` の 1 つ下のセルの値を代入する。
- ④ 変数 `rRng` のセルからそのセルの 6 列右側のセルまでを削除し、削除後、下の行を上シフトする。
- ⑤ 引数に変数 `no` を持たせて【レコード読み込み】 プロシージャを呼び出す。

02

簡易見積書作成システム

簡易見積書作成システムの概要と各シートの構成

本章では、見積書のような明細行のある表への効率的なデータ入力、一覧へのデータ転記や読み込みなどの処理を、Excelの機能とVBAの機能を組み合わせた簡易見積書作成システムを紹介します。まずは、これから作成するシステムの概要と各シートの構成を確認してください。

●【見積書】シート

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	見積書													
2				見積NO	1									
3				見積日	2022年07月28日									
4	あかさ商事株式会社			様										
5			C1001		株式会社	SB〇〇商事								
6	下記の通りお見積り申し上げます。				〒106-0032	東京都港区六本木 x - x - x								
7	見積金額		¥77,000		TEL: 03- x x x - x x x x									
8														
9	NO	商品NO	商品名		単価	数量	金額							
10	1	DE1001	事務用デスク		¥15,000	2	¥30,000							
11	2	CH2001	事務用チェア		¥20,000	2	¥40,000							
12														
13														
14														
15														
16						小計	¥70,000							
17						消費税 (10%)	¥7,000							
18						合計	¥77,000							
19														
20														

データクリア

新規入力

取引先入力

商品入力

データ転記

見積書NO

検索

印刷

1

2

3

4

5

6

7

8

9

<

>

見積書

見積一覧

取引先マスタ

商品マスタ

+

:

<

右側に配置したボタンをクリックして各処理を行います。

●【見積書】シートの項目

No	構成	内容
①	見積書	見積書のデータの入力、表示、印刷用。
②	[データクリア] ボタン	[データクリア] プロシージャ。見積 NO、見積日、取引先名、取引先 NO、明細データを消去する。
③	[新規入力] ボタン	[新規入力] プロシージャ。見積 NO の採番と見積日を入力する。
④	[取引先入力] ボタン	ユーザーフォーム [取引先] を表示し、取引先、取引先 NO を入力する。

①【顧客 No12】[松本敏次] が表示された状態で [削除] ボタンをクリックする

②確認メッセージで [はい] を選択する

③表示されていたデータが削除され、1つ下のレコードがフォームに表示される

No	構成	内容
⑤	[商品入力] ボタン	ユーザーフォーム [商品] を表示し、商品 NO、商品名、単価を入力する。
⑥	[データ転記] ボタン	見積書の内容を [見積一覧] シートの一覧に転記する。
⑦	[検索] ボタン	名前 [検索 NO] のセルに入力された値を [見積一覧] シートの一覧から検索し、[見積書] シートに読み込む
⑧	[印刷] ボタン	印刷設定をして、印刷を実行する。
⑨	名前 [検索 NO] のセル	セル I14 に名前 [検索 NO] を設定し、検索用の見積書 NO を入力する。

● [見積一覧] シート



セル A3 に名前「見積 NO」を設定

見積NO	見積日	取引先NO	取引先名	商品NO	商品名	単価	数量	金額
1	2022/07/28	C1001	あかさ商事株式会社	DE1001	事務用デスク	¥15,000	2	¥30,000
2	2022/07/28	C1001	あかさ商事株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000
3	2022/07/28	C1005	AAA商会株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000
4	2022/07/28	C1003	株式会社マクスワイ	DE1001	事務用デスク	¥15,000	5	¥75,000
5	2022/07/28	C1003	株式会社マクスワイ	CH2001	事務用チェア	¥20,000	5	¥100,000
6	2022/07/28	C1003	株式会社マクスワイ	PT4001	パーティション	¥9,000	1	¥9,000
7	2022/07/28	C1002	トッデイ工業株式会社	CH2001	事務用チェア	¥20,000	1	¥20,000
8	2022/07/28	C1002	トッデイ工業株式会社	WA3001	ワゴン (3段)	¥10,000	1	¥10,000
9	2022/07/28	C1002	トッデイ工業株式会社	WA3002	ワゴン (4段)	¥14,000	1	¥14,000
10	2022/07/29	C1004	株式会社天空産業	CH2001	事務用チェア	¥20,000	3	¥60,000
11	2022/07/29	C1004	株式会社天空産業	WA3001	ワゴン (3段)	¥10,000	3	¥30,000

[データ転記] ボタン ([見積書] シートの⑥) をクリックすると、見積書の内容が [見積一覧] シートに書き出されます。

● [取引先マスター] シート

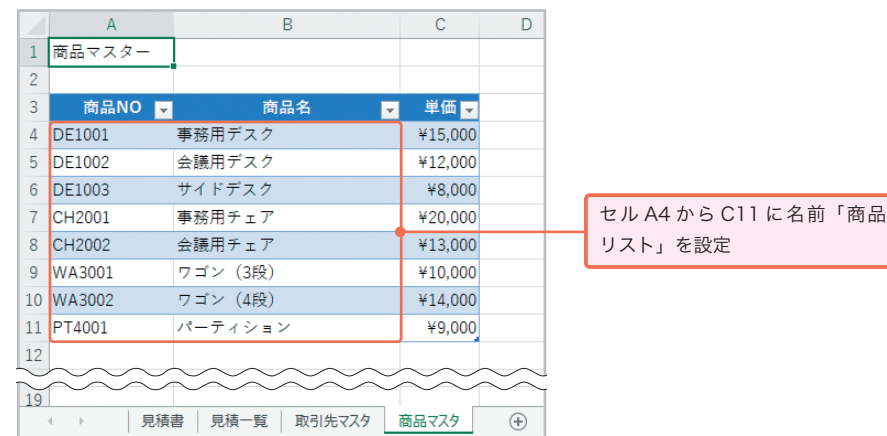


セル A4 ~ B8 に名前「取引先リスト」を設定

取引先NO	会社名	担当者
C1001	あかさ商事株式会社	田中 太郎
C1002	トッデイ工業株式会社	後藤 慎一郎
C1003	株式会社マクスワイ	山下 紀子
C1004	株式会社天空産業	鈴木 徹
C1005	AAA商会株式会社	斉藤 浩二

テーブル形式の取引先の一覧表です。セル A4 ~ B8 の値を [取引先入力] ボタン ([見積書] シートの④) をクリックしたときに開くユーザーフォームのリストボックスに表示します。テーブルにしているため、名前「取引先リスト」のセル範囲は、データの増減に対応できます。

