

2008 年 2 月 17 日改訂

VBA での教材・ツール作りのすすめ

～速い、安い、面白い～

尾崎文則（新南小学校）

2007 年の 2 学期から和歌山市の小学校では、生徒機にも Excel と Word を標準で搭載しています。これには VBA(Visual Basic for Applications)という便利なプログラミング機能が備わっています。VBA で教材を作れば、即、授業で使用可能なのです。教材作りの絶好の環境が整備できたといえます。そこで、取りあえずいくつかプログラミングをしてみたところ、アイデア次第で作りたいものが素早くできることを確かめることができました。試行してみた結果を報告します。(一部ワークシート関数のみ使ったものもあります。)

1. VBA での教材作りのメリットと注意点

- (ア) 「Visual Basic」が、Excel、Word の機能に加えて使えます。そのため、機能の高い作図ツールやワークシートのような入力インターフェイスも備えています。参考書も多く、ヘルプ、便利なデバッガも標準装備なので学習しやすい。
- (イ) ほとんどの Windows パソコンには Excel と Word を搭載しているので事実上開発環境が無料です(追加投資がいない)。しかも、今年度から和歌山市の小学校の生徒機にはもれなくインストールしているため教材配布も簡単です。教室など Internet のないオフライン環境でも使用できます。
- (ウ) 今ではパソコンの性能が向上しているため、VBA でも図形をコントロールしてアクションゲームを作れるぐらいビジュアルでインタラクティブなプログラムを作ることができます。生徒にもプログラムの面白さを伝えることが可能です。
- (エ) Excel2007 では機能の向上とともに仕様も一部変わったようなので、それ以前のバージョンとの互換性に少し問題があります。(Shape オブジェクト等)

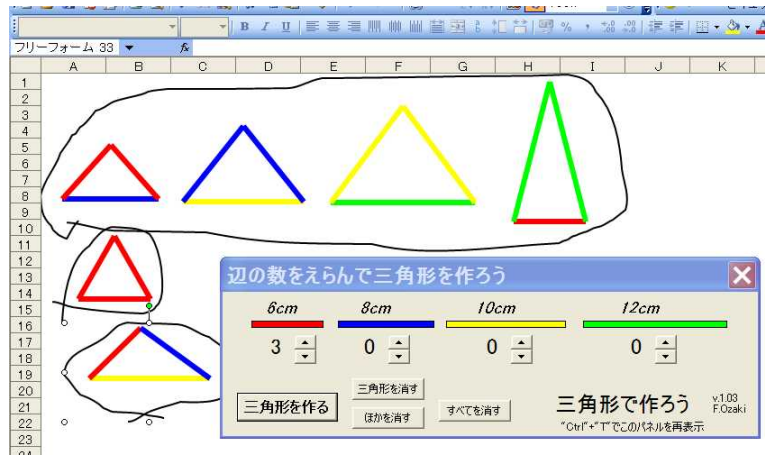
Excel2003 で作っていたものを Excel2007 で試してみたところうまく動きませんでした。どうも Excel2007 では図形関係等の仕様が変更になっているためにそのままでは同じように動かすことができないようです。(特に TextEffect プロパティや Group メソッドなど) 店頭の本に当たってもこの件についての明確な記述がありません。インターネットや MSDN、ヘルプにも詳細な記述を見つけることができません。ほとんど弱りましたが、なんとか試行錯誤で打開策を見つけました。とにかく図形関係は要注意です。Excel2007 では、図形関係のマクロコードが記録できない、というプログラム初心者にとってはひどい仕様になっています開発のためには Excel2003 以前のソフトはアンインストールしないで取っておいた方がよいと思います。

2. 「三角形で作ろう」: オートシェイプのグループ化、ユーザーフォーム

- (ア) 4 年生の三角形学習の導入に使えます。4 種類の辺から自由に三角形を描画することが出来ます。ここまではネット上にあるデジタルコンテンツと大差ありませんが、ExcelVBA が便

利なのは、作ったたくさんの三角形を Excel の機能を使って自由に移動、回転、図形の追加ができることです。しかも Excel2007 ならインクペン機能も使えるので作成した三角形を使って絵を作り上げることが出来ます。もちろん、こうやって三角形に慣れ親しんだ後、分類の学習にもこの機能は使えます。一人一人が何度も試行錯誤できるので学習効果も高くなる事が期待できます。

(イ) 以前はストローを使ってセロテープでつないで作っていました。しかし、クラスの一人一人に1セット分のストローを作るのはかなり大変。しかも繰り返し使うのは困難で使い捨て状態になります。「もったいないし、毎回作るのは大変だ」と思っていたものです。ExcelVBA は図形関係の教材作りには相性が良いようです。



