第4章

プログラミング

学習目標

- (1) プログラミングを理解する.
- (2) プログラムの作成方法を習得する.
- (3) セルや変数への代入文を習得する.
- (4) 反復処理を用いたプログラミングを習得する.
- (5) 選択処理を用いたプログラミングを習得する.

本章は、専修大学経営学部の大曽根匡の著作(2014年8月18日)である。

執筆にあたって, 専修大学経営学部兼任講師の新保好美氏と廣澤敏夫氏にご助言や題材の提供を受けた. この場を借りて, 感謝の意を表したい.

1 プログラミングとは

スマートフォーンにはいろいろなアプリがあり、みなさんはそれを利用していることでしょう。その「アプリ」とは、アプリケーション・プログラム(応用プログラム)の略であり、プログラムのことです。コンピュータの中のソフトウェアもプログラムであり、誰かがそのプログラムを作成しています。プログラムを作成することをプログラミングといいます。本章は、みなさんにプログラミングを体験してもらうことを目的としています。

プログラムは Excel 上でも作成できます. Excel 上でプログラムを作成できるようになると, Excel を使って行っていたいろいろな定型的な作業を, 自動化することもできます. また, 自分だけのプログラムも作成できます.

プログラムは、コンピュータへ指示する命令の集まりです。そこでは、処理の順序、あるいは、手順が大切となります。処理の手順が誤っていると、誤った答えが出てきてしまいます。したがって、処理手順を考えることから始めます。その処理手順のことをアルゴリズムといいます。アルゴリズムを考えるときは、通常、日本語や英語や中国語など、自然言語を使って考えます。アルゴリズムが固まったら、その手順を、あるプログラミング言語を使って記述します。その書き物のことをプログラムといいます。プログラミング言語は人工的な言語であり、コンピュータが理解でき、実行できる言語のことです。C、Basic、Cobol、Fortran、Pascal などいろいろな言語が存在しています。本章では、Excel 上でプログラムを作成できる言語である Visual Basic を使用して、プログラムを作成します。Visual Basic for Application を略して VBA ということもあります。その言語には文法が存在し、文法に誤りがあるとコンパイルエラーとなってしまいます。プログラムにエラーはつきものです。エラーにめげず、プログラムを作成していきましょう。みなさんには、プログラムが通って、正しい実行結果が出たときの爽快感を味わってほしいと思います。それでは、プログラミングの体験に出発しましょう。

2 題材

2.1 BMI の計算

本章では、各人の BMI (Body Math Index) を計算することを題材として取り上げます. BMI は肥満度を測る簡易な指標として広く用いられています. その計算式は

$$BMI = \frac{\text{体重 (kg)}}{\text{身長 (m)} \times \text{身長 (m)}}$$

です.

その BMI の値により、表 1 のように肥満度が判定されます(日本肥満学会による).

表1 BMIによる肥満度の分類

BMI	判定結果			
18.5 未満	低体重			
18.5 以上 25 未満	普通体重			
25 以上 30 未満	肥満(1度)			
30 以上 35 未満	肥満 (2 度)			
35 以上 40 未満	肥満 (3 度)			
40以上	肥満 (4 度)			

2.2 Excel のデータ

この章で使用する Excel のデータ (BMI.xlsx) は、図1のような構成になっています.

A列:番号

B列:氏名

C 列:性別

D 列:身長 (m)

E列: 体重 (kg)

4行目はフィールド名が表記され、5行目から24行目まで20人分の仮想のデータが入っています.

	Α		В	С	D	Е	F	G
1								
2								
3								
4	番号		氏名	性別	身長(m)	体重(kg)		
5	1	木村	哲久	男性	1.78	112		
6	2	西野	徹	男性	1.67	80		
7	3	木村	功一	男性	1.70	55		

図1 データの形式

3 準備

3.1 開発タブの表示

Excel 上でプログラムを開発するためには、開発タブを Excel に表示させる必要があります.

■開発タブの表示

- (1) ファイル タブ → オプション
- (2) 図 2 のような Excel のオプション画面が表示されます.
- (3) リボンのユーザ設定 → メインタブの 開発 に「チェック」する

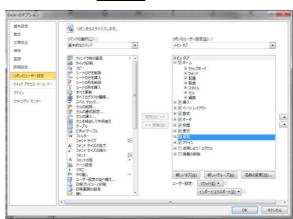


図2 Excel のオプション画面における開発タブの設定

3.2 開発タブの構成と機能

開発タブのツールボタンの構成を図3に示します. 主なツールボタンは下記のとおりです.

- (1) Visual Basic: Visual Basic によるプログラムの開発を行うための VBA 画面(図 4)を表示します.
- (2) 挿入: ユーザインタフェースの部品であるツールボックスを表示します.
- (3) デザイン: デザインモードの切り替えをします.
- (4) プロパティ:プロパティウィンドウを表示します.
- (5) コードの表示:プログラムのコードを記述するためのコードエディタを表示します.