● [商品マスタ] シート



セル A4 から C11 に名前「商品リスト」を設定

商品NO	商品名	単価
DE1001	事務用デスク	¥15,000
DE1002	会議用デスク	¥12,000
DE1003	サイドデスク	¥8,000
CH2001	事務用チェア	¥20,000
CH2002	会議用チェア	¥13,000
WA3001	ワゴン (3段)	¥10,000
WA3002	ワゴン (4段)	¥14,000
PT4001	パーティション	¥9,000

テーブル形式の商品の一覧表です。セル A4 ~ C11 の値を [商品入力] ボタン ([見積書] シートの④) をクリックしたときに表示するユーザーフォームのリストボックスに表示します。テーブルにしているため、名前「商品リスト」のセル範囲は、データの増減に対応できます。

見積書に設定されている計算式

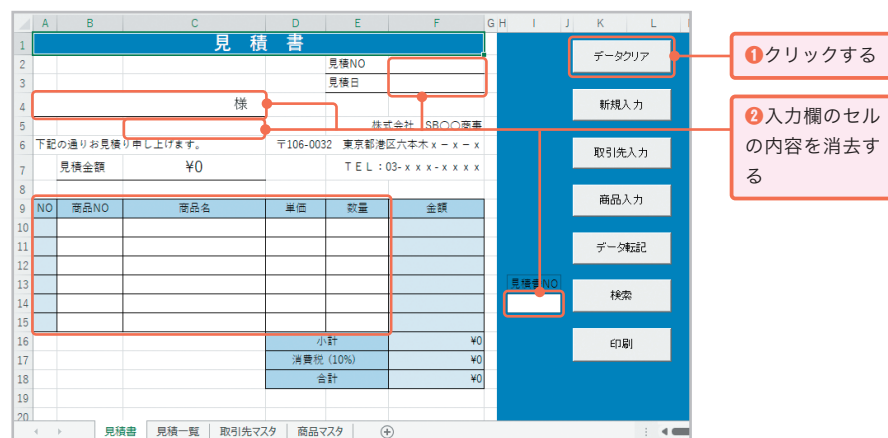
[見積書] シートには、次のような計算式が設定されています。見積書に商品 NO や数量などのデータが入力されたら自動的に表示・計算されます。

また、シートの保護を解除するには、[校閲] タブの [シート保護の解除] をクリックします。パスワードを設定した場合は、パスワード入力画面が表示されるので設定したパスワードを入力します。パスワードを忘れてしまうと解除できなくなるので気を付けてください。

【データクリア】ボタンで入力欄のデータを消去する

【データクリア】ボタンには【データクリア】プロシージャを割り当てます。【データクリア】ボタンをクリックすると、入力欄となるセル（セル F2、F3、A4、C5、B10～E15、I14）の内容を消去します。

なお、セル A4 はセル A4～C4 が結合されており、表示されている「様」は、図形のテキストボックスを配置しているため、消去されません。



【データクリア】プロシージャは以下のようになります。入力欄のセルに対して **ClearContents** メソッドでセルの内容を消去しています。セル A4 は結合セルであるため、**MergeArea** プロパティを使って結合セルを参照しています。

Sample 【データクリア】プロシージャ

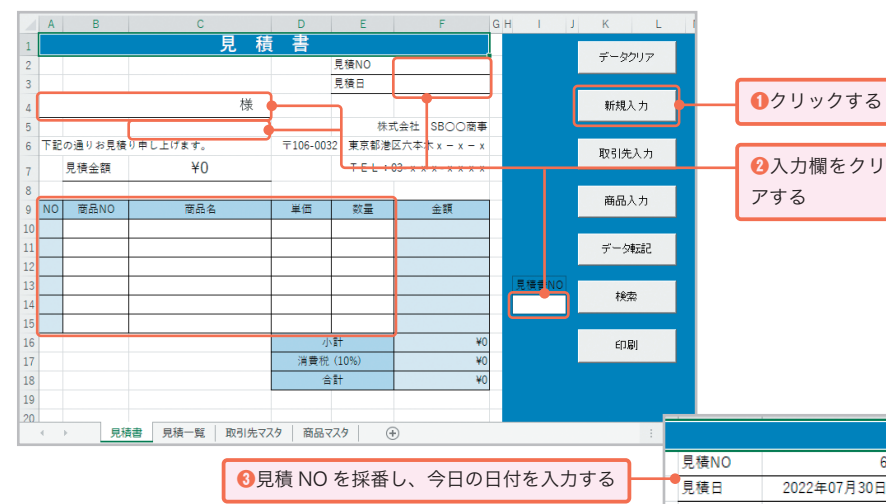
```
ap-02-01.xlsm
Sub データクリア ()
    With Worksheets("見積書")
        Range("F2:F3").ClearContents
        Range("A4").MergeArea.ClearContents
        Range("C5").ClearContents
        Range("B10:E15").ClearContents
        Range("検索 NO").ClearContents
    End With
End Sub
```

Memo

結合セルを参照する場合、MergeArea プロパティを使う以外に、結合されているすべてのセルを参照しても処理できます。ここではセル A4～C4 が結合されているので、「Range("A4:C4").ClearContents」と記述できます。

【新規入力】ボタンで 新規入力の準備をする

【新規入力】ボタンには【新規入力】プロシージャを割り当てます。【新規入力】ボタンをクリックすると、【データクリア】プロシージャを呼び出して、入力欄のデータを消去し、見積 NO を採番してセル F2 に入力し、セル F3 の見積日に今日の日付を入力します。



Sample 【新規入力】 プロシージャ

```

ap-02-01.xlsm
Sub 新規入力 ()
    Dim newID As Long
    Call データクリア
    newID = Application.WorksheetFunction.Max( _
        Range("見積 NO").CurrentRegion.Columns(1)) + 1
    Range("F2").Value = newID
    Range("F3").Value = Date
End Sub

```

- MAX 関数を使って名前「見積 NO」のセルを含む表全体の 1 列目の中から最大値を求め、1 を加算して変数 newID に代入する。
→変数 newID が採番された見積 NO になる。
- セル F2 に変数 newID の値を表示する。
- セル F3 に Date 関数で今日の日付を表示する。

