

# 第3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.
- (6) マクロの記録を取る方法を習得する.

本章は、専修大学経営学部の大曾根匡の著作（2014年8月18日作成，2015年8月10日改訂）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

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

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。

	A	B	C	D	E	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

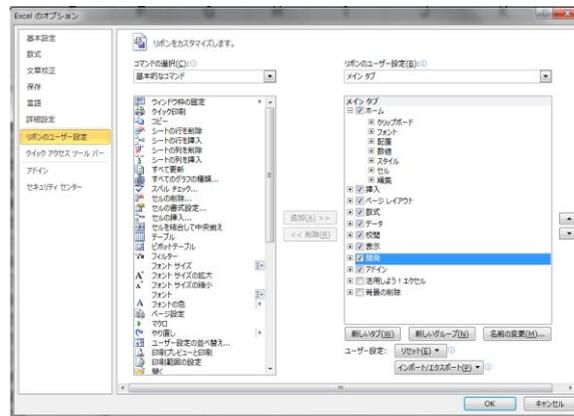


図 2 Excel のオプション画面における開発タブの設定

## 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（オン／オフ）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

### 3.3 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。

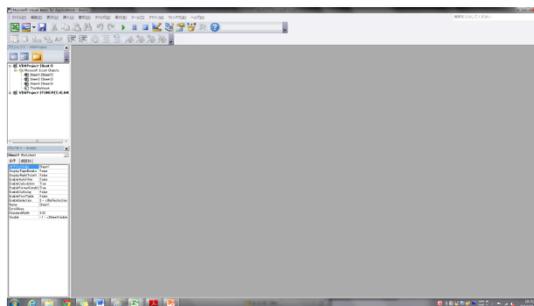


図 4 VBA 画面

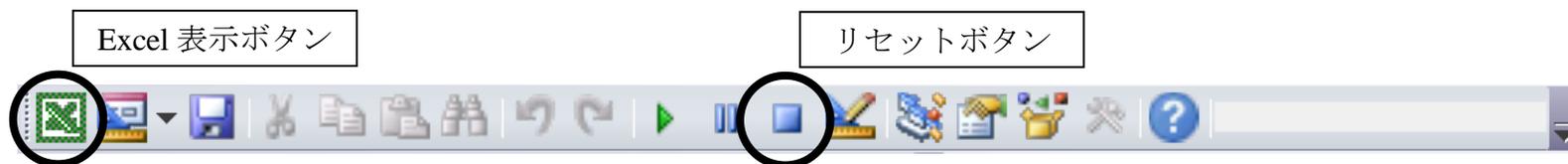


図 5 VBA 画面のツールバー

## 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

(1) **開発** タブ → **挿入** ボタン → **コマンドボタン(ActiveX コントロール)**

(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。

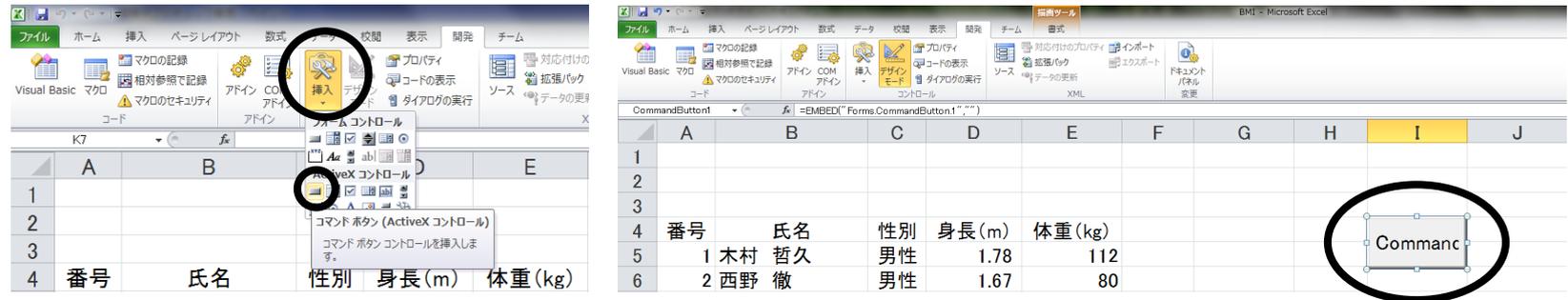


図 6 コマンドボタンの作成

## 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。

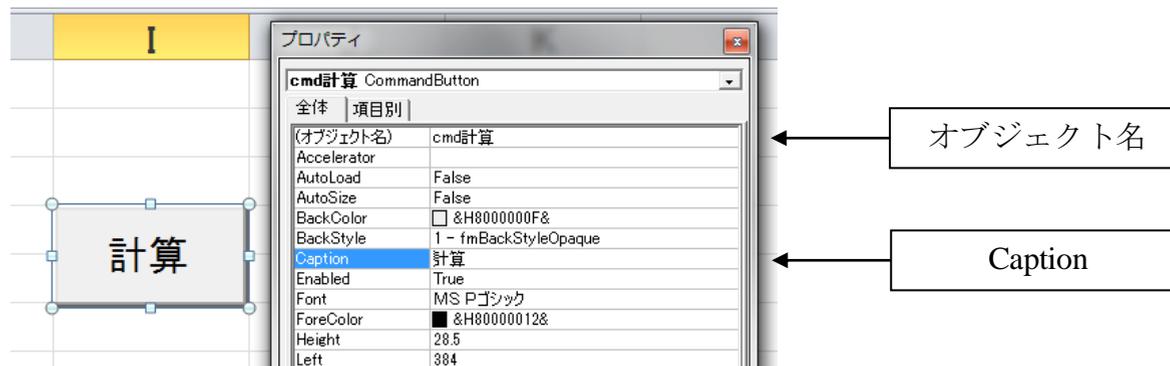


図7 プロパティの設定

## 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

$$\text{Cells}(i, j) = \text{“文字列”}$$

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、

$$\text{Cells}(2, 2) = \text{“文字列”}$$

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) 開発タブの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()
```

```
End 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）をす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。

