

表計算ソフト(Excel)のマクロを利用したプログラムの実践事例 ープログラム指導初心者向けマニュアルー

八幡浜高等学校
水成 洋

1 主題の設定理由

平成 34 年度から年次進行により実施予定の新学習指導要領では、新教科「情報 I」においてプログラミング教育が必修化される見通しである。

本校では 1 年次に「情報の科学」(2 単位)を履修(使用教科書は日本文教出版の「新・情報の科学」を使用)しており、実際にプログラミングの指導も行っている。しかし、この指導において私自身、様々なプログラム言語については不勉強であり、私のようなプログラム指導初心者にも簡単に扱えるものはないかと試行錯誤していた。これは、今後、必修化されることで私と同じようなプログラム指導初心者の先生方も大いに悩まれている問題ではないかと思う。

そこで、今回、私が実際に行っている表計算ソフトのマクロを利用した実践事例を示すことで、私と同じようなプログラム指導初心者の先生方の一助となればと思い、この主題を設定した。

2 現状の指導について

プログラム学習で使用するプログラム言語については、生徒の実情に応じたものを選択する必要がある。実際、本校の生徒も中学校段階での習熟度が様々で、その中には自分でプログラムを組むことができる者もいれば、表計算ソフトも使ったことのない生徒もあり、各生徒の情報リテラシーの格差が大きい。

そこで、私は、生徒が将来必要となるであろう情報リテラシーを高めさせるために、授業始めに 5 分間のタイピング練習を行わせ、授業では本校の商業科教員に作成してもらった実習プリントを使って表計算ソフト(Excel)の実習を行っている。

表計算ソフトの実習では、SUM や AVERAGE のような簡単な関数のほかに IF 関数などを使って簡単なアルゴリズムなどについても実習させており、実習のまとめとして最後に実技テストも実施して評価を行っている。また、プログラム作成の実習では、教科書内の「基本的なアルゴリズムとプログラム」の章に沿って行っており、ここでは、プログラム学習で使うプログラム言語として、JavaScript と表計算マクロ言語の両方が提示されているが、その言語の選択については、表計算マクロ言語を選択した。それは、

- ① 私が JavaScript についての知識が乏しいこと
 - ② 生徒が新しいもの(Javascript)でなく表計算実習の延長として取り扱えること
- 以上の理由からである。

3 実践(プログラム作成指導マニュアル)

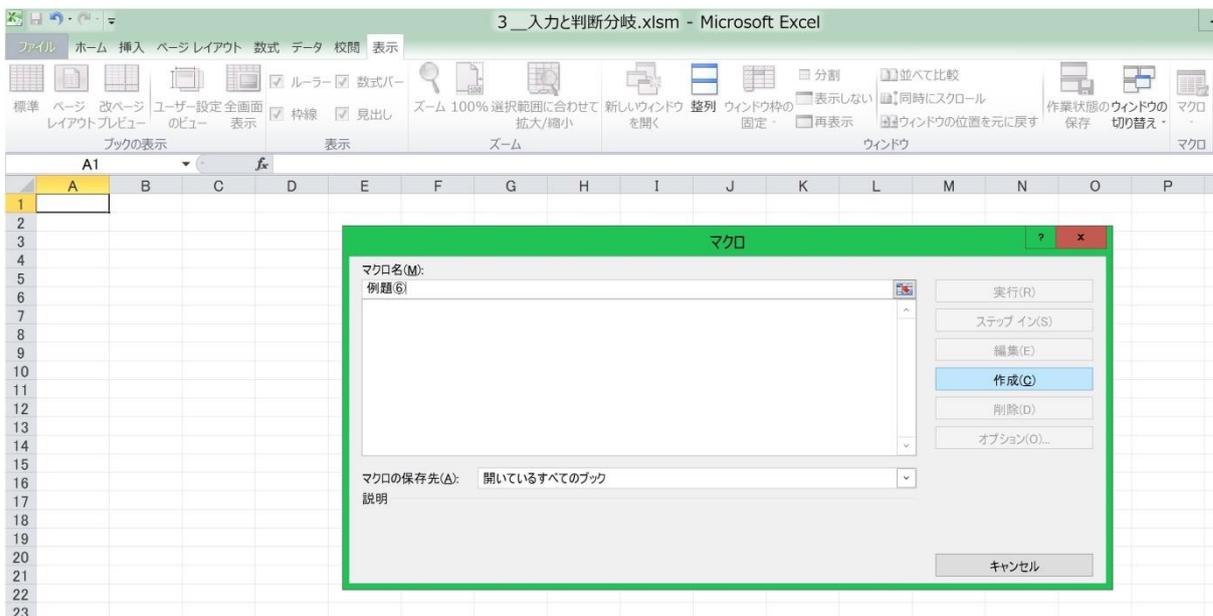
マクロとは、Excel において、複数の手順を記憶して、自動的に実行させる機能のことであり、そのマクロを作成するときに VBA(Visual Basic for Applications)というプログラミング言語を使用している。今回、VBA でのプログラミングを行うために、Excel に標準装備されている、VBE(Visual Basic Editor)という機能を使ってプログラムコードを打ち込んでいく。