Memo

【見積一覧】シートのセル A3 に名前「見積 NO」が設定されています。

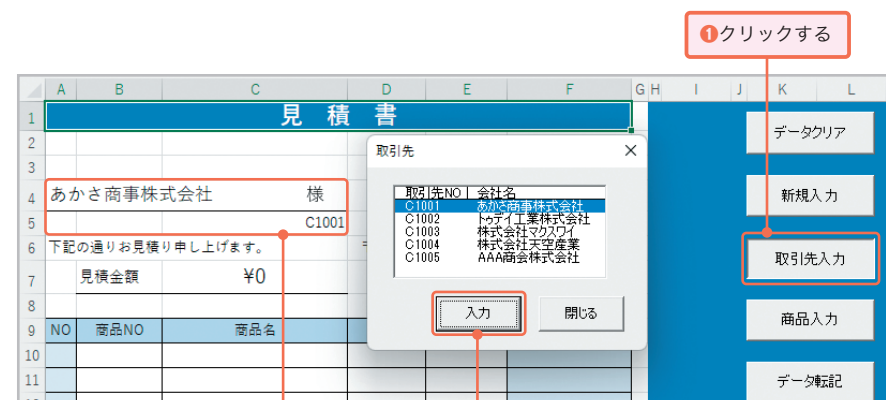
● 採番の仕組み

見積NO	見積日	取引先NO	取引先名	商品NO	商品名	単価	数量	金額
1	2022/07/28	C1001	あかさ商事株式会社	DE1001	事務用デスク	¥15,000	2	¥30,000
1	2022/07/28	C1001	あかさ商事株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000
2	2022/07/28	C1005	AAA商会株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000
3	2022/07/28	C1003	株式会社マクスワイ	DE1001	事務用デスク	¥15,000	5	¥75,000
3	2022/07/28	C1003	株式会社マクスワイ	CH2001	事務用チェア	¥20,000	5	¥100,000
3	2022/07/28	C1003	株式会社マクスワイ	PT4001	パーティション	¥9,000	1	¥9,000
4	2022/07/28	C1002	トゥデイ工業株式会社	CH2001	事務用チェア	¥20,000	1	¥20,000
4	2022/07/28	C1002	トゥデイ工業株式会社	WA3001	ワゴン (3段)	¥10,000	1	¥10,000
4	2022/07/28	C1002	トゥデイ工業株式会社	WA3002	ワゴン (4段)	¥14,000	1	¥14,000
5	2022/07/29	C1004	株式会社天空産業	CH2001	事務用チェア	¥20,000	3	¥60,000
5	2022/07/29	C1004	株式会社天空産業	WA3001	ワゴン (3段)	¥10,000	3	¥30,000

Application.WorksheetFunction.Max(Range("見積 NO").CurrentRegion.Columns(1)) + 1
名前が「見積 No」のセルを含む表全体の 1 列目の中で最大値を求め、1 を加算して見積番号を採番する。

【取引先入力】 ボタンとで取引先を一覧から選択・入力する

【取引先入力】 ボタンをクリックすると、取引先 NO と取引先名が一覧表示されているユーザーフォームを表示し、追加する取引先を選択して【入力】 ボタンをクリックすると見積書のセルに表示する仕組みを作ります。



④ 取引先名 (A4) と取引先 No (C5) が
見積書に入力される

② フォームの一覧から取引先を選択し、
【入力】 をクリックする

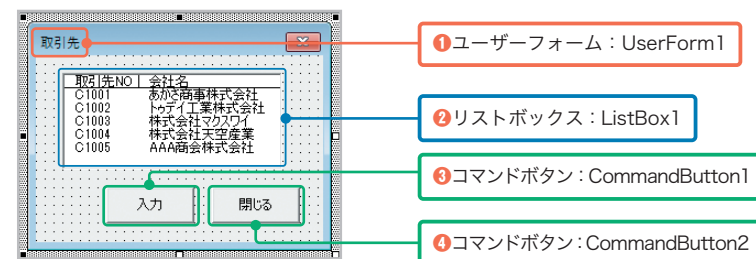
Sample 【取引先入力】 ボタンに割り当てられているプロシージャ

```

Sub 取引先フォーム表示 ()
    UserForm1.Show 'UserForm1を開く'
End Sub

```

【取引先】 フォームは、リストボックスとコマンドボタンを以下のように配置して作成します。



● 各コントロールの[プロパティウィンドウ]での設定内容

ユーザーフォーム / コントロール名	プロパティ	設定値
① ユーザーフォーム	(オブジェクト名)	UserForm1
	Caption	取引先
② リストボックス	(オブジェクト名)	ListBox1
	ColumnCount	2
	ColumnHeads	True
	ColumnWidths	55pt;70 pt
	RowSource	取引先リスト
③ コマンドボタン	(オブジェクト名)	CommandButton1
	Caption	入力
④ コマンドボタン	(オブジェクト名)	CommandButton2
	Caption	閉じる

Column

リストボックスの設定

ここでは、リストボックスを2列に設定し、選択肢には、名前「取引先リスト」が付いているセル範囲を指定しています。

リストボックスを2列に設定するには、ColumnCount プロパティで列数を2に指定し、列見出しを表示するには ColumnsHeads を True にします。

ColumnWidths プロパティで1列目と2列目をセミコロン「;」で区切って「55pt;70pt」のようにポイント単位で指定します。

ユーザーフォームでリストボックスの値を選択し、[入力] ボタン (CommandButton1) をクリックしたときに取引先 NO →セル C5、取引先名→セル A4 に表示したいです。そのため、リストボックスで選択された行の1列目、2列目の値をそれぞれセル C5、A4 の Value プロパティに代入します。

リストボックスで選択された行は ListIndex プロパティで取得できます。複数列の場合は、List プロパティを使って、リストボックス内の行と列を指定して取得します。行、列ともに1行目、1列目は0から順番に数えます。

書式 >> List プロパティの構文

ListBoxオブジェクト.List (Row,Column)

引数 Row : リストの行を、先頭行を0として上から何番目か数値で指定。
リストボックスのListIndex プロパティで取得した値を指定できる。
Column : リストの列を、先頭列を0として左から何番目か数値で指定。

Sample 【入力】 ボタン (CommandButton1) クリック時のイベントプロシージャ

```
Private Sub CommandButton1_Click()  
    Range("A4").Value = ListBox1.List(ListBox1.ListIndex, 1) ①  
    Range("C5").Value = ListBox1.List(ListBox1.ListIndex, 0) ②  
End Sub
```