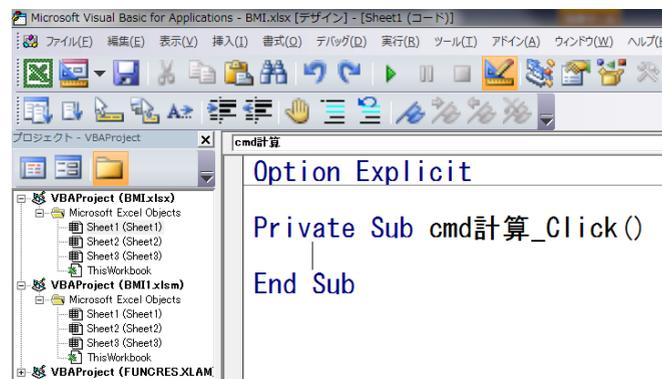


図 8 コードエディタの表示

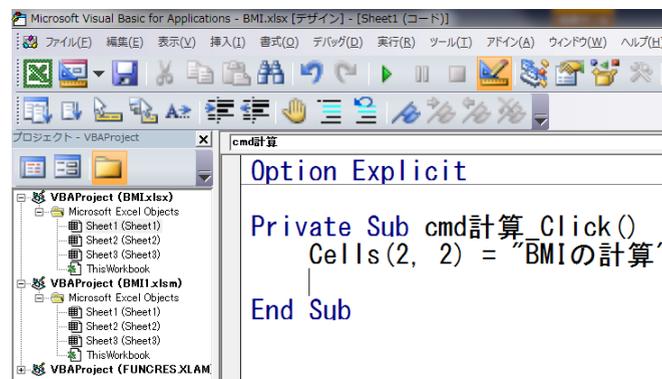
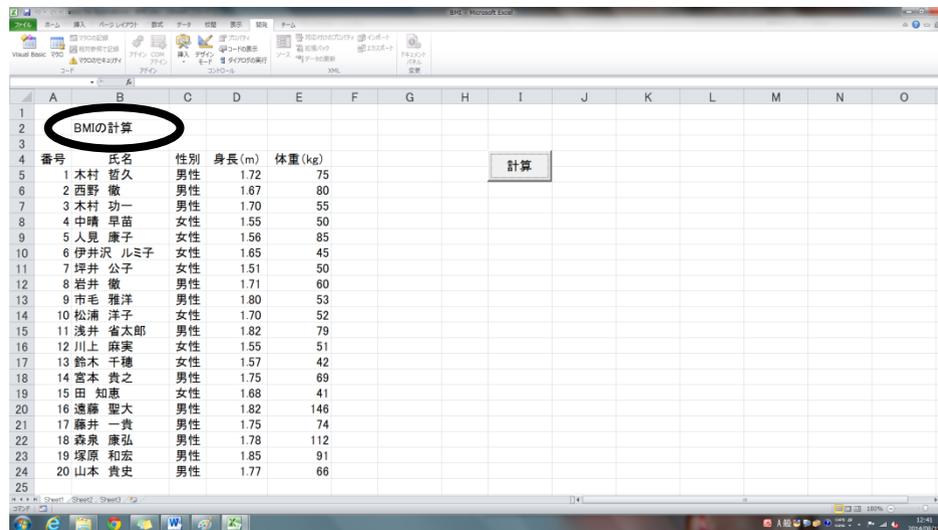


図 9 プログラムの入力

## ■プログラムの実行

- (1) **Excel 表示** ボタン (図 5 参照) を押し、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。



番号	氏名	性別	身長 (m)	体重 (kg)
1	木村 哲久	男性	1.72	75
2	西野 徹	男性	1.67	80
3	木村 功一	男性	1.70	55
4	中晴 早苗	女性	1.55	50
5	人見 康子	女性	1.56	85
6	伊井沢 ルミ子	女性	1.65	45
7	坪井 公子	女性	1.51	50
8	岩井 徹	男性	1.71	60
9	市毛 雅洋	男性	1.80	53
10	松浦 洋子	女性	1.70	52
11	浅井 春太郎	男性	1.82	79
12	川上 麻実	女性	1.55	51
13	鈴木 千穂	女性	1.57	42
14	宮本 貴之	男性	1.75	69
15	田 知恵	女性	1.68	41
16	遠藤 聖大	男性	1.82	146
17	藤井 一貴	男性	1.75	74
18	森泉 康弘	男性	1.78	112
19	塚原 和宏	男性	1.85	91
20	山本 貴史	男性	1.77	66

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.