図3 開発タブのツールボタンの構成

3.2 VBA のツールバー

VBA 画面とそのツールバーを図4と図5に示します. 主なツールボタンは下記のとおりです.

- (1) Excel 表示ボタン: Excel を表示します.
- (2) リセットボタン:プログラムのリセットをします.

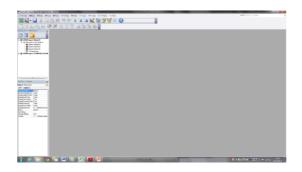


図4 VBA 画面



図5 VBA画面のツールバー

4 ユーザインタフェースの設計

4.1 コマンドボタンの作成

ユーザが BMI 計算プログラムを実行させるためのコマンドボタンを Excel 上に作成します. ユーザが操作する ために用いるユーザインタフェースの部品のことをコントロールといいます. ここでは,「計算」ボタンを作成してみましょう.

■コマンドボタンの作成

- (2) Excel 上の I 列 4 行目から 5 行目のセルの大きさにドラッグし, 図 6 のようなコマンドボタンを作成します.

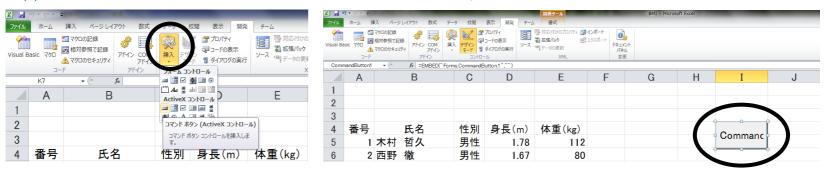


図6 コマンドボタンの作成

4.2 コマンドボタンのプロパティの設定

作成したコマンドボタンにプロパティを設定します. 図7のようにオブジェクト名と Caption を設定します.

オブジェクト名: cmd 計算

Caption:計算

オブジェクト名の最初の「cmd」は、コントロールの種類がコマンドボタンだということがプログラムの中ですぐにわかるようにするためにつけます. Caption はコマンドボタンに表示される文字列です.

■コマンドボタンのプロパティの設定

- (1) 開発 タブ→ プロパティ ボタン
- (2) プロパティウィンドウが表示されるので、オブジェクト名と Caption を設定します.

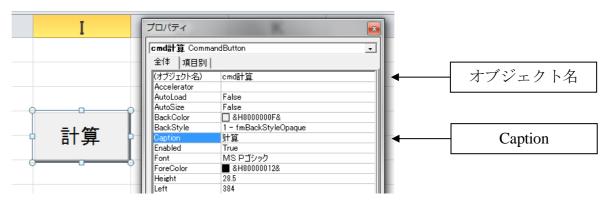


図7 プロパティの設定

5 プログラムの作成

5.1 プログラムの入力と実行

Excel 上の特定のセルに文字列を表示させるプログラムを作成してみよう. i 行j 列のセルへ文字列を代入する命令は、下記のように書きます.

■セルへの文字列の代入

Cells(i, j) = "文字列"

ここで、文字列の前後に"(ダブルクォーテーション)をつけることに注意してください。また、B列2行目のセルに文字列を表示させたいときは、B列は2列目なので、

と書きます.

それでは、次の例題を具体的にやってみましょう.

◇例題 1

B2(2行2列)のセルに「BMIの計算」という文字列を表示させるプログラムを作成しなさい.

■プログラムの入力

- (1) 開発タブがデザインモードがオンになっていることを確認してから、先ほど作成した「計算ボタン」をダブルクリックします.
- (2) すると、Visual Basic 画面のコードエディタが表示され、図 8 のように、 Private Sub cmd 計算 Click()

Fnd Sub

と cmd 計算ボタンがクリックされたときに起動するプログラムのひな形が表示されます.

(3) Private Sub の次の行に

Cells(2, 2) = "BMIの計算"

と入力します.この際,図9のようにTab キーを使用して,4 文字分字下げ(インデント)をつけるようにしましょう.字下げすることにより,プログラムが見やすくなり,文法上のエラーも発見しやすくなります.

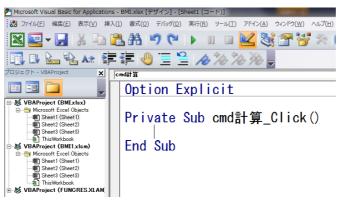


図8 コードエディタの表示

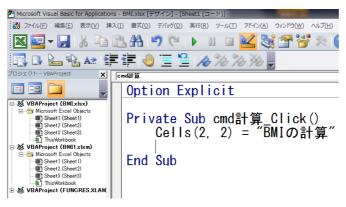


図9 プログラムの入力

■プログラムの実行

- (1) Excel 表示 ボタン (図 5 参照) により, Excel に切り替えます.
- (2) 開発 タブ→ デザイン ボタンにより, デザインモードをオフにします.
- (3) 「計算ボタン」をクリックします.
- (4) 図 10 のような実行結果が得られます. B2 のセルに「BMI の計算」と表示されていることを確認してください.