① リストボックスで選択された行の2列目の値をセル A4 に表示する。

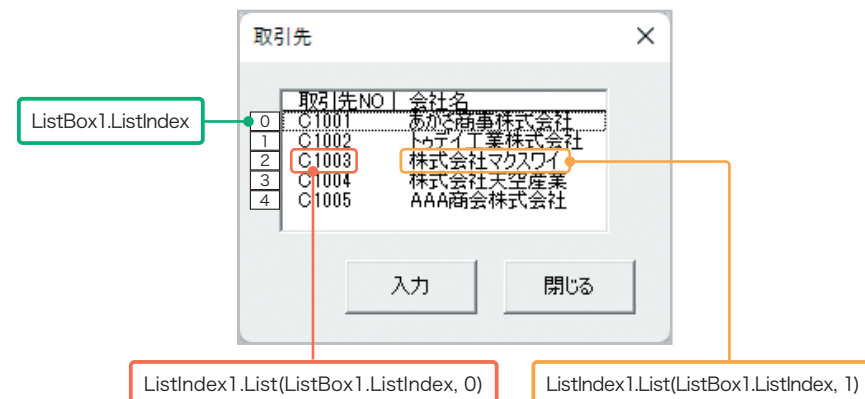
② リストボックスで選択された行の1列目の値をセル C5 に表示する。

セル A4 は結合セルですが、データを入力する場合は、結合セル内の先頭のセルを指定するだけで設定できます。

Sample 【閉じる】 ボタン (CommandButton2) クリック時のイベントプロシージャ

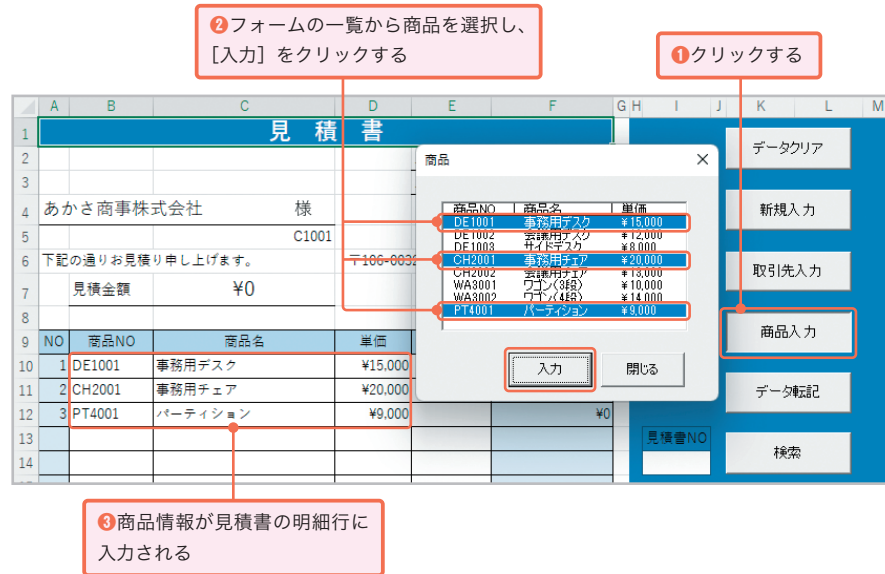
```
Private Sub CommandButton2_Click()  
    Unload Me 'ユーザーフォームを閉じる  
End Sub
```

● リストボックスの列の値を取得する



「商品入力」ボタンで商品を一覧から複数選択して入力

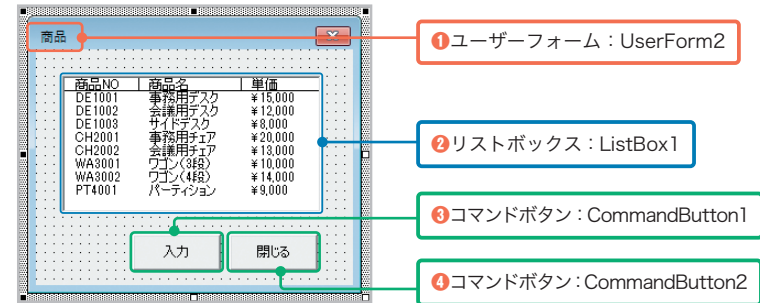
「商品入力」ボタンをクリックすると、商品 NO、商品名、単価が一覧表示されているユーザーフォームを表示し、追加する商品を複数選択して「入力」ボタンをクリックすると見積書の明細行のセルに表示する仕組みを作ります。



Sample 「商品入力」ボタンに割り当てられているプロシージャ

```
Sub 商品フォーム表示 ()
    UserForm2.Show 'UserForm2を開く
End Sub
```

「商品」フォームは、リストボックスとコマンドボタンを以下のように配置して作成します。リストボックスは、名前「商品リスト」のセル範囲を選択肢として表示し、3 列表示するため **ColumnCount プロパティ** を 3 に設定し、列見出しを表示するので **ColumnHeads プロパティ** を True にします。また、ここでは複数選択できるように **MultiSelect プロパティ** を True にします。



● 各コントロールの「プロパティウィンドウ」での設定内容

ユーザーフォーム / コントロール名		プロパティ	設定値
①	ユーザーフォーム	(オブジェクト名)	UserForm2
		Caption	商品
②	リストボックス	(オブジェクト名)	ListBox1
		ColumnCount	3
		ColumnHeads	True
		ColumnWidths	49.95 pt;70 pt;49.95 pt
		MultiSelect	fmMultiSelectMulti
		RowSource	商品リスト
③	コマンドボタン	(オブジェクト名)	CommandButton1
		Caption	入力
④	コマンドボタン	(オブジェクト名)	CommandButton2
		Caption	閉じる

ユーザーフォームでリストボックスの値を複数選択し、「入力」ボタン (CommandButton1) をクリックしたときに明細行に商品 NO、商品名、単価が入力されるようにするには、**List プロパティ**と **ListIndex プロパティ**を使ってリストボックスの列の値を取得します。

ここでは、選択された複数の商品を明細行に追加しています。

リストボックスの項目が選択されているかどうかを、1 つ目 (0) から、最後の項目 (ListBox1.ListCount-1) まで繰り返し処理で確認し、選択されていた場合に書き込みます。

また、選択されているかどうかは、リストボックスの **Selected プロパティ**を使います。例えば、「ListBox1.Selected(0)」が True の場合、1 つ目の項目が選択されています。

Sample [入力] ボタン (CommandButton1) クリック時のイベントプロシージャ

```

Private Sub CommandButton1_Click()
    Dim meisaiCell As Range '明細行の先頭のセル代入用
    Dim i As Integer        '繰り返し処理用
    Range("B10:E15").ClearContents '明細範囲クリア
    Set meisaiCell = Range("B10")
    For i = 0 To ListBox1.ListCount - 1
        If ListBox1.Selected(i) Then
            meisaiCell.Value = ListBox1.List(i, 0)
            meisaiCell.Offset(, 1).Value = ListBox1.List(i, 1)
            meisaiCell.Offset(, 2).Value = ListBox1.List(i, 2)
            Set meisaiCell = meisaiCell.Offset(1)
        End If
    Next
    Set meisaiCell = Nothing 'オブジェクト変数の解放
End Sub