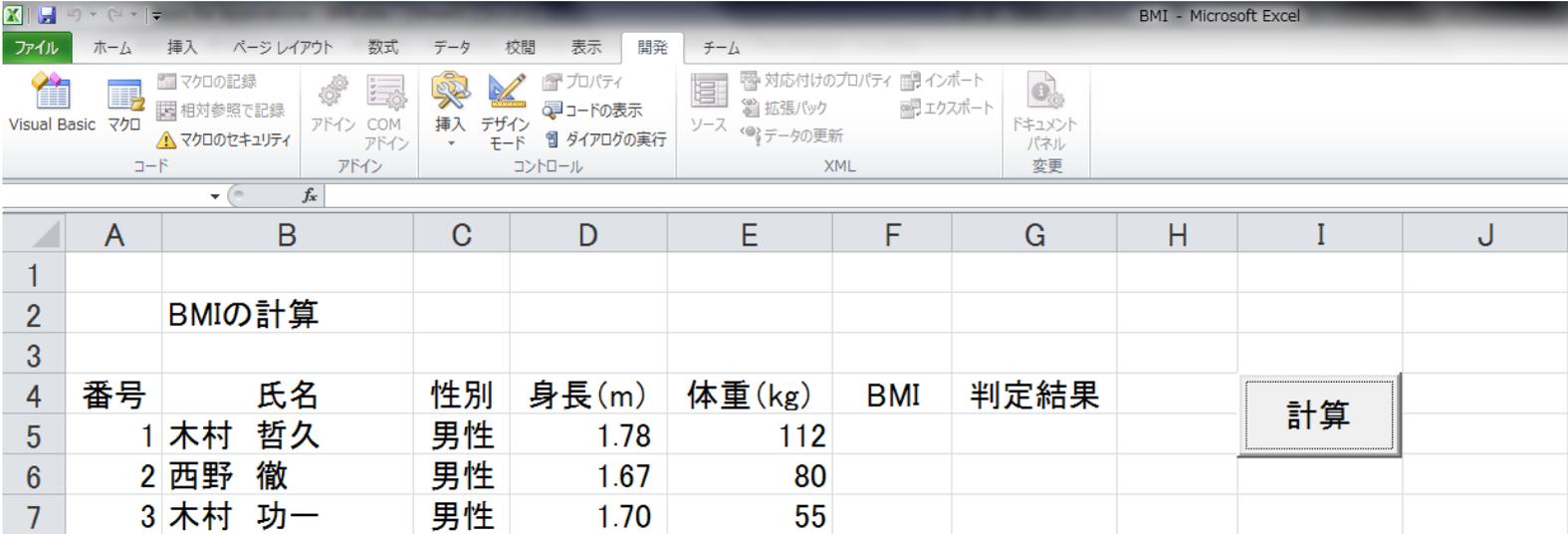


図 11 問題 1 の実行結果

## 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。

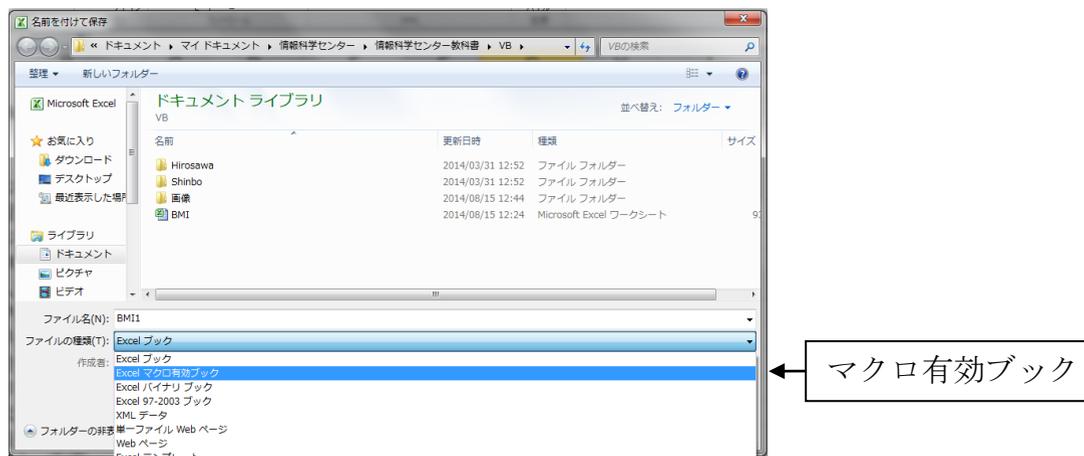


図 12 マクロ有効ブックとして保存

## 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。

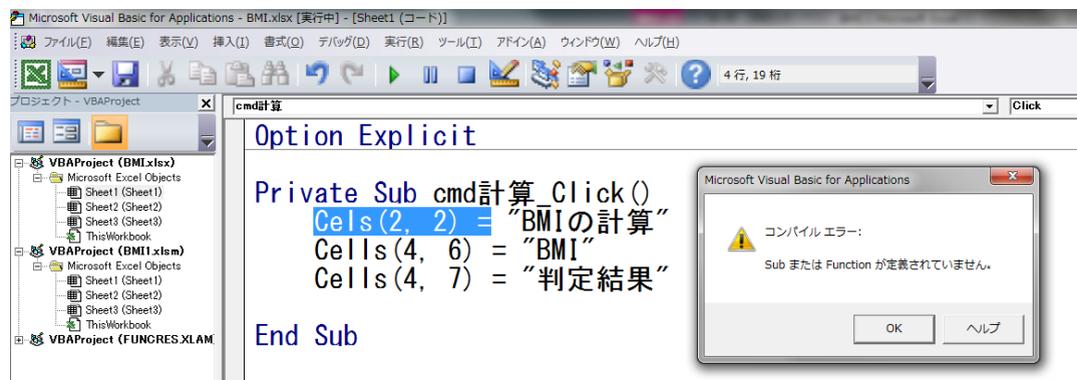


図 13 コンパイルエラーの発生

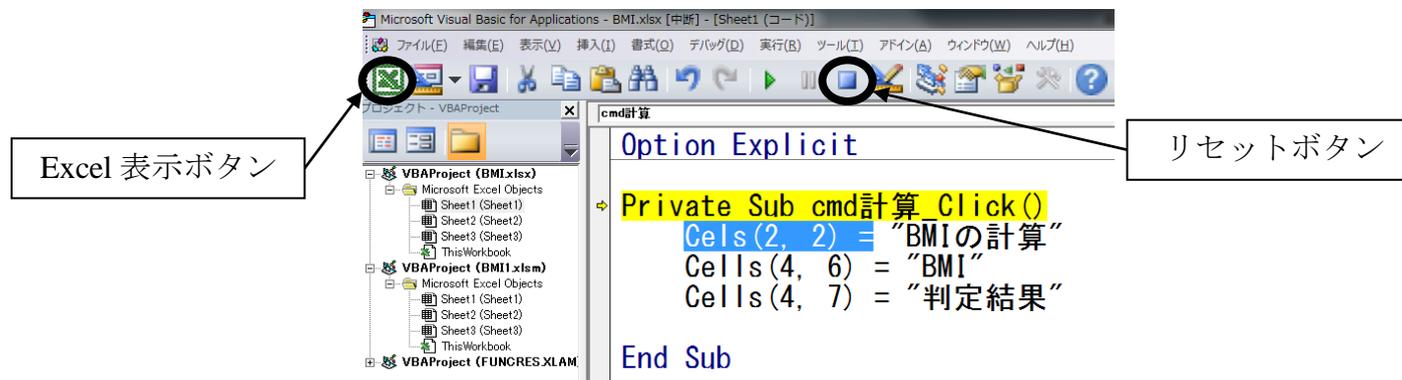


図 14 リセットボタン

## 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。

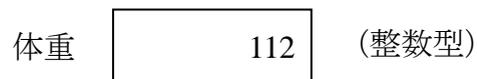


図 15 変数のイメージ

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ります。

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”

変数名 = セル名

変数名 = 計算式

(文字列の前後は ” で囲む)

(セル名は、Cells(i, j) のように記述する)

(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)

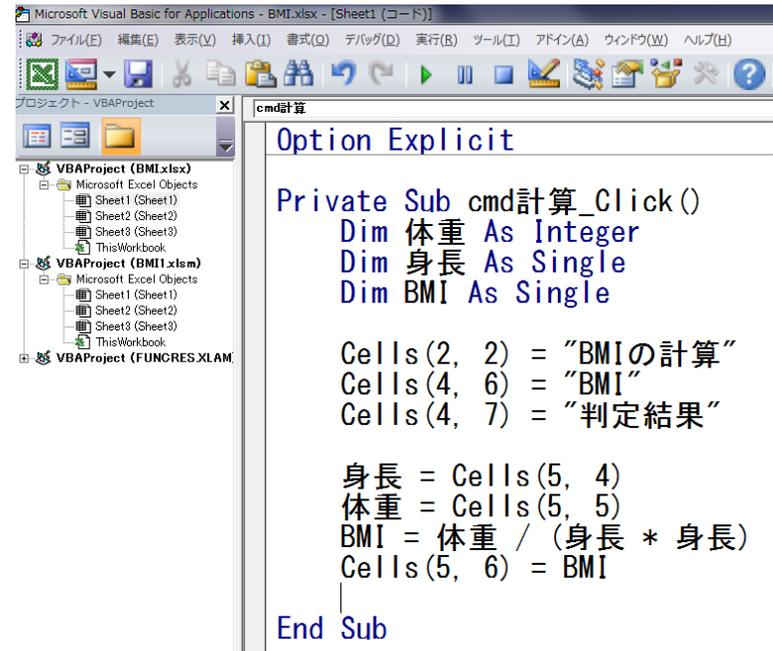
体重 = Cells(5, 5)

## ■BMI の計算

BMI = 体重 / (身長 \* 身長)

## ■セルへの代入

Cells(5, 6) = BMI



```

Option Explicit

Private Sub cmd計算_Click()
    Dim 体重 As Integer
    Dim 身長 As Single
    Dim BMI As Single

    Cells(2, 2) = "BMIの計算"
    Cells(4, 6) = "BMI"
    Cells(4, 7) = "判定結果"

    身長 = Cells(5, 4)
    体重 = Cells(5, 5)
    BMI = 体重 / (身長 * 身長)
    Cells(5, 6) = BMI
End Sub

```

図 16 例題 2 のコードエディタ

	A	B	C	D	E	F	G	H	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				
6	2	西野 徹	男性	1.67	80					
7	3	木村 功一	男性	1.70	55					

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

```
For N = 5 To 24
```

```
    身長 = Cells(N, 4)
```

```
    体重 = Cells(N, 5)
```

```
    BMI = 体重 / (身長 * 身長)
```

```
    Cells(N, 6) = BMI
```

```
Next N
```

その実行結果を図 18 に示します。

番号	氏名	性別	身長 (m)	体重 (kg)	BMI	判定結果
1	木村 哲久	男性	1.78	112	35.35	
2	西野 徹	男性	1.67	80	28.69	
3	木村 功一	男性	1.70	55	19.03	
4	中晴 早苗	女性	1.55	50	20.81	
5	人見 康子	女性	1.56	85	34.93	
6	伊井沢 ルミ子	女性	1.65	45	16.53	
7	坪井 公子	女性	1.51	50	21.93	
8	岩井 徹	男性	1.71	60	20.52	
9	市毛 雅洋	男性	1.80	53	16.36	
10	松浦 洋子	女性	1.70	52	17.99	
11	浅井 省太郎	男性	1.82	79	23.85	
12	川上 麻実	女性	1.55	51	21.23	
13	鈴木 千穂	女性	1.57	42	17.04	
14	宮本 貴之	男性	1.75	69	22.53	
15	田 知恵	女性	1.68	41	14.53	
16	遠藤 聖大	男性	1.82	146	44.08	
17	藤井 一貴	男性	1.75	74	24.16	
18	森泉 康弘	男性	1.72	75	25.35	
19	塚原 和宏	男性	1.85	91	26.59	
20	山本 貴史	男性	1.77	66	21.07	