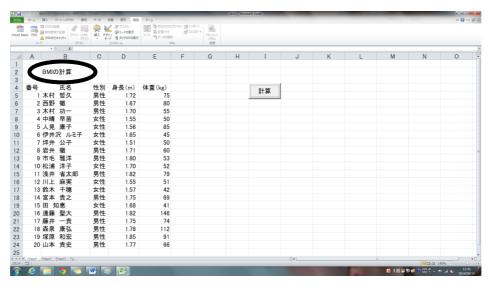


図10 例題1の実行結果

次の問題をやってみましょう.

◇問題1

図 11 のように、F4 のセルに「BMI」、G4 のセルに「判定結果」という文字列を表示させるプログラムを作成しなさい。



図11 問題1の実行結果

5.2 プログラムの保存

Excel 上で作成したプログラムは、「マクロ有効ブック」として保存します. 拡張子は「.xlsm」となります. 保存の方法は以下のとおりです.

■プログラムの保存

- (1) ファイル タブ→ 名前を付けて保存
- (2) 図 12 のように「名前を付けて保存」のウィンドウが表示される.
- (3) ファイルの種類として、「マクロ有効ブック」を選択し、「BMI」というファイル名のまま保存する.

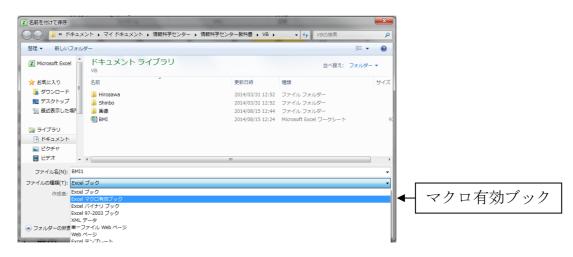


図 12 マクロ有効ブックとして保存

5.3 プログラムのエラーとその修正方法

プログラムに誤りがあるとエラーが発生します. プログラミングにはエラーはつきものです. エラーの種類には、次の3つのものがあります.

- ① コンパイルエラー:プログラムに文法上の誤りがあると発生します.
- ② 実行時エラー:プログラムを実行させたときに発生するエラーです.
- ③ 論理エラー:実行結果に誤りがあるエラーです.

ここでは、コンパイルエラーの修正方法について説明します. 問題 1 において、「Cells」を「Cels」に変えて実行させてみましょう. すると、「Cels」という用語は Visual Basic に存在しないので、実行させると図 13 のようにコンパイルエラーが発生します. このとき、青く反転されているところが、エラーが発生した箇所です. 下記の操作によりエラーを修正します.

- ■エラーの修正とプログラムの実行
 - (1) コンパイルエラーのメッセージの内容を読んでから, OK ボタンをクリックします.
 - (2) プログラムを修正します.
 - (3) リセット ボタンをクリックし、図 14 の黄色い表示を消します.
 - (4) Excel 表示 ボタンをクリックし Excel の画面に戻し,「計算ボタン」をクリックして, 修正したプログラムを実行させます.

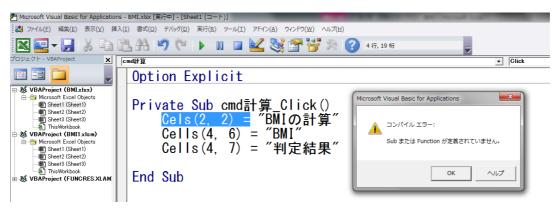


図 13 コンパイルエラーの発生

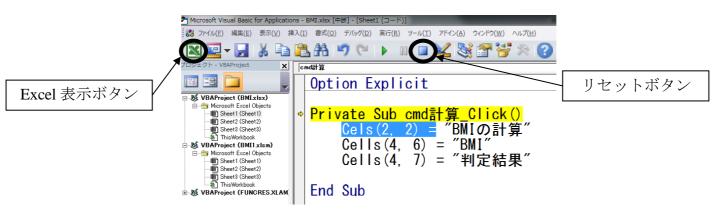


図 14 リセットボタン

5.4 変数と代入文

プログラムにおいて、計算途中のデータを一時的に記憶しておきたい場合があります。そのデータの保存場所のことを変数といいます。変数には適当な名前をつけ、プログラムを記述する際に用います。変数を使用することの利点は、プログラムがわかりやすくなることと、プログラムの中で何度でも利用できることです。

変数のイメージとして、図 15 のように、名前の付いた箱を想像してください。データの種類に応じて、その箱にも種類があります。それをデータ型といいます。例えば、整数を記憶する箱、実数を記憶する箱、文字列を記憶する箱といった具合です。図 15 では、変数名は「体重」、データ型は「整数」です。

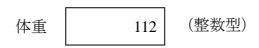


図 15 変数のイメージ

プログラムにおいて変数を使用するときは、最初に変数の宣言を行います.