(1) マクロの作成(初めて作成するとき)

- ① Excel を起動し、**ファイル→名前を付けて保存** でファイル名を付ける。
- ② **ファイル名**の下にある**ファイルの種類**から、**Excel マクロ有効ブック**を選択し保存する。



- ③ **表示→マクロ→マクロの表示**を選択しマクロ名を付け、**作成**のボタンを押す。



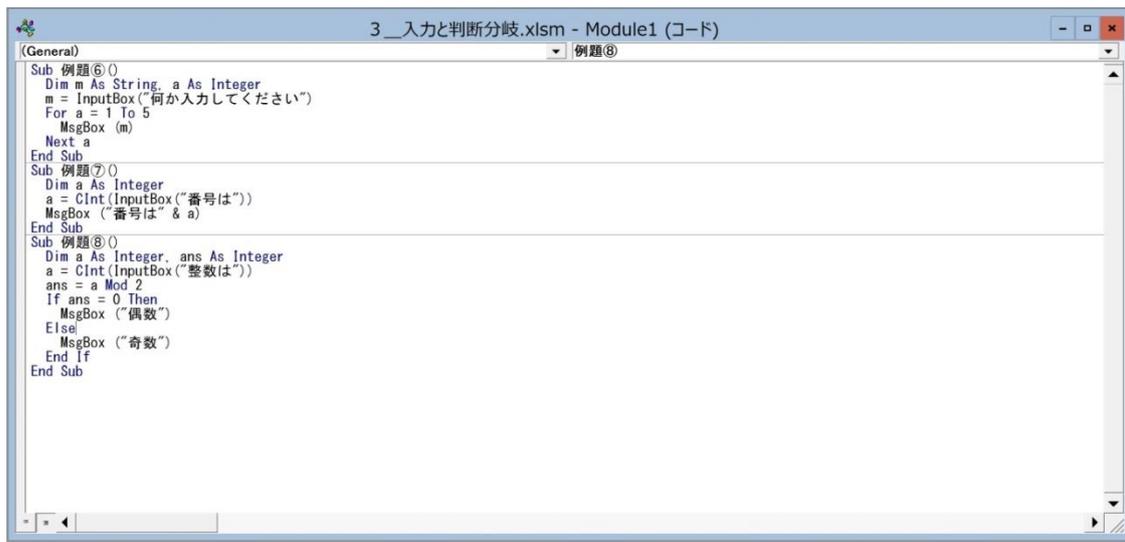
(2) プログラムの作成、編集

- ① プログラム入力画面の **Sub** と **End Sub** の間にプログラムを記述する。



```
Sub 例題⑥()  
Dim m As String, a As Integer  
m = InputBox("何か入力してください")  
For a = 1 To 5  
    MsgBox (m)  
Next a  
End Sub
```