```

- ①セル B10 (明細行の先頭) を変数 **meisaiCell** に代入する。
- ②変数 **i** が 0 からリストボックスの項目数 -1 の数になるまで以下の処理を繰り返す (リストボックスの項目は上から 0, 1, 2... と数えるため、0 から始まり項目数から 1 を引いた数ですべての項目を参照することになる)。
- ③リストボックスの **i** 番目の項目が選択されている場合、以下の処理を実行する。
- ④リストボックスの
 - i+1** 行目 1 列目の値を変数 **meisaiCell** のセルに入力する。
 - i+1** 行目 2 列目の値を変数 **meisaiCell** から 1 つ右のセルに入力する。
 - i+1** 行目 3 列目の値を変数 **meisaiCell** から 2 つ右のセルに入力する。
- ⑤変数 **meisaiCell** の 1 つ下のセルを変数 **meisaiCell** に設定し直し、繰り返しの処理 (②) に戻る。

Memo

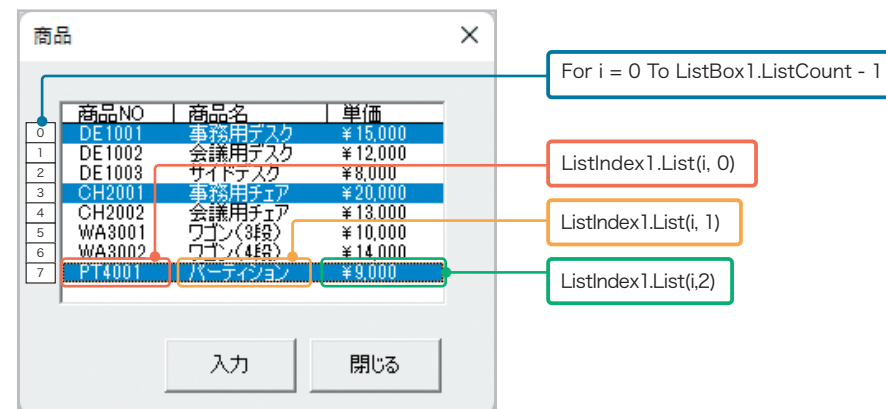
「**i+1** 行目」について、リストボックスの項目は 1 番上が 0 であり、変数 **i** は 0 から数えているので、1 を加算して実際の行数に直して説明しています。

Sample [閉じる] ボタン (CommandButton2) クリック時のイベントプロシージャ

```

Private Sub CommandButton2_Click()
    Range("E10").Select '数値入力セルを選択
    Unload Me
End Sub

```

**Column** ListCount プロパティと ListIndex プロパティ

リストボックスの **ListCount** プロパティはリストボックス内の項目数を返します。実際の数を返すので、3 項目あれば 3 が返ります。取得のみです。

一方、**ListIndex** プロパティは、現在選択されている項目を取得します。

ListIndex の値が 0 の時は 1 つ目、1 の時は 2 つ目が選択されています。何も選択されていない場合は、-1 が返ります。取得と設定ができます。

Column リストボックスの Selected プロパティ

リストボックスの **Selected** プロパティは、引数で指定したインデックス番号の項目が選択されていれば True、選択されていなければ False が返ります。例えば、1 つ目の項目が選択されているかどうかは「**ListBox1.Selected(0)**」で調べられます。

[データ転記] ボタンで見積書のデータを一覧に書き込む

[データ転記] ボタンをクリックすると、**[見積書]** シートの入力データチェック、見積 NO の重複チェックを行った後、**[見積一覧]** シートに転記する仕組みを作ります。

①クリックする

②書き込み

今回作成する「データ転記」プロシーダの処理内容は

- ①…入力データのチェック
- ②…見積 No の重複チェック
- ③…データ転記

の3パートに分かれています。それぞれについて見ていきましょう。

見積書に記入された内容を見積一覧に転記する前に、**入力欄に空欄があるかどうかを確認**するためのコードが必要です。ここでは、「データ転記」プロシーダにおける「入力データのチェック」を紹介します。

Sample 「データ転記」プロシーダ ①入力データのチェック

Sub データ転記 ()

```
'その① 空欄チェック *****
Dim meisaiRng As Range '明細入力範囲代入用 その①、③で使用
Dim cnt As Long '明細件数代入用 その①、③で使用
Dim ary As Variant '繰り返し処理用
Dim rArray As Variant
Dim rng As Range
```

'A: 見積 NO～取引 NO の空欄チェック

```
rArray = Array(Range("F2"), Range("F3"), Range("A4"), _
Range("C5"))
For Each ary In rArray
If ary.Value = "" Then
ary.Select
MsgBox "未入力箇所があります。", vbExclamation, "入力チェック"
Exit Sub
End If
Next
```

'B: 明細行の空欄チェック

```
Set meisaiRng = Range("B10:E15")
cnt = Application.WorksheetFunction.CountA(meisaiRng.Columns(1))
'データが1件も入力されていない場合
If cnt = 0 Then
Range("B10").Select
MsgBox "明細行が入力されていません", vbExclamation, "明細チェック"
Exit Sub
Else
'データが入力されていた場合、空欄が1つでもあったらキャンセル
Set meisaiRng = meisaiRng.Resize(cnt)
For Each rng In meisaiRng
If rng.Value = "" Then
rng.Select
MsgBox "明細行に空欄があります", vbExclamation, "明細チェック"
Exit For
End If
Next
End If
```

```
MsgBox "入力漏れチェック完了"
```

```
'その② 見積 No 重複チェック *****
'その③ データ転記 *****
End Sub
```

ブロック A

見積 NO、日付、取引先、取引先 NO のセルの空欄チェックをします。

'A: 見積 NO~取引 NOの空欄チェック

```

rArray = Array(Range("F2"), Range("F3"), Range("A4"), _
               Range("C5"))
For Each ary In rArray
    If ary.Value = "" Then
        ary.Select
        MsgBox "未入力箇所があります。", vbExclamation, "入力チェック"
        Exit Sub
    End If
Next