■変数の宣言

Dim 変数名 As データ型

ここで、データ型は、表 2 の用語を使用します.変数の宣言は、プログラムの先頭部分に記述します. それにより、プログラムで使用される変数が一目でわかるようになるからです.

表2 データ型

データ型	Visual Basic
整数型	Integer
実数型	Single
文字列型	String

変数へ数値や文字列、セルの値、計算式を代入する構文は下記のとおりです。

■変数への代入文

変数名 = 数値

変数名 = "文字列" (文字列の前後は "で囲む)

変数名 = セル名 (セル名は、Cells(i, j) のように記述する)

変数名 = 計算式 (使用できる演算子は Excel と同じで, +, -, *, /, ^などである)

これを使用して、例題2をやってみましょう.

◇例題 2

5行目の人のBMIを求め、それをF5(5行6列)のセルに表示させるプログラムを作成しなさい。ただし、変数として、「体重」と「身長」と「BMI」を用い、「体重」は整数型、「身長」と「BMI」は実数型として宣言すること。

例題 2 のプログラムの説明は以下のとおりです. そのプログラムを図16に, 実行結果を図17に示します. ◇例題 2 のプログラム

■変数の宣言

Dim 体重 As Integer Dim 身長 As Single Dim BMI As Single

■変数への代入

身長 = Cells(5, 4) 体重 = Cells(5, 5)

■BMI の計算

BMI = 体重 / (身長 * 身長)

■セルへの代入

Cells(5, 6) = BMI

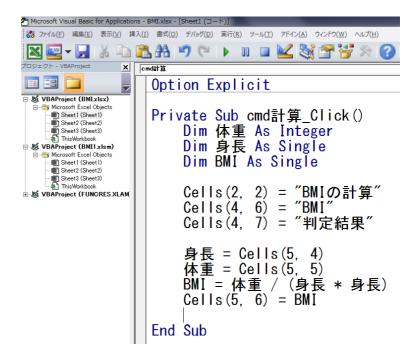


図 16 例題 2 のコードエディタ



図17 例題2の実行結果

5.5 反復処理を実現する文法: For 文

同じような処理を何回か繰り返す反復処理に対しては、For 文を使用することが多いです。For 文の構文は以下のとおりです。

■For 文の構文

For カウンタ変数 = 初期値 To 終了値 [文]

Next カウンタ変数

For 文では、カウンタ変数が初期値から始まり、[文]を実行し、Next の文でカウンタ変数が1つカウントアップされ、For の文に戻ります. そして、カウンタ変数が終了値になるまで[文]が繰り返し実行されます.

下記の例題3を、For 文を使用して作成してみましょう.

◇例題 3

5行目から24行目までのBMIを求め、同じ行のF列(6列)のセルに出力するプログラムを作成しなさい.

行数を表すカウンタ変数 N を導入し、例題 2 の 5 行目に対応する Cells の第 1 パラメタを N 行目とし、N を 5 行目から 24 行目まで反復処理すると考えましょう。すなわち、例題 3 のプログラムは下記のようになります。

◇例題3のプログラム

■カウンタ変数 N の宣言

Dim N As Integer

■5 行目から 24 行目までの反復処理

その実行結果を図18に示します.

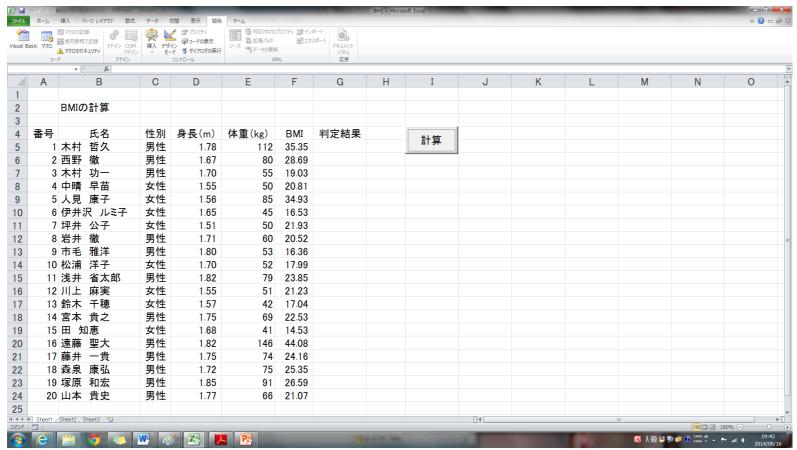


図18 例題3の実行結果

5.6 選択処理を実現する文法: If 文

条件が成立するかしないかで実行する処理を変える処理を選択処理といいます. 選択処理には If 文が用いられます. 選択処理は

- ① 1択処理:処理Aを実行するか,実行しないか
- ② 2択処理:処理Aを選択するか,処理Bを選択するか
- ③ 3択以上の処理:処理 A を選択するか、処理 B を選択するか、処理 C を選択するか、・・・の 3 種類に分類できます.