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 二分岐処理（選択肢 1 つ）：処理 A を実行するか，実行しないか
- ② 二分岐処理（選択肢 2 つ）：処理 A を選択するか，処理 B を選択するか
- ③ 多分岐処理（選択肢 3 つ以上）：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 二分岐処理（選択肢 1 つ）：If ～ Then ～ End If 形

二分岐処理（選択肢 1 つ）には，If ～ Then ～ End If 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。比較される対象が数値データの場合は >= 30 のように記述しますが、文字列データの場合は ="男性" のように、前後を" (ダブルクォーテーション) で囲みます。

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
<>	等しくない

If ~ Then ~ End If 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
End If
```

	A	B	C	D	E	F	G	H	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				
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 の塗りつぶし処理を、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

```
For N = 5 To 24
    If Cells(N, 6) >= 30 Then
        Cells(N, 6).Interior.Color = vbYellow
    End If
Next N
```

番号	氏名	性別	身長(m)	体重(kg)	BMI	分類
1	木村 哲久	男性	1.78	112	35.35	
2	西野 徹	男性	1.67	80	28.69	
3	木村 功一	男性	1.70	55	19.03	
4	中晴 早苗	女性	1.55	50	20.81	
5	人見 康子	女性	1.56	85	34.93	
6	伊井沢 ルミ子	女性	1.65	45	16.53	
7	坪井 公子	女性	1.51	50	21.93	
8	岩井 徹	男性	1.71	60	20.52	
9	市毛 雅洋	男性	1.80	53	16.36	
10	松浦 洋子	女性	1.70	52	17.99	
11	浅井 省太郎	男性	1.82	79	23.85	
12	川上 麻実	女性	1.55	51	21.23	
13	鈴木 千穂	女性	1.57	42	17.04	
14	宮本 貴之	男性	1.75	69	22.53	
15	田 知恵	女性	1.68	41	14.53	
16	遠藤 聖大	男性	1.82	146	44.08	
17	藤井 一貴	男性	1.75	74	24.16	
18	森泉 康弘	男性	1.72	75	25.35	
19	塚原 和宏	男性	1.85	91	26.59	
20	山本 貴史	男性	1.77	66	21.07	

図 20 例題 5 の実行結果

## 5.6.2 二分岐処理（選択肢2つ）：If ～ Then ～ Else 形

二分岐処理（選択肢2つ）には、If ～ Then ～ 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

二分岐処理（選択肢2つ）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、図21のように、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。

番号	氏名	性別	身長(m)	体重(kg)	BMI	分類
1	木村 哲久	男性	1.78	112	35.35	
2	西野 徹	男性	1.67	80	28.69	
3	木村 功一	男性	1.70	55	19.03	
4	中晴 早苗	女性	1.55	50	20.81	
5	人見 康子	女性	1.56	85	34.93	
6	伊井沢 ルミ子	女性	1.65	45	16.53	
7	坪井 公子	女性	1.51	50	21.93	
8	岩井 徹	男性	1.71	60	20.52	
9	市毛 雅洋	男性	1.80	53	16.36	
10	松浦 洋子	女性	1.70	52	17.99	
11	浅井 省太郎	男性	1.82	79	23.85	
12	川上 麻実	女性	1.55	51	21.23	
13	鈴木 千穂	女性	1.57	42	17.04	
14	宮本 貴之	男性	1.75	69	22.53	
15	田 知恵	女性	1.68	41	14.53	
16	遠藤 聖大	男性	1.82	146	44.08	
17	藤井 一貴	男性	1.75	74	24.16	
18	森泉 康弘	男性	1.72	75	25.35	
19	塚原 和宏	男性	1.85	91	26.59	
20	山本 貴史	男性	1.77	66	21.07	

図 21 問題 2 の実行結果

### 5.6.3 多分岐処理（選択肢3つ以上）：If ～ Then ～ ElseIf 形

多分岐処理には、If ～ Then ～ 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]

ElseIf 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分岐処理の問題です。N行 (N=5~24) 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分岐処理のプログラム