```

① Array 関数を使ってセル F2、F3、A4、C5 を配列変数にし、Variant 型の変数 rArray に代入する。

② 配列 rArray の各要素を順番に変数 ary に代入しながら次の処理を実行する。

変数 ary の値が空欄の場合

→変数 ary を選択し、「未入力箇所があります」とメッセージを表示して処理を終了する。

ブロック B

明細行のセルの空欄をチェックします。

'B: 明細行の空欄チェック

```

Set meisaiRng = Range("B10:E15")
cnt = Application.WorksheetFunction.CountA(meisaiRng.Columns(1))
'データが 1 件も入力されていない場合
If cnt = 0 Then
    Range("B10").Select
    MsgBox "明細行が入力されていません", vbExclamation, "明細チェック"
    Exit Sub
Else
    'データが入力されていた場合、空欄が 1 つでもあったらキャンセル
    Set meisaiRng = meisaiRng.Resize(cnt)
    For Each rng In meisaiRng
        If rng.Value = "" Then
            rng.Select
            MsgBox "明細行に空欄があります", vbExclamation, "明細チェック"
            Exit For
        End If
    Next
End If

MsgBox "入力漏れチェック完了"

```

③ セル範囲 B10 から E15 を変数 meisaiRng に代入する。

④ 変数 meisaiRng の 1 列目のデータ数を COUNTA 関数で取得し、変数 cnt に代入する。

⑤ 変数 cnt の値が 0 の場合（明細行にデータが入力されていない場合）

→セル B10 を選択して、「明細行が入力されていません」とメッセージを表示して処理を終了する。

⑥ 0 でない場合（明細行にデータが入力されている場合）

→変数 meisaiRng のセル範囲の行数を、データ件数分にサイズ変更する。

⑦ 変数 meisaiRng の各セルを変数 rng に代入しながら次の処理を繰り返す。

⑧ セル rng の値が空欄の場合

→セル rng を選択し、「明細行に空欄があります」とメッセージを表示して処理を終了する。

⑨ 処理が終了したことを確認するためにメッセージを表示する（完成時、この行は削除する）。

① セル F2、F3、A4、C5 を配列にして rArray に代入する

② For Each ステートメントで空欄かどうかチェックする（空欄の場合、メッセージを表示して処理を終了する）

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2						見積NO	5						データクリア
3						見積日	2022年07月29日						新規入力
4			株式会社天空産業	様									
5			C1004										
6			下記の通りお見積り申し上げます。			〒106-0032 東京都港区六							
7			見積金額	¥99,000		TEL: 03-							
8													
9	NO	商品NO	商品名	単価	数量	金額							商品入力
10	1	CH2001	事務用チェア	¥20,000	3	¥60,000							データ確認
11	2	WA3001	ワゴン (3段)	¥10,000	3	¥30,000							見積書NO
12													検索
13													印刷
14													
15													
16						小計	¥90,000						
17						消費税 (10%)	¥9,000						
18						合計	¥99,000						
19													

④ cnt = Application.WorksheetFunction.CountA(meisaiRng.Columns(1))
meisaiRng(B10 ~ E15) の 1 列目のデータの個数を数えて変数 cnt へ代入する

⑤ cnt が 0 の場合、メッセージを表示して処理を終了する

⑥ Set meisaiRng = meisaiRng.Resize(cnt)
cnt が 0 でない場合、変数 meisaiRng の行数を cnt（ここでは「2」）に設定変更してデータが入力されている範囲だけを取得し、⑦~⑧それぞれのセルの中で空欄がないかチェックする（空欄があればメッセージ文を表示して処理を終了する）

入力データのチェックが終わったら、データを転記する前に、**一覧表と同じ見積 NO があるかどうか**を確認します。

ここでは、見積 No の重複をチェックする部分を作成します。

同じ NO が見つかった場合はデータを置き換えるかどうかを確認します。置き換えない場合は重複するセルを選択して終了し、置き換える場合は一覧表で重複するデータを削除します。

Sample 【データ転記】 プロシージャ ②見積 NO の重複チェック

Sub データ転記 ()

'その① 空欄チェック *****

'その② 見積 No重複チェック *****

Dim mNo As Long '見積 Noチェック用

Dim rng2 As Range '見積 No重複セル格納用

Dim ans As Integer 'msgbox関数の戻り値格納用

Dim r As Integer '重複番号の行数取得

mNo = Range("F2").Value ①

With Range("見積 NO").CurrentRegion

Set rng2 = .Columns(1).Find(what:=mNo, lookat:=xlWhole) ②

If Not rng2 Is Nothing Then ③

ans = MsgBox("同じ見積番号があります。データを更新しますか？", _
vbYesNo, "見積番号の重複") ④

If ans = vbNo Then ⑤

MsgBox "転記をキャンセルします。"

Application.Goto rng2

Exit Sub

Else

r = Application.WorksheetFunction.CountIf(.Columns(1), mNo) ⑥

rng2.Resize(r, 9).Delete xlShiftUp ⑦

End If

End If

End With

MsgBox "重複 NOチェック完了" ⑧

'その③ データ転記 *****

End Sub

- ①セル F2 の値（見積 NO の数値）を変数 mNo に代入する。
- ②名前「見積 NO」のセル（「見積一覧」シートのセル A3）を含む表について、1 列目の中で変数 mNo の値を検索し、見つかったセルを変数 rng2 に代入する。
- ③変数 rng2 が Nothing でない場合（重複データのセルが見つかった場合）、以下の処理を実行する。
- ④更新確認メッセージを表示する。
- ⑤ [いいえ] ボタンがクリックされた場合
→キャンセルメッセージを表示後、変数 rng2 のセルを選択して終了する。
- ⑥ そうでなかった場合（[はい] ボタンがクリックされた場合）
→見積一覧表の 1 列目の中で、変数 mNo と同じ数値の個数を数えて変数 r に代入する。
- ⑦ 変数 rng2 のセルを基準に r 行、9 列のセル範囲を削除後、下のセルを上へシフトする。
- ⑧ 処理の終了を確認するためにメッセージ表示する（完成時は削除する）。

● 見積 NO の重複が見つかり、データを更新する場合の処理

A	B	C	D	E	F	G
1						
2					見積 NO	4
3					見積日	2022年07月28日

① mNo=Range("F2")

② Set rng2 =Range(" 見積 NO").CurrentRegion .Columns(1).Find(_
what:=mNo, lookat:=xlWhole)
見積一覧の 1 列目で mNo（4）を探して見つかったセルを rng2 に代入する