5.6.1 1択処理: If ~ Then ~ EndIf 形

1択処理には、If \sim Then \sim EndIf 形の If 文を用います、その構文は以下のとおりです、

■If ~ Then ~ End If 形の構文

If 条件式 Then

[条件式が成立しているときに実行する文]

EndIf

条件式には、Cells(4,3) = "男性" や BMI >= 25 など、セルや変数、数値、文字列などの比較条件を記述します。 比較条件に使用する比較子を表 3 に示します。

表 3 比較子

比較子	意味
>	より大きい
>=	以上
<	より小さい
<=	以下
=	等しい
\Diamond	等しくない

If \sim Then \sim EndIf 形の If 文を用いて、例題 4 のプログラムを作成してみましょう.

◇例題 4

5行目の人のBMIが30以上のとき、BMIのセル(F列)を黄色で塗りつぶすプログラムを作成しなさい.

例題 3 により, 5 行目の人の BMI は Cells(5,6)に入っているので, そのセルと 30 を比較して, 30 以上だったらそのセルを黄色で塗りつぶします. セルの内部の色の塗りつぶしは, 下記の構文を用います.

■セルの内部の色の塗りつぶし

Cells(i, j).Interior.Color = 色

(色: vbBlack, vbRed, vbGreen, vbBlue, vbMagenta, vbYellow, vbCyan, vbWhite)

上記の構文で、i行j列のセルの内部が指定された色で塗りつぶすことができます。もっとたくさんの色を使用したいときは、RGB 関数を使うとよいですが、ここでは省略します。

例題4のプログラムは下記のようになります. その実行結果を図19に示します.

◇例題4のプログラム

If Cells (5. 6) >= 30 then

Cells (5, 6). Interior. Color = vbYellow

EndIf

	Α	В	С	D	Е	F	G	Н	I	J
1										
2		BMIの計算								
3										
4	番号	氏名	性別	身長(m)	体重(kg)	BMI	分類		計算	
5	1	木村 哲久	男性	1.78	112	35.35			可异	
6	2	西野 徹	男性	1.67	80	28.69				
7	3	木村 功一	男性	1.70	55	19.03				

図19 例題4の実行結果

次に、For 文を使用して、下記の例題のプログラムを作成してみましょう.

◇例題 5

5 行目から 24 行目までの人について, BMI が 30 以上のとき, BMI のセル (F列) を黄色で塗りつぶすプログラムを作成しなさい.

この例題は、例題 4 の処理を、For 文で反復処理させればよいと考えることができます。すなわち、例題 4 の 5 行目のセルに対応する Cells の第 1 パラメタを N 行目とし、N を 5 行目から 24 行目まで繰り返すように変更します。 具体的には、下記のようなプログラムとなります。 その実行結果を図 20 に示します。

◇例題5のプログラム

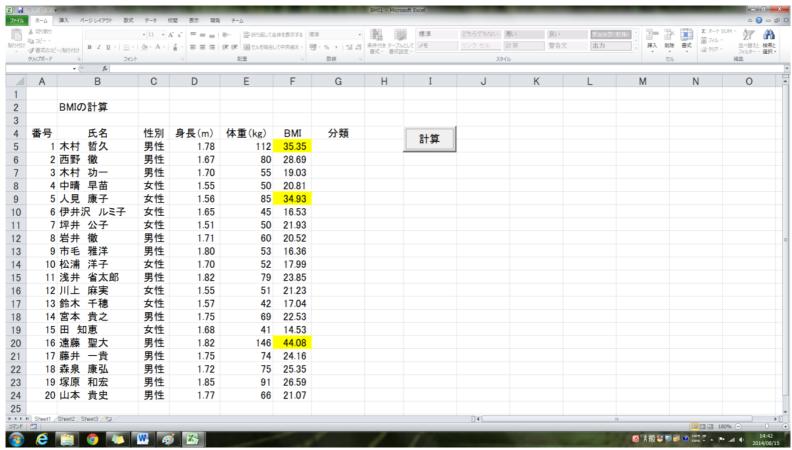


図20 例題5の実行結果

5.6.2 2択処理: If ~ Then ~ Else 形

2択処理には、If \sim Then \sim Else 形の If 文を用います. その構文は以下のとおりです.

■If ~ Then ~ Else 形の構文

If 条件式 Then

[条件式が成立しているときに実行する文]

Else

[条件式が成立していないときに実行する文]

End If

2択処理の構文を使用して、下記の問題のプログラムを作成してみましょう.

◇問題 2

5行目から24行目までの人について、性別が男性のときはその行のC列のセルを青色で、女性のときは赤色で 塗りつぶすプログラムを作成しなさい(その実行結果は図21).

