

## ■ 制作課題 1-1

使用ブック: Practice1\_1.xlsx

下記の解説を参考にして、集計を主な目的としたプログラムを作成しましょう。  
「売上(出庫)」「発注」「納品(入庫)」のボタンは図形で作成し、「発注伝票の発行」ボタンはフォームボタンで作成します。

緑で塗りつぶされた「B4」と「D4」はユーザーが入力します。

「売上(出庫)」「発注」「納品(入庫)」をクリックして作業を行います。

## ■ 完成例

	A	B	C	D	E	F	G	H	I	J
1										
2										
3		品番	品名	数量		発注伝票の発行				売上 (出庫)
4										
5										
6		在庫管理表								
7		品番	品名	現在の在庫	最低の在庫	発注残	発注先			発注
8		101	リンゴジュース	150	6		田沢食品			
9		102	コーヒー豆	50	5		コロラド			
10		103	フランスパン	36	4		abc Cook			
11		104	酵母パン	100	4		abc Cook			納品 (入庫)
12		105	アールグレー	98	2		コロラド			

## ■ HINT-考え方のフロー

## ■ オブジェクト変数を利用します。

処理の汎用性や簡略化、他のプロシージャへの流用などを考え、オブジェクト型変数を利用すると便利です。

## ▶ ワークシートをオブジェクト型変数に代入します。

```
Set ws = Worksheets("発注履歴")
```

## ▶ セルをオブジェクト型変数に代入します。

```
Set Target = Application.InputBox _  
("印刷する No を選択してください", Type:=8)
```

## ■ エラーが起こる可能性を回避する工夫をします。

## ▶ それぞれのプログラムの実行時には、利用するワークシートを検索し、ワークシートが見つからなければ処理を中止します。

\* 下記の変数の宣言とプロシージャを先に作成しておくとい良いでしょう。  
変数はプロシージャの宣言セクションに記述します。

## ■ 共通で利用する変数はプロシージャレベルの変数として宣言します。

- ▶ ワークシート作業で利用するワークシート型変数 (ws)
- ▶ フラグで利用する Boolean 型変数 (flag) (flag2)
- ▶ For 文で利用する変数 (i)

## ■ 共通で利用するプログラムは Call ステートメントを利用します。

## ▶ 発注伝票の印刷は Call ステートメントで呼び出します。

サブプロシージャ「発注伝票」を作成します。

コードはあとで作成します。

## Memo

プログラムの仕様がつかみづらい場合は、Practice1\_1ans.xlsm を実行して確認しましょう。

## Point

## ▶ フォームボタンの作成

- ① [開発] タブの [コントロール] グループから [挿入] を選択します。
- ② ボタン (フォームコントロール) を選択します。
- ③ ワークシート上でドラッグしてボタンを作成します。
- ④ [マクロの登録] ダイアログボックスで登録するマクロを選択します。
- 図形へのマクロの登録  
図形の上で右クリックして、[マクロの登録] を選択し、[マクロの登録] ダイアログボックスで登録するマクロを選択します。

## Memo

「発注伝票の発行」ボタンをクリックしたときの処理で使用します。

## Memo

変数名はテキスト通りでなくてもかまいません。

## Point

「発注伝票」プロシージャのコードはまだ書かなくて良いです。

## ■ 「売上(出庫)」ボタンに登録するプログラムの作成

## ■ 処理の流れ

- ① 使用するシートの有無を確認します。
- ② 「B4」には「B8 から B12」の品番に該当する品番を表示します。
- ③ 「D4」には売上数量を入力して、「売上(出庫)」をクリックします。
- ④ 「D4」に表示された数値が品番が一致する商品の「現在の在庫を上回る」「0 以下」「数値以外」の場合は正しい数値を入力し直すための InputBox を表示します。
- ⑤ InputBox が「在庫数以下」「0 より大きい」「数値」になるまで入力を促します。
- ⑥ 数量が在庫数からマイナスされ、「B4 から D4」の値が「売上履歴」シートに転記されます。

## ■ 作業シート(「入力窓口」シートと「売上履歴」シート)の有無によって処理を分岐します。

- ① シート名を調べるための繰り返し処理を行い、「入力窓口」シートがあれば、flag を「True」にして、繰り返し処理を終了します。シートがなければメッセージボックスを表示して、プロシーダを終了します。
- ② flag を「False」に戻します。
- ③ 「売上履歴」シートがあれば、flag を「True」にして、繰り返し処理を終了します。シートがなければメッセージボックスを表示して、プロシーダを終了します。
- ④ flag を「False」に戻します。

## ■ 「B4」に入力された品番と、「B8」から「B12」までの品番を照合して、一致する品番がある場合と、ない場合で処理を分岐します。

Step 1 ▶ 「B4」と同じ値が「B8」から「B12」にある場合の処理を行います。

- ① 同じ値があれば、処理を行いプロシーダを抜けます。

```
For i = 8 To .Cells (Rows.Count, 2) .End (xlUp) .Row
  If .Range ("B4") = .Cells (i, 2) Then
    ' 処理 Step2
    Exit For ' 同じ値があれば、Step2 の処理後 For 文を抜ける
  End If
Next
```

Step 2 ▶ 「B4」と同じ値があった場合の処理を行います。

- ① 「C4」に対応する品名を表示します。
- ② flag を「True」とします。
- ③ 「D4」に入力された数値が下記の条件に当てはまるかを調べ、「D4」の値が1つでも下記に当てはまる場合は「適切な値ではない」として、InputBox を表示し、正しい値の入力を促します。

「D8 から D12」より大きい(在庫を上回る)、または 0 以下、または数値以外

品番	品名	数量	発注伝票の発行		
101	リンゴジュース	160			
在庫管理表					
品番	品名	現在の在庫	最低の在庫	発注残	発注先
101	リンゴジュース	150	6		田沢食品

正しい数値が入力されていないので InputBox を表示します

## Memo

## ▶ Exit For

For文の繰り返し処理を中断します。繰り返し処理を抜け、余分な処理を省略します。ここでは、同名のシート名が見つかったら処理を終了しています。

## ▶ Exit Sub

Sub プロシーダを終了します。Exit Subがないと、エラー時の処理が実行されます。通常はある条件の時、それ以降の処理をせずにプロシーダを終了する際に使います。

## 記述例

```
If 条件 Then
  Exit Sub
End If
```

## ▶ flag

シートがない場合とある場合の処理を Boolean型の変数で分岐します。既定値は「False」ですので、「True」を代入した場合、別の処理で利用する時は、既定値の「False」に戻します。

## 記述例

```
For Each ws In Worksheets
  If ws.Name="入力窓口" _
  Then
    flag = True
  Exit For
  End If
Next
If flag = False Then
  MsgBox "処理を中止。"
  Exit Sub
End If
flag = False
```

## Point

## ▶ 「D8からD12」以上の確認

For文の中に記述されていますので下記とします。

And条件で記述する方法もあります。

## 記述例

```
If .Range ("D4") > .Cells (i,4) _
  Or IsNumeric _
  (.Range ("D4"))=False Or...
```

## Point

## ▶ IsNumeric関数

構文: IsNumeric (expression)

引数 expression が数値として認識できる場合は「True」、そうでない場合は「False」を返します。