A	B	C	D	E	F	G	H	I	J
1	見積一覧								
2									
3	見積 NO	見積日	取引先 NO	取引先名	商品 NO	商品名	単価	数量	金額
4	1	2022/07/28	C1001	あかさ商事株式会社	DE1001	事務用デスク	¥15,000	2	¥30,000
5	1	2022/07/28	C1001	あかさ商事株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000
6	2	2022/07/28	C1005	AAA商会株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000
7	3	2022/07/28	C1003	株式会社マクスワイ	DE1001	事務用デスク	¥15,000	5	¥75,000
8	3	2022/07/28	C1003	株式会社マクスワイ	CH2001	事務用チェア	¥20,000	5	¥100,000
9	3	2022/07/28	C1003	株式会社マクスワイ	PT4001	パーティション	¥9,000	1	¥9,000
10	4	2022/07/28	C1002	トゥデイ工業株式会社	CH2001	事務用チェア	¥20,000	1	¥20,000
11	4	2022/07/28	C1002	トゥデイ工業株式会社	WA3001	ワゴン（3段）	¥10,000	1	¥10,000
12	4	2022/07/28	C1002	トゥデイ工業株式会社	WA3002	ワゴン（4段）	¥14,000	1	¥14,000
13	5	2022/07/29	C1004	株式会社天空産業	CH2001	事務用チェア	¥20,000	3	¥60,000
14	5	2022/07/29	C1004	株式会社天空産業	WA3001	ワゴン（3段）	¥10,000	3	¥30,000
15									

rng2

rng2.Resize(r, 9) : セル rng2 を
基準に r 行、9 列のセル範囲

r

⑥ r = Application.WorksheetFunction.CountIf(.Columns(1), mNo)

⑦ rng2.Resize(r, 8).Delete xlShiftUp

重複データを更新（置き換える）場合は、mNo と同じ値をもつセルを数えて r に代入し（③）、重複データ（rng2.Resize(r, 9) まで）を削除する

見積書のデータを見積一覧表に転記していきます。

データの転記先の先頭行を求め、見積 NO、日付、取引先、取引先 NO の転記と明細行の転記を分けています。

Sample [データ転記] プロシージャ ③データ転記部分

Sub データ転記()

'その① 空欄チェック *****

```
Dim cnt As Long           '明細件数代入用 その①、③で使用
Dim meisaiRng As Range    '明細入力範囲代入用 その①、③で使用
: (省略)
```

'その② 見積 No重複チェック *****

'その③ データ転記 *****

```
Dim fRow As Long          'データ転記先頭行用
Dim copyRng1 As Variant, copyRng2 As Variant '見積書のコピー元用
Dim dataRng1 As Range, dataRng2 As Range      '見積一覧の転記先用
```

'A: 見積 NO、日付、取引先、取引先 NOの転記

With Worksheets("見積書")

```
copyRng1 = Array(.Range("F2").Value, .Range("F3").Value, _
                 .Range("C5").Value, .Range("A4").Value) ①
```

```
copyRng2 = meisaiRng.Resize(, 5).Value '明細行 ②
```

End With

'B: 明細行の転記

With Worksheets("見積一覧")

```
fRow = .Range("A" & .Rows.Count).End(xlUp).Offset(1).Row ③
```

```
Set dataRng1 = .Range(.Cells(fRow, "A"), _
                    .Cells(fRow + cnt - 1, "D")) ④
```

```
Set dataRng2 = .Range(.Cells(fRow, "E"), _
                    .Cells(fRow + cnt - 1, "I")) ⑤
```

```
dataRng1.Value = copyRng1 ⑥
```

```
dataRng2.Value = copyRng2 ⑦
```

.Select

```
Range("見積 NO").Sort Key1:=Range("A3"), _
                     Order1:=xlAscending, Header:=xlYes ⑧
```

Range("見積 NO").Select

End WithEnd Sub

☑ブロック A

見積 NO、日付、取引先、取引先 NO を転記します。

'A: 見積 NO、日付、取引先、取引先 NOの転記

With Worksheets("見積書")

```
copyRng1 = Array(.Range("F2").Value, .Range("F3").Value, _
                 .Range("C5").Value, .Range("A4").Value) ①
```

```
copyRng2 = meisaiRng.Resize(, 5).Value '明細行 ②
```

End With

①[見積書] シートのセル F2 (見積 NO)、F3 (見積日)、C5 (取引先)、A4 (取引先 NO) の値を、配列変数 copyRng1 に代入する。

②変数 meisaiRng のセル範囲を列数を 5 列にサイズ調整したセル範囲の値を、配列変数 copyRng2 に代入する。

(変数 meisaiRng には商品から数量までの明細行が代入されているため、列数を 5 に変更して金額まで含めて、変数 copyRng2 に代入している。)

	A	B	C	D	E	F	G
1	見積書						
2				見積NO		5	
3				見積日	2022年07月29日		
4	株式会社天空産業	様					
5		C1004		株式会社	SB〇〇商事		
6	下記の通りお見積り申し上げます。			〒106-0032	東京都港区六本木 x - x - x		
7	見積金額	¥99,000		TEL : 03-x x x - x x x x			
8							
9	NO	商品NO	商品名	単価	数量	金額	
10	1	CH2001	事務用チェア	¥20,000	3	¥60,000	
11	2	WA3001	ワゴン (3段)	¥10,000	3	¥30,000	
12							
13							
14							
15							
16				小計		¥90,000	

①変数 copyRng1 に代入する
copyRng1=Array("5","2022
年 07 月 29 日","C1004","株
式会社天空産業")② meisaiRng のセル範囲 (商品 NO から数量まで) の列数を 5 にサイズ
変更した範囲 (金額列まで) の値を配列変数 copyRng2 に代入する

☑ブロック B

明細行を転記します。

'B: 明細行の転記

With Worksheets("見積一覧")

```
fRow = .Range("A" & .Rows.Count).End(xlUp).Offset(1).Row ③
```

```
Set dataRng1 = .Range(.Cells(fRow, "A"), _
                    .Cells(fRow + cnt - 1, "D")) ④
```

```
Set dataRng2 = .Range(.Cells(fRow, "E"), _
                    .Cells(fRow + cnt - 1, "I")) ⑤
```

```
dataRng1.Value = copyRng1 ⑥
```

```
dataRng2.Value = copyRng2 ⑦
```

.Select

```
Range("見積 NO").Sort Key1:=Range("A3"), _
                     Order1:=xlAscending, Header:=xlYes ⑧
```

Range("見積 NO").Select

End WithEnd Sub


```

End If
Call データクリア 'データを読み込む前にデータを削除
srcRow = srcRng.Row '見つかったセルの行番号を調べる
cnt2 = Application.WorksheetFunction.CountIf( _
    Range("見積 NO").CurrentRegion.Columns(1), srcNo)
With Worksheets("見積一覧")
    Range("F2").Value = .Cells(srcRow, "A").Value '見積 NO
    Range("F3").Value = .Cells(srcRow, "B").Value '受注日
    Range("A4").Value = .Cells(srcRow, "D").Value '取引先名
    Range("C5").Value = .Cells(srcRow, "C").Value '取引先 NO
    sData = .Range(.Cells(srcRow, "E"), _
        .Cells(srcRow + cnt2 - 1, "H")).Value
End With
Range(Cells(10, "B"), Cells(10 + cnt2 - 1, "E")).Value = sData
End Sub