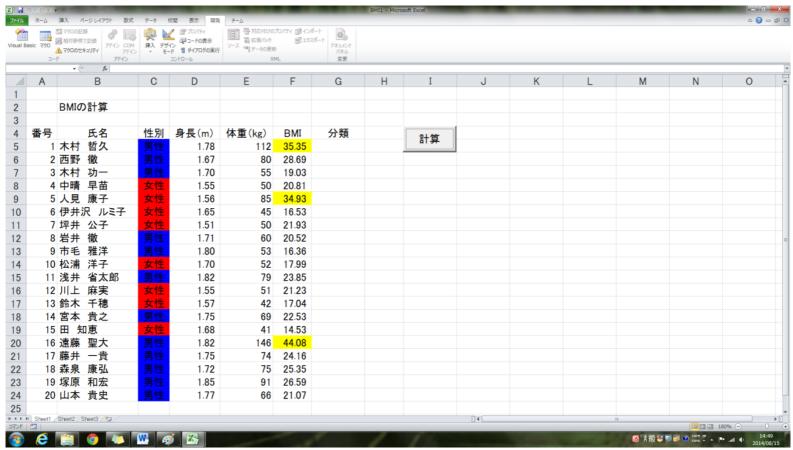


図21 問題2の実行結果

5.6.3 3択以上の処理: If ~ Then ~ ElseIf 形

3択以上の処理には、If \sim Then \sim ElseIf 形の If 文を使います. その構文は以下のとおりです.

■If ~ Then ~ ElseIf 形の構文

If 条件式 1 Then

[条件式1が成立しているときに実行する文]

Else If 条件式 2 Then

[条件式2が成立しているときに実行する文]

Else

[上記のどの条件も成立していないときに実行する文]

End If

この構文を使用して、下記の例題を考えましょう.

◇例題 6

5行目から24行目までの人について、BMIが18.5未満のとき判定結果を「低体重」、BMIが18.5以上25未満のとき判定結果を「普通体重」、BMIが25以上のとき判定結果を「肥満」とし、それをその行のG列のセルに表示するプログラムを作成しなさい。

この例題は、「低体重」か「標準体重」か「肥満」かを判別する 3 択処理の問題です。N 行(N = $5\sim24$)6 列目のセル Cells(N, 6)を参照し、その値で判定結果を判断すればよいでしょう。ここでは、「判定結果」という文字列型の変数を用意してプログラムを作成してみましょう。その実行結果を図 22 に示します。

◇例題6のプログラム

- ■「判定結果」という文字列型の変数の宣言 Dim 判定結果 As String
- ■3 択のプログラム

```
For N = 5 To 24

If Cells(N, 6) < 18.5 Then

判定結果="低体重"

Else If Cells(N, 6) < 25 Then

判定結果="普通体重"

Else

判定結果="肥満"

End If

Cells(N, 7) = 判定結果

Next N
```

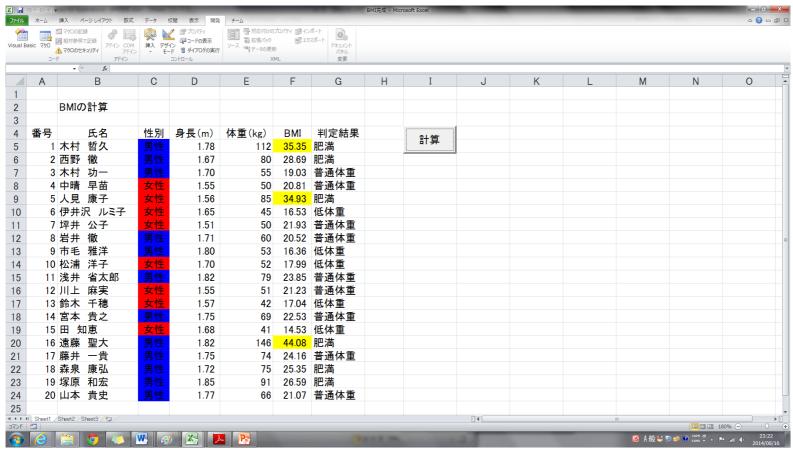


図22 例題6の実行結果

上記の構文を使用して、下記の6択の問題のプログラムを作成してみましょう. その実行結果を図23に示します.

◇問題3

5 行目から 24 行目までの人について,表 1 に基づいて判定結果を出力しなさい. すなわち,BMI が 18.5 未満のとき「低体重」,BMI が 18.5 以上 25 未満のとき「普通体重」,BMI が 25 以上 30 未満のとき「肥満 (1 度)」,BMI が 30 以上 35 未満のとき「肥満 (2 度)」,BMI が 35 以上 40 未満のとき「肥満 (3 度)」,BMI が 40 以上のとき「肥満 (4 度)」とし、それをその行の G 列のセルに表示するプログラムを作成しなさい.