3. 「円形名簿」: オートシェイプとテキスト、三角関数、Rnd 関数

(ア) いわゆる唐傘連判状のような円形の名簿が作れます。学級委員の投票などで PTA の方から要望されていたものです。そのときは、名簿からランダム関数を使って並べただけのものを使ってもらいました。しかし、これを使えばセル上に名簿データを貼り付けるだけでランダムな並び替えと円形に並べることをそれぞれワンクリックでできます。(Undo 機能も付けてみました。)

(イ) VBA を使うとこのように三角関数などを利用して、かなり自由に図形を操作することができます。しかもワークシートの機能も併用できるのでけっこう簡単にオリジナルのグラフ機能なども作れるはずですよ。

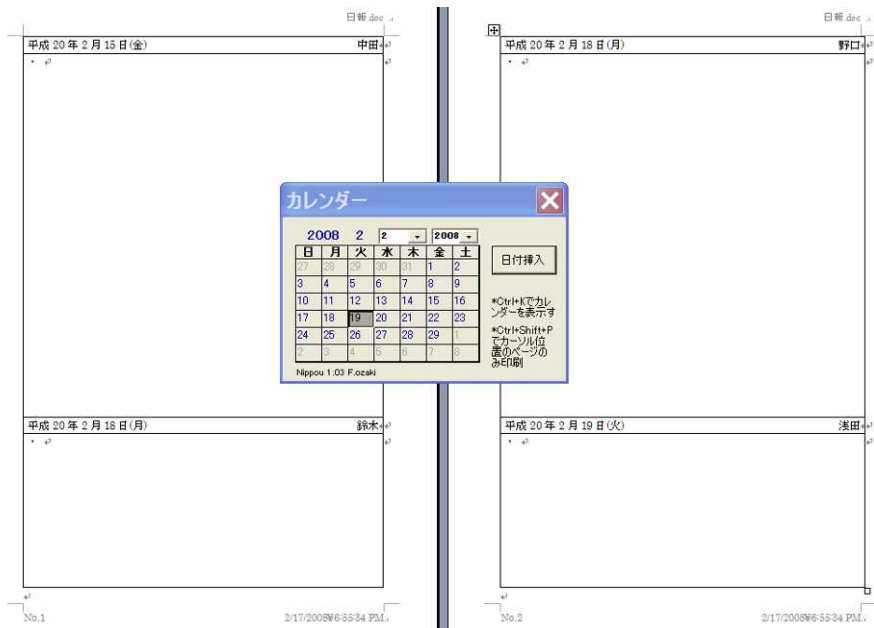
項目	名前
1	皇弥呼
2	聖徳太子
3	鑑真
4	藤原道長
5	柴 式部
6	清少納言
7	平 清盛
8	源 頼朝
9	源 義経
10	北条時宗
11	足利義満
12	足利義政
13	舟舟
14	フランシスコ・ザビエル
15	織田信長
16	豊臣秀吉
17	徳川家康
18	徳川家光
19	三井高利
20	近松門左衛門
21	歌川弘茂
22	本居宣長
23	杉田玄白
24	伊能忠敬
25	ペリー
26	勝 海舟
27	西郷隆盛
28	大久保利通
29	木戸孝允
30	明治天皇
31	福沢諭吉
32	大隈重信
33	板垣退助
34	伊藤博文
35	陸奥宗光
36	東郷平八郎

4. 「日報」: カレンダーコントロール、日付関数、クラスモジュール

(ア) Word でも VBA が使えるのでアイデアしだいで学習に使えるものができそうです。これは、Access のカレンダーコントロールを使っているの、生徒機では使えませんが、日付関数を使っても面白い

ものが作れるかもしれません。

(イ) ファイルを開くとその日の日時にまで自動でスクロールします。同時にカレンダーコントロールも表示して、カーソルで指定した場所にカレンダーを選ぶだけで日付を挿入でき



ます。簡単なクラスモジュールも使ってみました。

5. 「年間行事」「月間行事」: DateSerial、CHAR(10)

(ア) 「年間行事」は日付欄に自動的に縦に「日付け」と「曜日」を並ぶ所がちょっと工夫したところです。CHAR(10)が改行コードだということに気づくのがちょっとホネでした。

(イ) 「月間行事」はマクロコードで日付や休日の色塗りなどを自動化できるようにしてみました。DateSerial 関数をうまく利用して望む日付を求める方法は勉強になりました。