```
For N = 5 To 24
    If Cells(N, 6) < 18.5 Then
        判定結果 = “低体重”
    ElseIf 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
```

番号	氏名	性別	身長(m)	体重(kg)	BMI	判定結果
1	木村 哲久	男性	1.78	112	35.35	肥満
2	西野 徹	男性	1.67	80	28.69	肥満
3	木村 功一	男性	1.70	55	19.03	普通体重
4	中晴 早苗	女性	1.55	50	20.81	普通体重
5	人見 康子	女性	1.56	85	34.93	肥満
6	伊井沢 ルミ子	女性	1.65	45	16.53	低体重
7	坪井 公子	女性	1.51	50	21.93	普通体重
8	岩井 徹	男性	1.71	60	20.52	普通体重
9	市毛 雅洋	男性	1.80	53	16.36	低体重
10	松浦 洋子	女性	1.70	52	17.99	低体重
11	浅井 省太郎	男性	1.82	79	23.85	普通体重
12	川上 麻実	女性	1.55	51	21.23	普通体重
13	鈴木 千穂	女性	1.57	42	17.04	低体重
14	宮本 貴之	男性	1.75	69	22.53	普通体重
15	田 知恵	女性	1.68	41	14.53	低体重
16	遠藤 聖大	男性	1.82	146	44.08	肥満
17	藤井 一貴	男性	1.75	74	24.16	普通体重
18	森泉 康弘	男性	1.72	75	25.35	肥満
19	塚原 和宏	男性	1.85	91	26.59	肥満
20	山本 貴史	男性	1.77	66	21.07	普通体重

図 22 例題 6 の実行結果

例題 6 のプログラムにおいて、「判定結果」の前後には ” がなく、「低体重」や「普通体重」、「肥満」の前後には ” があることに注意しましょう。「判定結果」は変数なので ” がなく、「低体重」は文字列の定数なのでその前後に ” をつける必要があります。” がないと変数と解釈されてしまい、不具合が生じます。

さて、上記の構文を使用して、下記の 6 分岐処理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。

番号	氏名	性別	身長(m)	体重(kg)	BMI	判定結果
1	木村 哲久	男性	1.78	112	35.35	肥満(3度)
2	西野 徹	男性	1.67	80	28.69	肥満(1度)
3	木村 功一	男性	1.70	55	19.03	普通体重
4	中晴 早苗	女性	1.55	50	20.81	普通体重
5	人見 康子	女性	1.56	85	34.93	肥満(2度)
6	伊井沢 ルミ子	女性	1.65	45	16.53	低体重
7	坪井 公子	女性	1.51	50	21.93	普通体重
8	岩井 徹	男性	1.71	60	20.52	普通体重
9	市毛 雅洋	男性	1.80	53	16.36	低体重
10	松浦 洋子	女性	1.70	52	17.99	低体重
11	浅井 省太郎	男性	1.82	79	23.85	普通体重
12	川上 麻実	女性	1.55	51	21.23	普通体重
13	鈴木 千穂	女性	1.57	42	17.04	低体重
14	宮本 貴之	男性	1.75	69	22.53	普通体重
15	田 知恵	女性	1.68	41	14.53	低体重
16	遠藤 聖大	男性	1.82	146	44.08	肥満(4度)
17	藤井 一貴	男性	1.75	74	24.16	普通体重
18	森泉 康弘	男性	1.72	75	25.35	肥満(1度)
19	塚原 和宏	男性	1.85	91	26.59	肥満(1度)
20	山本 貴史	男性	1.77	66	21.07	普通体重

図 23 問題 3 の実行結果

## 6 BMI カードの作成

### 6.1 マクロの記録

Excel で操作した手順をマクロプログラムとして記録してくれる便利な機能があります。それが「マクロの記録」機能です。それを使うと、Excel のある操作が Visual Basic でどう記述されるかがわかるので、プログラムを作成するときの参考になります。ここでは、マクロの記録を利用して、シートのコピーが Visual Basic でどう記述されるかをみていきましょう。

ここで使用するツールボタンは、

マクロの記録：マクロの記録を開始させるボタン

記録終了：マクロの記録を終了させるボタン（マクロの記録中、マクロの記録ボタンの位置に出現する）

マクロ：記録したマクロプログラムを、編集したり、実行したりするボタン

の3つです。



図 24 マクロ関連のツールボタン

## ◇例題7 (マクロの記録)

マクロ記録.xlsx ファイルのカードシートを、そのシートの後ろにコピーする操作のマクロプログラムの記録をとりなさい、

## ◇例題7の手順

- ① マクロ記録.xlsx ファイルを開き、開発タブを表示させます。
- ② 「カード」シート (Sheets(2)) を選択して、[マクロの記録]ツールボタンを押します。
- ③ 図 25 のダイアログボックスからマクロ名「シートのコピー」と入力し、OK ボタンをクリックします。  
ここからの操作がマクロプログラムとして記録されます。
- ④ 「カード」シートのシート名の上にカーソルを移動し、右ボタンをクリックします。するとショートカットメニューが出現するので、「移動またはコピー」を選択します。
- ⑤ 図 26 のように、[コピーを作成する]のチェックボックスにチェックし、コピーの挿入先として「カード」の後ろの「(末尾へ移動)」を選択し、OK ボタンをクリックします。
- ⑥ 「カード」シートの後ろに「カード(2)」シートが作成されて、その内容が「カード」シートと同じであることを確かめます。
- ⑦ [記録終了]ツールボタンを押します。これにより、記録が終了します。
- ⑧ 記録したマクロプログラムを確かめるために、[マクロ]ツールボタンを押します。すると、図 27 のような[マクロ]ダイアログボックスが出現するので、マクロ名として「シートのコピー」とを選択し、[編集]ボタンを押します。
- ⑨ 図 28 のように、記録を取得したマクロプログラムが出現します。

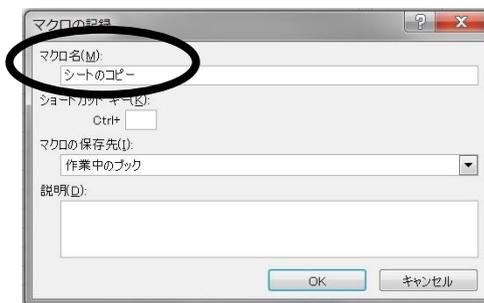


図 25 マクロ名の入力

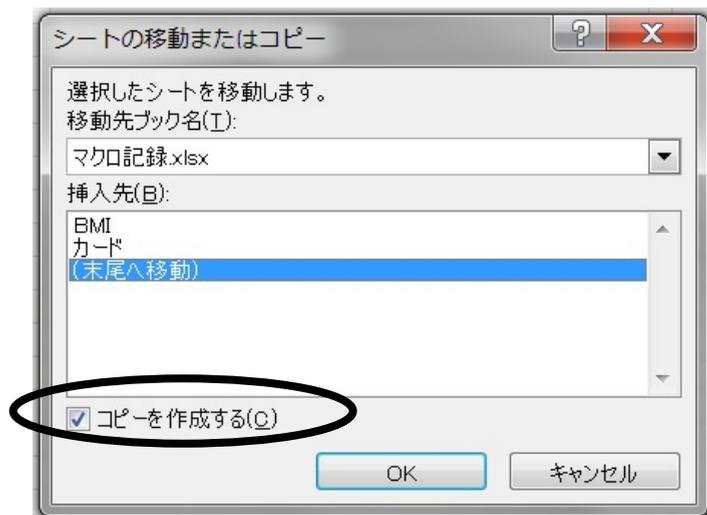


図 26 シートのコピー

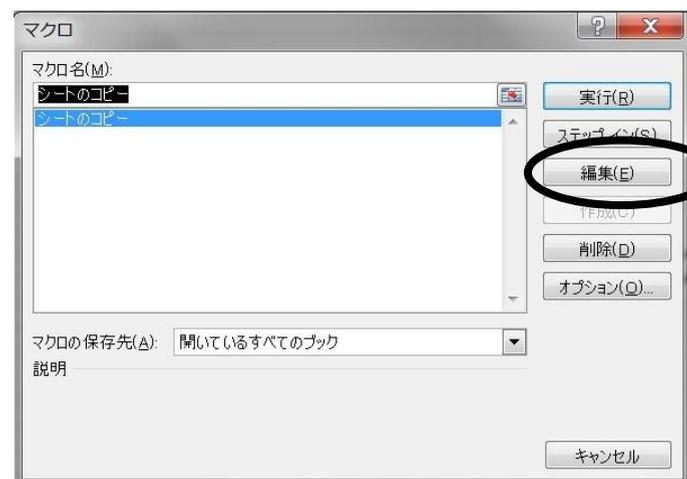


図 27 「マクロ」ダイアログボックス

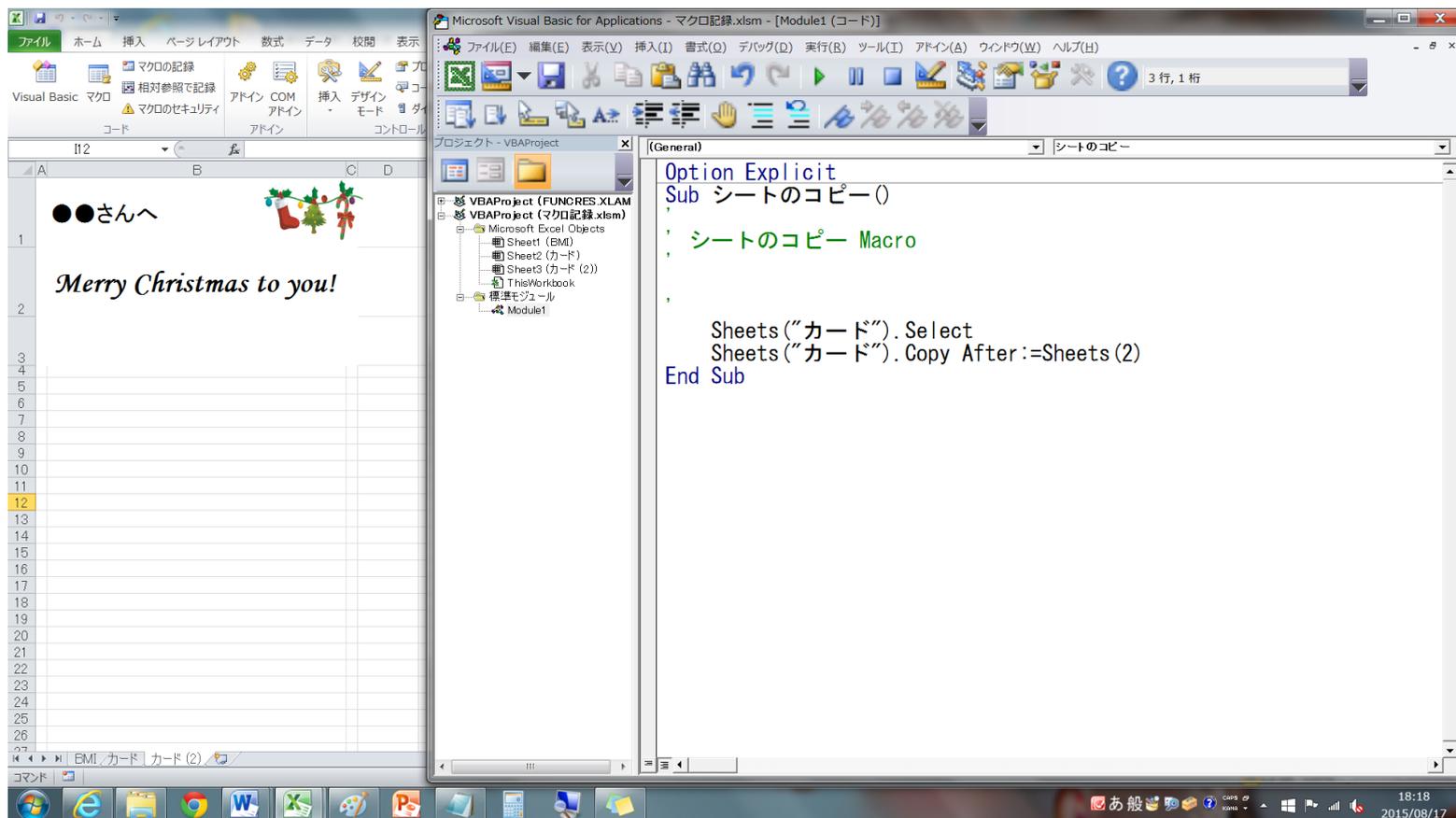


図 28 マクロプログラムの編集

◇ 「シートのコピー」のマクロプログラム