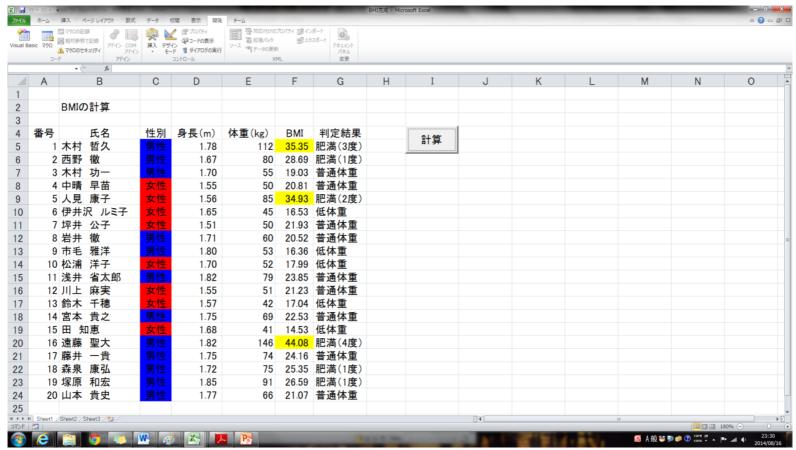


図23 問題3の実行結果

7 BMI カードの作成

7.1 カードボタンの作成

Excel の Sheet2 に BMI カードのひな形があります. このひな形のカードを利用して, Sheet1 の各人の BMI を報せる BMI カードを生成してみましょう. まず, Sheet1 の「計算」ボタンの下の I7 と I8 のセルの場所に「カード」というコマンドボタンを作成しましょう. ここで,

オブジェクト名: cmd カード

Caption:カード

とします. 図 24 に作成したコマンドボタンを示します.

	Α	В	С	D	Е	F	G	Н	I	J
1										
2		BMIの計算								
3										
4	番号	氏名	性別	身長(m)	体重(kg)	BMI	判定結果		計算	
5	1	木村 哲久	男性	1.78	112	35.35	肥満(3度)		il 另	
6	2	西野 徹	男性	1.67	80	28.69	肥満(1度)			
7	3	木村 功一	男性	1.70	55	19.03	普通体重		カード	
8	4	中晴 早苗	女性	1.55	50	20.81	普通体重		/J _L	
9	5	人見 康子	女性	1.56	85	34.93	肥満(2度)			

図 24 カードボタンの作成

7.2 シートのコピー

次に、作成した「カード」ボタンをクリックすると、Sheet2のコピーを現在のSheet2の後ろに生成するプログラムを作成してみましょう。シートのコピーを挿入する構文は以下のとおりです。

■シートのコピー

Sheets(シート名).Copy After := Sheets(シート名)

ここで、シート名は前後に"をつけてください。また、Afterは「後ろに生成する」ことを指示しています。

◇例題 7

Sheet2 の後ろにそのシートのコピーを生成するプログラムを作成しなさい.

◇例題7のプログラム

Sheets ("Sheet2"). Copy After:=Sheets ("Sheet2")

ここで、生成されたシート名は Sheet2(2)となります. その実行結果とプログラムを図 25 に示します.

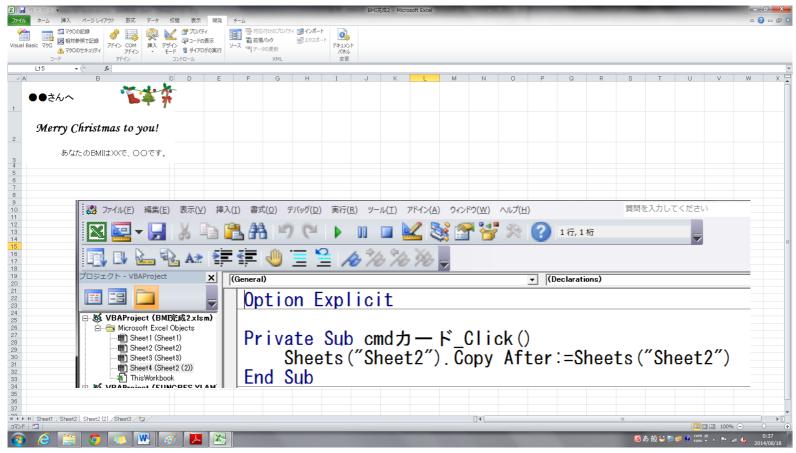


図 25 例題 7 の実行結果とプログラム

7.3 宛先とBMI情報の表示

◇例題 8

Sheet2 をコピーしたシート Sheet2(2)の BMI カードに Sheet1 の 5 行目の人の宛先と BMI 情報を表示させるプログラムを作成しなさい.

まず、「宛先」と「BMI情報」という名前の文字列型の変数を用意します.

■変数宣言

Dim 宛先 As String Dim BMI 情報 As String

指定されたシートの特定のセルからデータを参照し、それを変数に代入する構文は以下のとおりです.

■指定されたシートのi行j列のセルの代入文

変数名 = Sheets(シート名).Cells(i, j)

また、文字列1の後ろに文字列2を連結させる演算は、演算子&を用い、以下のように記述します。

■文字列の連結

文字列1& 文字列2

例えば、文字列 1 が "ABC"、文字列 2 が "DEF"の場合、文字列 1 & 文字列 2 は、"ABCDEF" となります。 さて、この例題では、Sheet1 の 5 行目の人の氏名を変数「宛先」に代入し、さらに、その後ろに "さんへ" という文字列を連結させたいので、変数「宛先」への代入文は

宛先 = Sheets ("Sheet1"). Cells (5, 2) & "さんへ"

となります.この代入文により、変数「宛先」には、"木村 哲久さんへ"が代入されます.