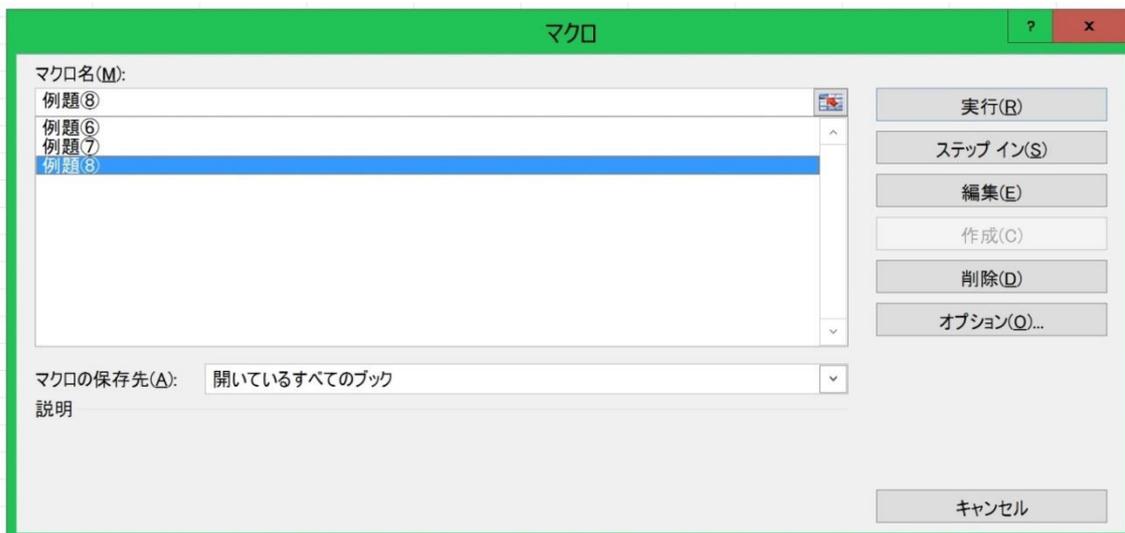
- ② コードは簡単にコピー、ペーストできるので、新規のプログラムは新しいマクロ名を付けて追加作成する。作ったコードはこまめに上書き保存しておく。



```
Sub 例題⑥()  
Dim m As String, a As Integer  
m = InputBox("何か入力してください")  
For a = 1 To 5  
    MsgBox (m)  
Next a  
End Sub  
Sub 例題⑦()  
Dim a As Integer  
a = CInt(InputBox("番号は"))  
MsgBox ("番号は" & a)  
End Sub  
Sub 例題⑧()  
Dim a As Integer, ans As Integer  
a = CInt(InputBox("整数は"))  
ans = a Mod 2  
If ans = 0 Then  
    MsgBox ("偶数")  
Else  
    MsgBox ("奇数")  
End If  
End Sub
```

(3) プログラムの編集、作成(2回目以降)

表示→マクロ→マクロの表示を選択すると、作成されたマクロ名が表示されているので、編集したいマクロ名を選択し**編集**のボタンを押す。マクロの追加作成は編集画面から行う。



(4) プログラムの実行

表示→**マクロ**→**マクロの表示**を選択すると、作成されたマクロ名が表示されているので、実行したいマクロ名を選択し**実行**のボタンを押す。

(5) 動作確認およびデバッグ

作成したプログラムが想定した通り動作するか確認し、想定した動作をしなかった場合(エラー表示されるなど)は、誤り箇所(bug バグ)を見つけ、プログラムを修正(debug デバッグ)する。

プログラミングは正確に記述することが大切であり、ピリオドやコンマ、スペースの有る無し、スペースを半角で入力していない等でもエラー表示されてしまう。また、プログラムコードは **ASCII** 文字で入力する必要があるため、プログラミングの途中で、表示される文字列として日本語など全角文字を入力した場合は、プログラムコードの打ち込みを再開する時に半角英数字入力に戻しておく。プログラムコードを入力する際には、こまめに保存、実行、修正することが必要である。

4 プログラム作成例

教科書に掲載されているプログラムや練習問題等で作成したプログラム例を示す。

1 __簡単なプログラム～文字の表示と繰り返し



```
Sub 例題①()  
MsgBox ("情報")  
End Sub  
Sub 例題②()  
MsgBox ("情報")  
MsgBox ("の")  
MsgBox ("科学")  
End Sub  
Sub 例題③()  
Dim a As Integer  
For a = 1 To 5  
MsgBox ("情報")  
Next a  
End Sub
```

○Msgbox (表示文字列)

かっこ内の文字列などをメッセージボックスとして表示する。日本語など全角文字を表示させるときは ” ” が必要である。

○繰り返し処理

```
For 変数=初期値 To 最終値 (Step 加算値)  
    実行される内容  
Next 変数
```

初期値から最終値まで変数が加算値ずつ増えながら (Step 以下は省略することができ、その場合の加算値は 1)、For 文から Next 文までの命令文が繰り返し実行される。使用する変数については、最初にその型を定義しておく。

○変数の定義 (Dim 変数 As 変数の型)

変数の型 (主なもの)

Integer 整数型 (-32,768～32,768 の整数) String 文字列型 (文字列)
Variant バリエーション型 (すべてのデータ)

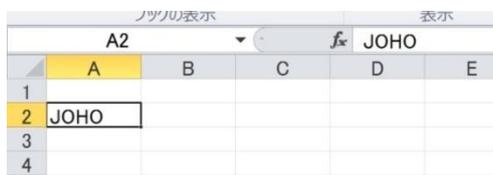
2 __簡単なプログラム～変数と計算



```
Sub 例題④()  
    Dim a As Integer, b As Integer  
    a = 1  
    b = 4  
    Cells(1, 1).Value = a + b  
    Cells(2, 1).Value = a - b  
    Cells(3, 1).Value = a * b  
    Cells(4, 1).Value = a / b  
End Sub  
Sub 例題⑤()  
    Dim s As Integer, a As Integer  
    s = 0  
    For a = 1 To 5  
        s = s + a  
    Next a  
    MsgBox (s)  
End Sub
```