Q22		=DAY(AC22)&CHAR(10)&AP22						
A	B	C	D	E	F	G	H	
2008年度 年間行事予定表								
4月		5月		6月		7月		
1	火	1	木	1	火	1	火	
2	水	2	金	2	水	2	水	
3	木	3	土	3	木	3	木	
4	金	4	日	4	金	4	金	
5	土	5	月	5	土	5	土	
6	日	6	火	6	日	6	日	
7	月	7	水	7	月	7	月	
8	火	8	木	8	火	8	火	
9	水	9	金	9	水	9	水	
10	木	10	土	10	木	10	木	
11	金	11	日	11	金	11	金	

A	B	C	D	E	F	G	H	I	J	K	L
4月行事予定			平成19年度			月初め記入	日付を記入	土日を塗りつぶす	行の高さを揃える		
1	日	行事	保健関係	PTA	備考						
2	月	新任10:00 P役員と新任者の挨拶11:00 給食契約者への連絡									
3	火	初任研									
4	水	旧運営委員会8:30 職員会議8:45 給食関係者の打ち合わせ11時半 職員会議2時 送別会6時半									
5	木										
6	金	離任式8:55 P役員と転任者の挨拶、作業9:30 教科書わけ11時		旧常任委員会							
7	土	給食物資納入者会1:30～ 時間割編成 旧掃除場所									
8	日	職員会議選挙									
9	月	新任式、始業式8:55 入学式準備11:00 部会(午後) 登校指導(専、兼) 旧掃除場所		家庭環境調査表配布	農志						

6. 「百マス計算ワークシート」: RAND、RANK、INT、MOD

(ア) 百マス計算に関してはネット上にもいろいろなワークシート等が見つかりますが、余りのある割り算を入れたのはほとんどありません。ここは少しオリジナリティのあるところでは？ 正答欄には余り表記(…で)もするようにしています。INT 関数と MOD 関数を使うこと

で実現しています。このワークシートに関してはマクロコードを使っていません。

- (イ) 1分30秒近辺で自己記録と競い合う百マス計算は、現段階ではやはり紙メディアが一番取り組みやすいようです。答え合わせが慣れるまで面倒なので、同じデザインで正答も印刷できるようにしました。F9キーを押す度にランダムに問題を生成します。

回	月	日	分	秒	★F9キーを押す毎に問題を更新します。
わり算					解答
50問					名前:
1	60	÷	1	=	60...0
2	80	÷	6	=	13...2
3	57	÷	8	=	7...1
4	90	÷	3	=	30...0
5	75	÷	10	=	7...5
6	52	÷	5	=	10...2
26	63	÷	5	=	12...3
27	11	÷	4	=	2...3
28	96	÷	7	=	13...5
29	48	÷	2	=	24...0
30	60	÷	9	=	6...6
31	110	÷	6	=	18...2

7. せっかくの環境を生かしましょう

- (ア) ゲーム・クイズ・ドリル形式のものでも、ツールのなものでもたいいていのは VBA で作れるはず。(ポリゴンによる 3D 表現とかは無理ですが、透視図のようなものはできる) 情報交換しながら研究すればかなり進歩すると思います。

- (イ) 次の Web サイトにファイルを公開しています。ぜひ、ご検討の上、ご意見をいただけるとうれいす。以下には資料として一部のコードや関数の解説をつけました。

<http://www7b.biglobe.ne.jp/~ozaki/>

e-mail:f.ozaki@gmail.com

8. 資料「三角形で作ろう」マクロコード

- (ア) 標準モジュールのコード

Option Explicit

Dim A As Integer, B As Integer, C As Integer '辺の長さ

Dim colA As Integer, colB As Integer, colC As Integer '辺の色

Dim Hen As String, Hen2 As String, Hen3 As String, Hen4 As String '辺の名前を取得

Dim y2 As Single '垂線の長さ

'作図の原点の座標

Const xPos As Integer = 100

Const yPos As Integer = 200

変数と定数の型宣言と初期化

'色を指定する定数

Const Red As Integer = 2

Const Blu As Integer = 4

Const Yel As Integer = 5

Const Gre As Integer = 3

'長さを指定する定数：色別の長さ

```
Const leR As Integer = 60
Const leB As Integer = 80
Const leY As Integer = 100
Const leG As Integer = 120
```