また、変数「BMI情報」への代入文を

BMI 情報 = "あなたの BMI は" & Sheets ("Sheet1"). Cells (5, 6) & "です。" とすることにより、変数「BMI 情報」には、

" あなたの BMI は 35.34907……です。"

が代入されます. ところが,これでは BMI の数値の桁数が多すぎるので,左から 5 文字までの文字列だけを抜き出すことにします. 文字列の左端から指定された文字数を抜き出すためには,下記の関数を使用します.

■文字列の左端から指定した文字数分の文字列を抜き出す関数 Left Left(文字列、文字数)

この Left 関数を使用して、「BMI 情報」への代入文を

BMI 情報 = "あなたの BMI は" & Left (Sheets ("Sheet1"). Cells (5, 6), 5) & "です。" とすると,「BMI 情報」は,

" あなたの BMI は 35.34 です。"

となり, 見やすくなります.

最後に、「宛先」と「BMI情報」を、コピーしたシートの特定のセルに書き込むことを行います.「宛先」はB1のセルに、「BMI情報」はB3のセルに書き込めばよいので、

ActiveSheet. Cells(1, 2) = 宛先

ActiveSheet.Cells(3.2) = BMI 情報

とします. ここで、ActiveSheet は現在アクティブになっているシート、すなわち、コピーされたシートを指しています.

以上をまとめると、例題 8 のプログラムは下記のようになります. なお、プログラムを実行する際は、例題 7 で生成したコピーシート Sheet2(2)を削除してから実行するようにして下さい. その実行結果を図 26 に示します.

◇例題8のプログラム

■変数宣言

Dim 宛先 As String Dim BMI 情報 As String

■「宛先」と「BMI 情報」への代入

宛先 = Sheets ("Sheet1"). Cells (5, 2) & "さんへ"

BMI 情報 = "あなたの BMI は" & Left (Sheets ("Sheet1"). Cells (5, 6), 5) & "です。"

■コピーしたシートへの「宛先」と「BMI情報」の書き出し

ActiveSheet. Cells (1, 2) = 宛先

ActiveSheet. Cells(3, 2) = BMI 情報



図 26 例題 8 の実行結果

7.4 複数の BMI カードの生成

◇例類 9

Sheet1の5行目から7行目までの3人分に対して、BMIカードを生成するプログラムを作成しなさい.

この例題は、For 文を使用して、例題 7 と例題 8 を 5 行目から 7 行目まで 3 回繰り返すとよいと考えられます. そこで、行数を表す整数型の変数 N を用意し、以下のようなプログラムに変更します.ここで、「宛先」と「BMI情報」の代入文において、Cells の第 1 パラメタを N としていることに注意してください.N 行目の人の情報を代入したいからです.実行結果を図 27 に示します.Sheet2 のコピーが 3 つ生成されていることに注意してください. For 文において 7 を 24 に変えるだけで、20 名分まとめて BMI カードを生成するようにできます.

◇例題9のプログラム

■変数宣言

Dim N As Integer

■反復処理による繰り返し

For N = 5 To 7

Sheets("Sheet2"). Copy After:=Sheets("Sheet2")

宛先 = Sheets("Sheet1"). Cells(N, 2) & "さんへ"

BMI 情報 = "あなたの BMI は" & Left (Sheets ("Sheet1"). Cells (N, 6), 5) & "です。"

ActiveSheet. Cells(1, 2) = 宛先

ActiveSheet. Cells (3, 2) = BMI 情報

Next N

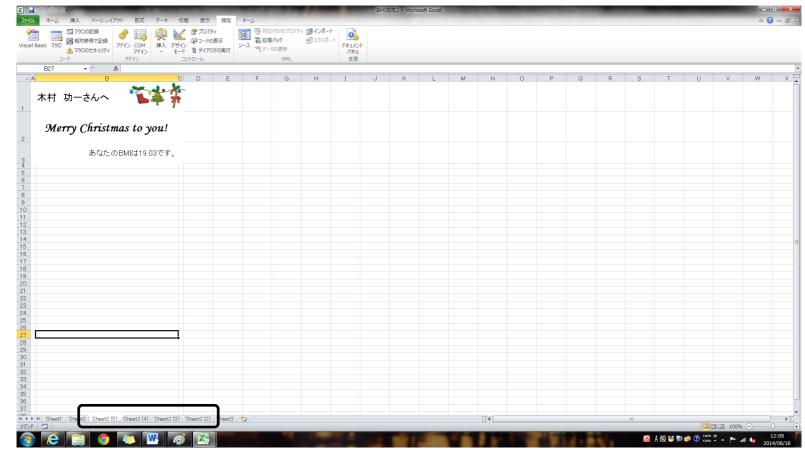


図27 例題9の実行結果