```
Sub シートのコピー()  
    ' シートのコピー Macro  
    Sheets("カード").Select  
    Sheets("カード").Copy After := Sheets(2)  
End Sub
```

このプログラムでは、

① 「カード」という名前のシートを選択する

② 「カード」シートをコピーし、それを **Sheets(2)**（「カード」シートのこと）の後ろに挿入する

という手順が記録されています。なお、プログラムの1行目のシングル・クォーテーション（'）で始まる緑色の文はコメント文を表し、プログラムには影響を与えない文です、プログラムの説明に使用されます。

## ◇例題 8 (マクロの編集と実行)

例題 7 で取得したマクロプログラムを少しだけ編集し、編集したプログラムを実行させなさい。編集内容は、「カード」シートをコピーし、そのシートを「カード」シートの後ろに挿入させることです。

## ◇例題 8 の手順

- ① [マクロ]ツールボタンを押すと、[マクロ]ダイアログボックスが出現するので、「シートのコピー」を選択し、[編集]ボタンを押します。
- ② プログラムを編集するためのコードエディタが、図 28 のように表示されるので、2 行目の  
`Sheets("カード").Copy After :=Sheets(2)`  
の部分を  
`Sheets("カード").Copy After := Sheets("カード")`  
に置き換えます。
- ③ プログラムを実行するために、Excel の[開発]タブに切り替えて、[マクロ] → [実行]ボタンを押します。
- ④ 「カード」シートがコピーされ、その「カード(3)」シートが「カード」シートの後ろに挿入されます。

マクロの実行を繰り返すと、次々と「カード」シートがコピーされ、「カード」シートの後ろに挿入されていくことが観察できます。このように、「マクロの記録」機能は、Excel の操作を Visual Basic ではどう記述されるかを確認するためにも便利な機能です。最後に、マクロ有効ブックとして上書き保存しましょう。

## 6.2 カードボタンの作成

5.6 節で作成した Excel マクロ有効ブック BMI.xlsm を開き、その「カード」シートを選択すると、そこに健康診断結果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、「BMI」シートで求めた各人の BMI の結果を報せる健康診断結果カードを生成してみましょう。まず、「BMI」シートの「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 29 に作成したコマンドボタンを示します。

	A	B	C	D	E	F	G	H	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)		カード	
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			
8	4	中晴 早苗	女性	1.55	50	20.81	普通体重			
9	5	人見 康子	女性	1.56	85	34.93	肥満(2度)			

図 29 カードボタンの作成

## 6.3 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、「カード」シートのコピーを生成し、それを「カード」シートの後ろに挿入するプログラムを作成してみましょう。6.1 節のマクロの記録を使用してわかったように、シート 1 のコピーを生成し、それをシート 2 の後ろに挿入させる処理の構文は以下のとおりです。

### ■シートのコピー

```
Sheets("シート 1").Copy After := Sheets("シート 2")
```

ここで、シート名の前後に ” をつけることに注意しましょう。また、どのシートの後ろに挿入させるかは、After := で指定します。指定されたシートの後ろに挿入されます。

### ◇例題 9

「カード」シートのコピーを生成し、それを「カード」シートの後ろに挿入するプログラムを作成しなさい。

### ◇例題 9 のプログラム