'操作パネルを表示する

```
Sub Main()
    Edge.Show
End Sub
```

“操作パネル”を表示するサブルーチン。

“Edge”は”操作パネル”のオブジェクト名。

通常は Workbook を開いたときに同じスクリプトを Workbook オブジェクトに記述して起動する。

'辺の種類と数を判断する

```
Sub CreTri()
    Select Case Edge.R
        Case 3
            Call HenInit(leR, leR, leR, Red, Red, Red)
            Call Sankaku(A, B, C)
        Case 2
            If Edge.B = 1 Then
                Call HenInit(leB, leR, leR, Blu, Red, Red)
                Call Sankaku(A, B, C)
            ElseIf Edge.Y = 1 Then
                Call HenInit(leY, leR, leR, Yel, Red, Red)
                Call Sankaku(A, B, C)
            ElseIf Edge.G = 1 Then
                MsgBox "三角形になりません！"
            End If
        Case 1
            If Edge.B = 2 Then
                Call HenInit(leR, leB, leB, Red, Blu, Blu)
                Call Sankaku(A, B, C)
            ElseIf Edge.Y = 2 Then
                Call HenInit(leR, leY, leY, Red, Yel, Yel)
                Call Sankaku(A, B, C)
            ElseIf Edge.G = 2 Then
                A = leR
                B = leG
                C = leG
            End If
    End Select
End Sub
```

“操作パネル”の”三角形を作る”ボタンをクリックしたときに呼び出すサブルーチン。どんな三角形を描くか判断する。

三角形の描画に使う 4 種類の辺の数を以下の 4 個の変数に代入してどんな三角形かを判断している。

Edge.R は赤色 6 cm

Edge.B は青色 8 cm

Edge.Y は黄色 10 cm

Edge.G は緑色 12 cm

それぞれ Edge オブジェクトの変数 R、B、Y、G を指す。HenInit() は辺の色と長さを決定する自作関数。

Sankaku() は実際に三角形を描画する自作関数。

```
colA = Red
colB = Gre
colC = Gre
Call HenInit(leR, leG, leG, Red, Gre, Gre)
Call Sankaku(A, B, C)
Elseif Edge.B = 1 Then
  If Edge.Y = 1 Then
    Call HenInit(leY, leR, leB, Yel, Red, Blu)
    Call Sankaku(A, B, C)
  Else
    Call HenInit(leG, leR, leB, Gre, Red, Blu)
    Call Sankaku(A, B, C)
  End If
Elseif Edge.Y = 1 Then
  Call HenInit(leG, leR, leY, Gre, Red, Yel)
  Call Sankaku(A, B, C)
End If
Case 0
Select Case Edge.B
  Case 3
    Call HenInit(leB, leB, leB, Blu, Blu, Blu)
    Call Sankaku(A, B, C)
  Case 2
    If Edge.Y = 1 Then
      Call HenInit(leY, leB, leB, Yel, Blu, Blu)
      Call Sankaku(A, B, C)
    Elseif Edge.G = 1 Then
      Call HenInit(leG, leB, leB, Gre, Blu, Blu)
      Call Sankaku(A, B, C)
    End If
  Case 1
    If Edge.Y = 2 Then
      Call HenInit(leB, leY, leY, Blu, Yel, Yel)
      Call Sankaku(A, B, C)
    Elseif Edge.G = 2 Then
      Call HenInit(leB, leG, leG, Blu, Gre, Gre)
      Call Sankaku(A, B, C)
    Elseif Edge.Y = 1 Then
      Call HenInit(leG, leB, leY, Gre, Blu, Yel)
```

```

        Call Sankaku(A, B, C)
    End If
Case 0
    If Edge.Y = 3 Then
        Call HenInit(leY, leY, leY, Yel, Yel, Yel)
        Call Sankaku(A, B, C)
    ElseIf Edge.Y = 2 Then
        Call HenInit(leG, leY, leY, Gre, Yel, Yel)
        Call Sankaku(A, B, C)
    ElseIf Edge.Y = 1 Then
        Call HenInit(leY, leG, leG, Yel, Gre, Gre)
        Call Sankaku(A, B, C)
    Else
        Call HenInit(leG, leG, leG, Gre, Gre, Gre)
        Call Sankaku(A, B, C)
    End If
End Select
End Select
End Sub

```

'辺の長さと色を設定する

```

Sub HenInit(le1, le2, le3, col1, col2, col3)
    A = le1 '底辺の長さ
    B = le2
    C = le3
    colA = col1 '底辺の色
    colB = col2
    colC = col3
End Sub