○Cells(行, 列). Value

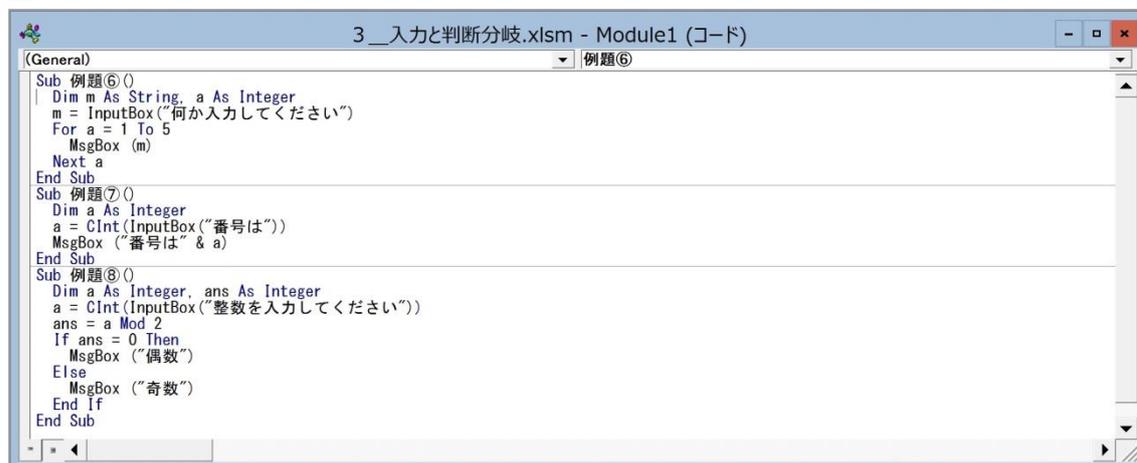
セルに表示させたい場合は Cells(□, □). Value を使う。例えば、A 列 2 行目のセルに JOHO を表示したいときは Cells(2, 1). Value = “JOHO” とする。



○例題⑤ s = s + a での = の意味

数学で扱う「等しい」という意味の = ではなく、「代入」(右辺 s + a の値を左辺の s の値とする命令)。

3 __簡単なプログラム～入力と判断分岐



```
Sub 例題⑥()  
    Dim m As String, a As Integer  
    m = InputBox("何か入力してください")  
    For a = 1 To 5  
        MsgBox (m)  
    Next a  
End Sub  
Sub 例題⑦()  
    Dim a As Integer  
    a = CInt(InputBox("番号は"))  
    MsgBox ("番号は" & a)  
End Sub  
Sub 例題⑧()  
    Dim a As Integer, ans As Integer  
    a = CInt(InputBox("整数を入力してください"))  
    ans = a Mod 2  
    If ans = 0 Then  
        MsgBox ("偶数")  
    Else  
        MsgBox ("奇数")  
    End If  
End Sub
```

○InputBox 関数

値の入力を要求し、入力された値を文字列として扱う関数。

○CInt 関数

InputBox でフォームに記入されたものは文字列とみなされるため、その数字を使って計算をすることができないので文字列を数値に変換する関数。

○例題⑦ MsgBox (“番号は” & a) について

文字列には ” ” が必要であるが、変数には” ” が必要ない。また、文字列や変数など複数のものを同時に表示させるときは & で繋げておく。

○Mod(例題⑧)

プログラムで用いる演算子。割り算の余りを意味する。

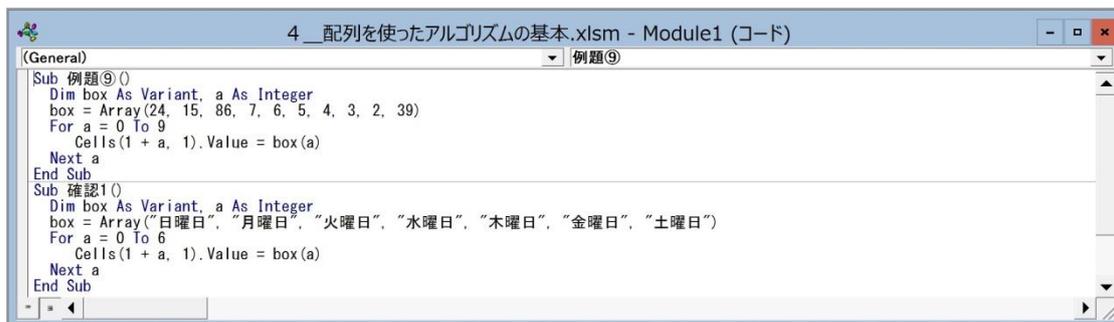
その他、×(かけ算)は*、÷(割り算)は/、等しくないは<>、以上は>=等表計算ソフトで扱った関数の演算子と同じである。