```
Sheets("カード").Copy After := Sheets("カード")
```

ここで、挿入されたシート名は「カード(2)」となります。その実行結果とプログラムを図 30 に示します。

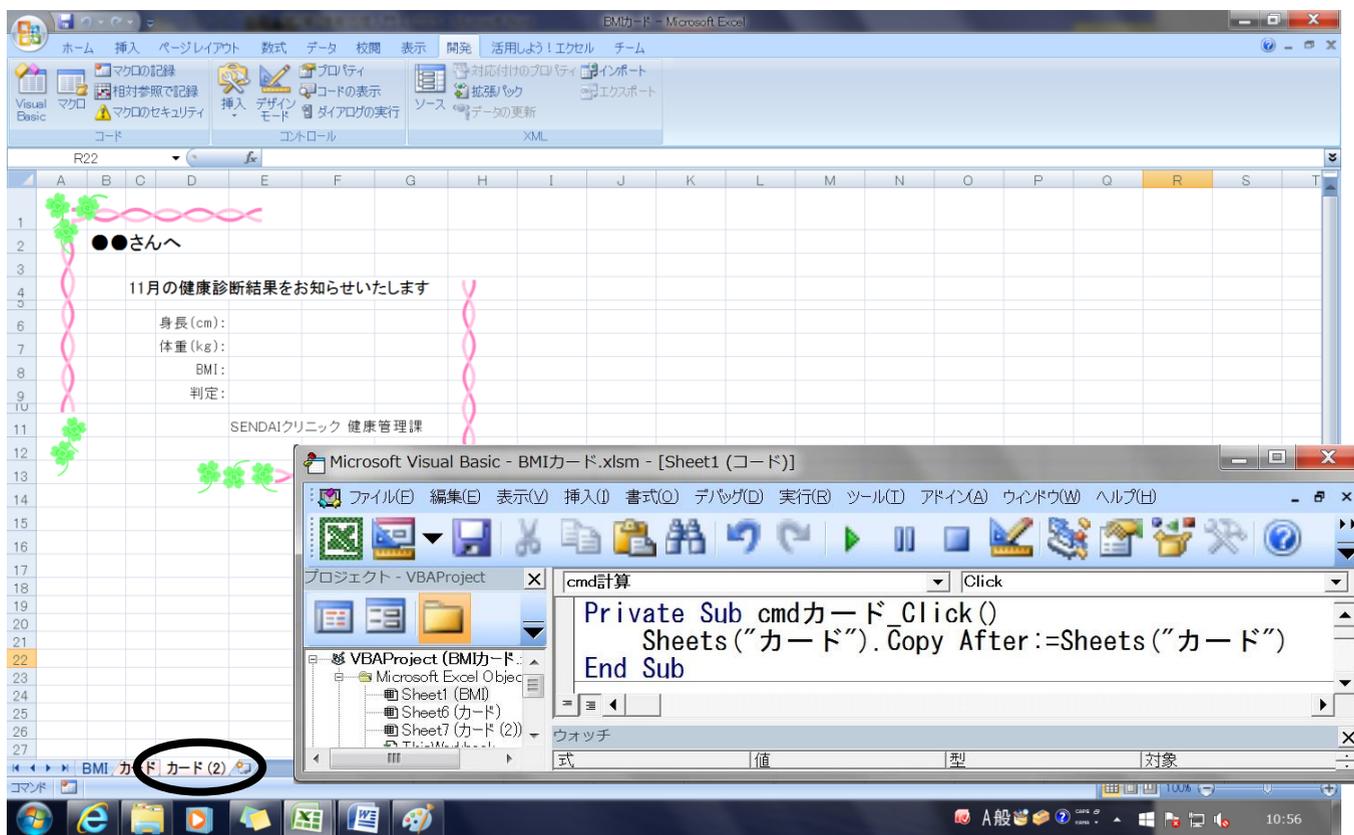


図 30 例題 9 の実行結果とプログラム

## 6.4 診断結果カードの作成

### ◇例題 10

コピーした「カード(2)」シートに「BMI」シートの5行目の人の宛先と身長と体重とBMIを図31のように表示させるプログラムを作成しなさい。

まず、「宛先」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`

5行目の宛先の氏名「木村 哲久」は「BMI」シートの5行2列目のセルに記憶されています。それを変数「宛先」に代入しましょう。

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。

	A	B	C	D	E	F	G	H
1								
2		木村 哲久さんへ						
3								
4		11月の健康診断結果をお知らせいたします						
5								
6			身長(cm):	178				
7			体重(kg):	112				
8			BMI:	35.34907				
9			判定:					
10								
11						SENDAIクリニック 健康管理課		
12								
13								
14								

図 31 診断結果カード

**■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)

次に、「木村 哲久さんへ」とするために、「木村 哲久」という文字列と「さんへ」という文字列を連結させましょう。

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、「BMI」シートの 5 行目 2 列目の氏名「木村 哲久」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(“BMI”).Cells(5, 2) & “さんへ”

とすればよいです。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。

次に、「宛先」と「身長」と「体重」と「BMI」の値を、コピーした「カード」シートの決められたセルに書き込む処理を行います。「宛先」は B2 のセルに、「身長」は E6 のセルに、「体重」は E7 のセルに、「BMI」は E8 のセルに代入すればよいので、プログラムは、

```
ActiveSheet.Cells(2, 2) = 宛先
```

```
ActiveSheet.Cells(6, 5) = Sheets(“BMI”).Cells(5, 4)
```

```
ActiveSheet.Cells(7, 5) = Sheets(“BMI”).Cells(5, 5)
```

```
ActiveSheet.Cells(8, 5) = Sheets(“BMI”).Cells(5, 6)
```

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 10 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 9 でコピーした「カード(2)」シートを削除してから実行するようにして下さい。

◇例題 10 のプログラム

```
Sheets(“カード”).Copy After := Sheets(“カード”)
```

```
宛先 = Sheets(“BMI”).Cells(5, 2) & “さんへ”
```

```
ActiveSheet.Cells(2, 2) = 宛先
```

```
ActiveSheet.Cells(6, 5) = Sheets(“BMI”).Cells(5, 4)
```

```
ActiveSheet.Cells(7, 5) = Sheets(“BMI”).Cells(5, 5)
```

```
ActiveSheet.Cells(8, 5) = Sheets(“BMI”).Cells(5, 6)
```

この実行結果 (図 31) では、BMI の数値の桁数が多すぎることや、数値のフォントが目立たないことが気になります。そこで、次の例題 11 をやってみましょう。

## ◇例題 11

例題 10 に変更を加え，図 32 のように，BMI の桁数を小数点以下 2 桁で表示し，数値データのフォントサイズを 16 ポイント，さらに，フォントの色を赤にしなさい。



	A	B	C	D	E	F	G	H
1	木村 哲久さんへ							
2	11月の健康診断結果をお知らせいたします							
3								
4								
5								
6				身長 (cm):	1.78			
7				体重 (kg):	112			
8				BMI:	35.35			
9				判定:				
10								
11				SENDAIクリニック 健康管理課				
12								
13								
14								

図 32 例題 11 の診断結果カード

数値データの桁数を調整するには，下記の書式を使用します。

■ i 行 j 列のセルの数値データを小数点以下 2 桁で表示させる書式

Cells(i, j).NumberFormat = "0.00"

書式の基本は「#」と「0」と小数点の「.」です。「#」の場合は、表示する数値の桁がない場合は表示しませんが、「0」の場合は0を補って表示します。整数部分が書式で指定した桁数よりも多いときはそのまま表示します。一方、小数点以下の場合、書式で指定した桁数の位置で四捨五入して表示します。その例を表4に示します。

表4 書式の例

書式	数値データ	表示結果
##.##	1.2	1.20
##.##	12.34	12.34
##.##	123.456	123.46
00.00	1.2	01.20
00.00	12.34	12.34
00.00	123.456	123.46

フォントのサイズや色を指定するには、以下の代入文を使います。

- i行j列のセルのフォントサイズを16ポイントで表示させる

```
Cells(i, j).Font.Size = 16
```

- i行j列のセルのフォントの色を赤で表示させる

```
Cells(i, j).Font.Color = vbRed
```

例題 11 のプログラムは下記のようになります。

◇例題 11 のプログラム

```
ActiveSheet.Cells(8, 5).NumberFormat = "0.00"
```

```
ActiveSheet.Cells(6, 5).Font.Size = 16
```

```
ActiveSheet.Cells(7, 5).Font.Size = 16
```

```
ActiveSheet.Cells(8, 5).Font.Size = 16
```

```
ActiveSheet.Cells(6, 5).Font.Color = vbRed
```

```
ActiveSheet.Cells(7, 5).Font.Color = vbRed
```

```
ActiveSheet.Cells(8, 5).Font.Color = vbRed
```

## ◇問題 4

例題 11 の診断結果カード (5 行目の人) の「判定」として「BMI」シートの判定結果を図 33 のように表示させなさい。その際、フォントサイズを 20 ポイントとし、フォントの色は、低体重の場合は赤、普通体重の場合は黒、肥満の場合は青で表示させなさい。

	A	B	C	D	E	F	G	H
1								
2				木村 哲久さんへ				
3								
4				11月の健康診断結果をお知らせいたします				
5								
6				身長(cm):	1.78			
7				体重(kg):	112			
8				BMI:	35.35			
9				判定:	肥満(3度)			
10								
11					SENDAIクリニック 健康管理課			
12								
13								
14								

図 33 問題 4 の実行結果

## 6.5 複数の BMI カードの生成

### ◇例題 12

例題 11 を変更して、「BMI」シートの 5 行目から 7 行目までの 3 人分に対して、健康診断結果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 11 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、行番号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、変数「宛先」への代入文と、「身長」と「体重」と「BMI」の情報をカードの決められたセルへ表示させる代入文において、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 (N=5 ~ 7) の人の情報を代入するためです。

その実行結果を図 34 に示します。「カード」シートのコピーが「カード(4)」と「カード(3)」と「カード(2)」の 3 つのシートが生成され、7 行目から 5 行目までの診断結果が書き込まれています。

さらに、For 文におけるカウンタ **N** の終了値 7 を 24 に変えれば、20 名分の診断結果カードを生成できます。

## ◇例題 12 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("カード").Copy After:=Sheets("カード")
```