```

- ① 名前「検索 NO」の値を変数 **srcNo** に代入する。
- ② 名前「見積 NO」のセルを含む表（見積一覧表）の 1 列目の中で変数 **srcNo** の値を検索し、見つかったセルを変数 **srcRng** に代入する。
- ③ 見つからなかった場合、メッセージを表示して処理を終了する。
- ④ 見つかった場合、「データクリア」プロシージャを呼び出して入力欄をリセットする。
- ⑤ 変数 **srcRow** に見つかったセルの行番号を代入する。
- ⑥ 見積一覧表の 1 列目で変数 **srcNo** と同じ値をもつ数値の個数を調べ、変数 **cnt2** に代入します。
- ⑦ 見積一覧表の **srcRow** 行の A 列、B 列、D 列、C 列の値を、それぞれ見積書のセル F2（見積 NO）、F3（受注日）、A4（取引先名）、C5（取引先）に入力する。
- ⑧ 見積一覧表の E 列 **srcRow** 行目のセルから、H 列の「**srcRow+cnt2-1**」行目までのセル範囲の値を変数 **sData** に代入する。
- ⑨ B 列の 10 行目から E 列の「**10+cnt2-1**」行目までのセル範囲に変数 **sData** の値を入力し、明細行のデータを表示する。

● Memo ●

[見積一覧] シートのセル A3 に名前「見積 NO」が設定されています。

② Set srcRng = Range(" 見積 NO").CurrentRegion.
Columns(1).Find(what:=srcNo, lookat:=xlWhole)

⑤ srcRow=srcRng.Row

見積 NO	見積日	取引先 NO	取引先名	商品 NO	商品名	単価	数量	金額
1	2022/07/28	C1001	あかさ商事株式会社	DE1001	事務用デスク	¥15,000	2	¥30,000
2	2022/07/28	C1001	あかさ商事株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000
3	2022/07/28	C1005	AAA商会株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000
4	2022/07/28	C1003	株式会社マクスワイ	DE1001	事務用デスク	¥15,000	5	¥75,000
5	2022/07/28	C1003	株式会社マクスワイ	CH2001	事務用チェア	¥20,000	5	¥100,000
6	2022/07/28	C1003	株式		パーティション			
7	2022/07/28	C1002	トゥ		事務用チェア			
8	2022/07/28	C1002	トゥデイ工業株式会社	WA3001	ワゴン (3段)	¥10,000	1	¥10,000
9	2022/07/28	C1002	トゥデイ工業株式会社	WA3002	ワゴン (4段)	¥14,000	1	¥14,000
10	2022/07/29	C1004	株式会社天空産業	CH2001	事務用チェア	¥20,000	3	¥60,000
11	2022/07/29	C1004	株式会社天空産業	WA3001	ワゴン (3段)	¥10,000	3	¥30,000

Cells(srcRow, "E")

Cells(srcRow + cnt2 - 1, "H")

⑧ sData = .Range(.Cells(srcRow, "E"), .Cells(srcRow + cnt2 - 1, "H")).Value

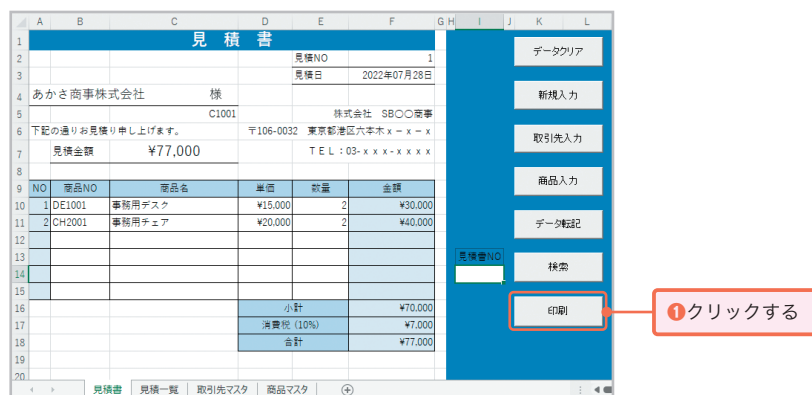
⑥ cnt2 = Application.WorksheetFunction.CountIf(Range (" 見積 NO").CurrentRegion.Columns(1), srcNo)

見 積 書					
			見積 NO	1	
			見積日	2022年07月28日	
あかさ商事株式会社 様					
C1001			株式会社	SB〇〇商事	
下記のとおり見積り申し上げます。			〒106-0032	東京都港区六本木 x - x - x	
見積金額			¥77,000	TEL : 03- x x x - x x x x	
NO	商品 NO	商品名	単価	数量	金額
1	DE1001	事務用デスク	¥15,000	2	¥30,000
2	CH2001	事務用チェア	¥20,000	2	¥40,000

⑨ Range(Cells(10, "B"), Cells(10 + cnt2 - 1, "E")).Value = sData

【印刷】ボタンで印刷設定をして見積書を印刷する

【印刷】ボタンをクリックしたら、見積書を印刷設定して印刷します。



【見積書印刷】プロシージャは、印刷範囲の設定をして印刷を実行します。印刷設定をするには、**PageSetup プロパティ**でページ設定情報を行います。

ここでは、**印刷範囲、用紙サイズ、用紙の向き、印刷倍率、ページ中央設定**をして、**プレビュー**を表示したのち印刷を実行します。

Sample 【見積書印刷】プロシージャ

```
Sub 見積書印刷 ()
    Dim ws As Worksheet
    Set ws = Worksheets("見積書")
    With ws.PageSetup
        .PrintArea = Range("A1:F18").Address ①
        .PaperSize = xlPaperA4 ②
        .Orientation = xlPortrait ③
        .Zoom = False
        .FitToPagesTall = 1 ④
        .FitToPagesWide = 1
        .CenterHorizontally = True ⑤
    End With
    ws.PrintOut Preview:=True ⑥
    Set ws = Nothing
End Sub
```

①【見積書】シートについて、印刷範囲をセルA1 からF18 に設定する。

②用紙サイズをA4 に設定する。

③印刷の向きを縦置きに設定する。

④用紙の縦、横方向に1 ページで収まるように設定する。

⑤水平方向に用紙の中央に配置する。

⑥印刷プレビューを表示する設定で印刷を実行する。