○判断分岐処理

```
If 条件式 Then
    (条件式が真の場合)実行される内容
Else
    (条件式が偽の場合)実行される内容
End If
```

条件が満たされた場合のみ処理を実行する場合、Else~の記述は不要である。条件が2種類以上あるときはElseを複数加えることで条件を追加できる。

4 __配列を使ったプログラム



```
Sub 例題⑨()
    Dim box As Variant, a As Integer
    box = Array(24, 15, 86, 7, 6, 5, 4, 3, 2, 39)
    For a = 0 To 9
        Cells(1 + a, 1).Value = box(a)
    Next a
End Sub
Sub 確認1()
    Dim box As Variant, a As Integer
    box = Array("日曜日", "月曜日", "火曜日", "水曜日", "木曜日", "金曜日", "土曜日")
    For a = 0 To 6
        Cells(1 + a, 1).Value = box(a)
    Next a
End Sub
```

○Array 関数

```
Dim 配列名 As Variant
配列名 = Array( )
```

Variant型は、データの型を指定せずにデータを格納できる変数で、Array関数を使うことで配列を格納することができる。Array関数のカッコ内ではコンマを使って区切ることで、複数の値を定義し、値を代入できる。

上記では、授業で行った簡単なプログラム例を挙げ、実施に授業する際の注意点等について示した。この後の授業では、以下の内容でプログラムの学習を進めていく予定であるがプログラム例は割愛させていただく。

5 __配列を使ったプログラム(応用)

6 __探索の手順とプログラム~逐次探索

7 __探索の手順とプログラム~二分探索

8 __整列の手順とプログラム

授業では生徒に興味を持たせるため、プログラムの内容を工夫し、その実行内容も様々なものを示している。次にその例を挙げる。



```
Sub 例①御唱和ください()  
Dim m As String, a As Integer  
m = InputBox("あなたの名前は?")  
MsgBox("御唱和ください")  
For a = 1 To 3  
    MsgBox(a)  
Next a  
MsgBox("ダー やったね " & m)  
End Sub  
Sub 例②合同式()  
Dim a As Integer, b As Integer, ans As Integer  
a = CInt(InputBox("最初の数は?"))  
b = CInt(InputBox("法とする数は?"))  
ans = a Mod b  
MsgBox(a & " ≡ " & ans & " (mod " & b & ")")  
End Sub  
Sub 例③評定()  
Dim a As Integer, hyoka As Integer  
a = CInt(InputBox("得点"))  
If a >= 75 Then  
    MsgBox("評定 5 やったね。")  
Else  
    If a >= 55 Then  
        MsgBox("評定 4 まあまあかな。")  
    Else  
        If a >= 35 Then  
            MsgBox("評定 3 普通です。")  
        Else  
            If a >= 30 Then  
                MsgBox("評定 2 頑張りましょう。")  
            Else  
                MsgBox("評定 1 ダメだ。コリャ。")  
            End If  
        End If  
    End If  
End If  
End Sub  
Sub 例④おそ松さん()  
Dim box As Variant, a As Integer, b As Integer  
box = Array("いません。", "おそ松", "カラ松", "チョロ松", "一松", "十四松", "トド松", "いません。")  
b = CInt(InputBox("六つ子のうち何番目?"))  
MsgBox("それは " & box(b) & " さんですね。")  
For a = b - 1 To b + 1  
    Cells(a + 1, 1).Value = box(a)  
Next a  
MsgBox(box(b) & " の前は " & box(b - 1) & vbCrLf & box(b) & " の後ろは " & box(b + 1) & vbCrLf & " ダヨーン")  
End Sub
```

5 おわりに

今回の実践事例で、プログラム指導初心者である私でも、マクロを用いてのプログラミングは比較的容易にできたように思う。ただ、マクロについては主なものを覚えていかなければならないが、慣れると比較的簡単にコードをかくことができる。

今後とも自己研鑽に勤めながら、しっかりと教材研究に励みたい。

参考 「教授資料【指導編 新・情報の科学】」（日本文教出版）