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

```
    ActiveSheet.Cells(2, 2) = 宛先
```

```
    ActiveSheet.Cells(6, 5) = Sheets("BMI").Cells(N, 4)
```

```
    ActiveSheet.Cells(7, 5) = Sheets("BMI").Cells(N, 5)
```

```
    ActiveSheet.Cells(8, 5) = Sheets("BMI").Cells(N, 6)
```

```
    ActiveSheet.Cells(8, 5).NumberFormat = "0.00"
```

```
    ActiveSheet.Cells(6, 5).Font.Size = 16
```

```
    ActiveSheet.Cells(7, 5).Font.Size = 16
```

```
    ActiveSheet.Cells(8, 5).Font.Size = 16
```

```
    ActiveSheet.Cells(6, 5).Font.Color = vbRed
```

```
    ActiveSheet.Cells(7, 5).Font.Color = vbRed
```

```
    ActiveSheet.Cells(8, 5).Font.Color = vbRed
```

```
Next N
```

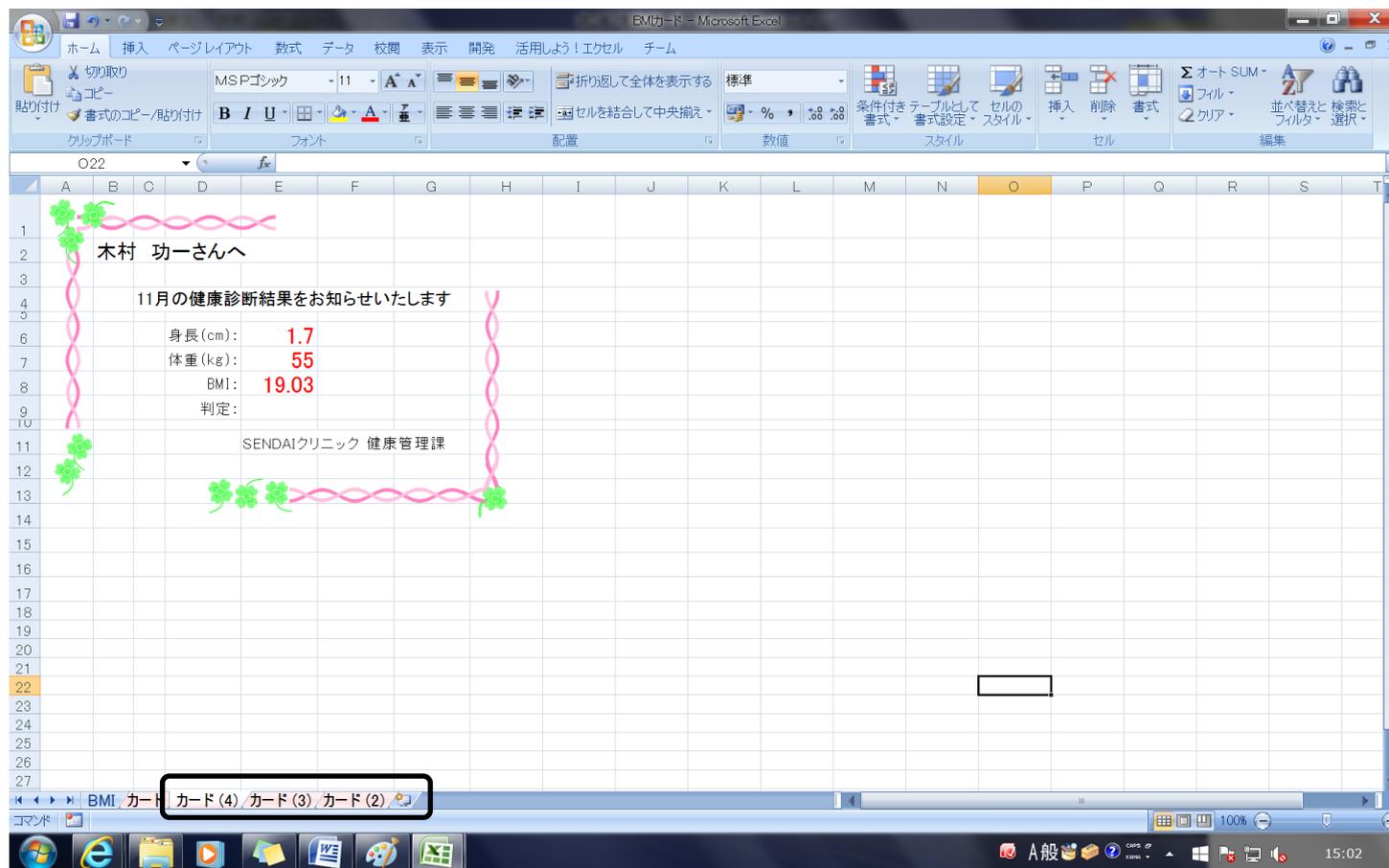


図 34 例題 12 の実行結果

## ◇問題 5

例題 12 の診断結果カード (5 行目から 7 行目の人) の「判定」として「BMI」シートの判定結果を図 35 のように表示させなさい。その際、フォントサイズを 20 ポイントとし、フォントの色は、低体重の場合は赤、普通体重の場合は黒、肥満の場合は青で表示させなさい。

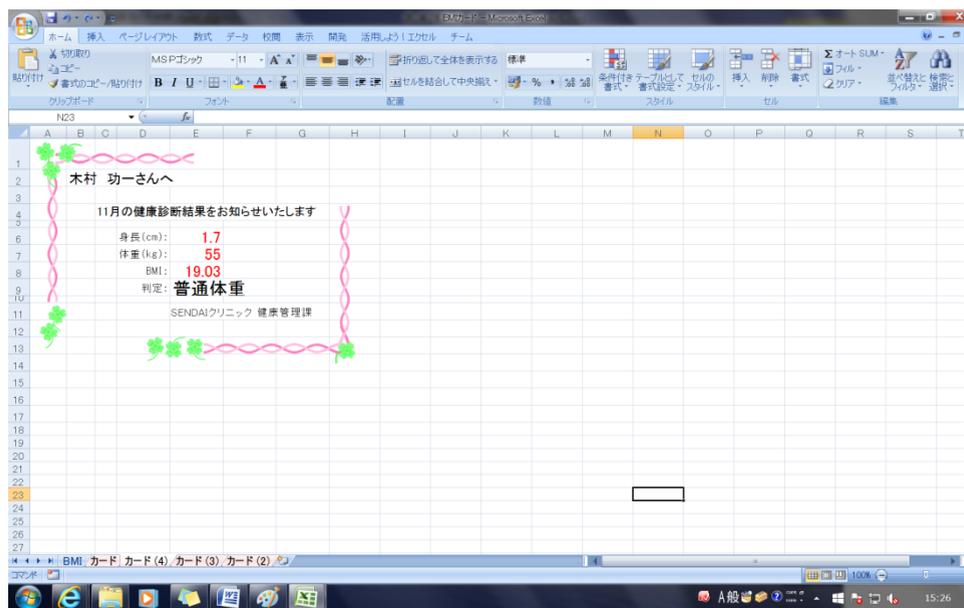


図 35 問題 5 の実行結果