```

辺の長さと色を設定する自作関数。

'二等辺三角形を描画する(使用していない)

```

Sub Sankaku2(A, B, C)
    Call Sen(xPos, yPos, xPos + A, yPos, colA)
    Selection.Name = "hen"
    y2 = Sqr(B ^ 2 - (A / 2) ^ 2)

    Call Sen(xPos, yPos, xPos + A / 2, yPos - y2, colB)
    Selection.Name = "hen2"
    ActiveSheet.Shapes.Range(Array("hen", "hen2")).Select

```

```
Selection.ShapeRange.Group.Select
Selection.Name = "hen"
```

```
Call Sen(xPos + A, yPos, xPos + A / 2, yPos - y2, colC)
Selection.Name = "hen2"
ActiveSheet.Shapes.Range(Array("hen", "hen2")).Select
Selection.ShapeRange.Group.Select
Selection.Name = "Tri"
```

DoEvents

End Sub

'三角形を描画する。Excel2007 では同じ名前があると Group 化できないのですべてのオブジェクトに
違う名前をつける

Sub Sankaku(A, B, C)

```
Call Sen(xPos, yPos, xPos + A, yPos, colA) '底辺の描画
Hen = Selection.Name 'A 辺の名前を Hen に代入
```

```
y2 = Sqr(B ^ 2 - (A * (B / (B + C))) ^ 2) '垂線の長さを算出
```

```
Call Sen(xPos, yPos, xPos + (A * (B / (B + C))), yPos - y2, colB) '左斜辺の描画
Hen2 = Selection.Name 'B 辺の名前を Hen2 に代入
ActiveSheet.Shapes.Range(Array(Hen, Hen2)).Select 'A 辺と B 辺をグループ化する
Selection.ShapeRange.Group.Select
Hen3 = Selection.Name 'AB 辺をグループ化したものの名前を Hen3 に代入
```

```
Call Sen(xPos + A, yPos, xPos + (A * (B / (B + C))), yPos - y2, colC) '右斜辺の描画
Hen4 = Selection.Name 'C 辺の名前を Hen4 に代入
ActiveSheet.Shapes.Range(Array(Hen3, Hen4)).Select 'AB 辺と C 辺をグループ化する
Selection.ShapeRange.Group.Select
Selection.Name = "Tri" 'グループ化した三角形に「Tri」と名前をつける
```

End Sub

'直線を描画する

Sub Sen(x1, y1, x2, y2, col)

```
ActiveSheet.Shapes.AddLine(x1, y1, x2, y2).Select
```

```
Selection.ShapeRange.Line.Weight = 4.5
```

```
Selection.ShapeRange.Line.ForeColor.SchemeColor = col
```

黒 0&8、白 1&9、黄緑 3&11、桃 6&14、水 7&15

三角形を描画する自作関数。
描画した三角形をグループ
化して名前を付けることで、
その他の図形と区別して操
作できる。

直線を描画する自作関数。

'線を引く

'線の太さ

'線の色: 青 12&4、赤 10&2、黄 5&13、

End Sub

(イ) 操作パネルオブジェクト(”Edge”)のコード

Public R As Integer, B As Integer, Y As Integer, G As Integer

Private Sub AllDel_Click() '全てを消す

 If MsgBox("本当に消してもいいですか?", vbOKCancel) = vbOK Then

 For Each s In ActiveSheet.Shapes

 s.Delete

 Next

 End If

 DoEvents

End Sub

Private Sub Cre_Click() '三角形を作る

 R = CInt(Label1.Caption) '色辺別の数を取得

 B = CInt(Label2.Caption)

 Y = CInt(Label3.Caption)

 G = CInt(Label4.Caption)

 If R + B + Y + G = 3 Then '辺の合計が3なら三角形の描画を実行、違うならメッセージを出す

 Call CreTri

 Else

 MsgBox ("辺の数の合計を3にしてください")

 End If

End Sub

標準モジュールの”CreTri”サブルーチンを実行させる。

Private Sub Del_Click() '三角形だけを消す

 On Error GoTo Error1 '三角形が書かれていない場合のエラー処理

 ActiveSheet.Shapes("Tri").Delete

 Exit Sub

Error1:

 MsgBox ("三角形がありません")

End Sub

描画した三角形には”Tri”という名前を付けているので、他の図形と区別して操作できる。

Private Sub OtrDel_Click() '三角形以外の図形を消す

 If MsgBox("本当に消してもいいですか?", vbOKCancel) = vbOK Then

 For Each s In ActiveSheet.Shapes

 s.Select

```
        If Selection.Name <> "Tri" Then
            Selection.Delete
        End If
    Next
End If
End Sub

Private Sub SpinButton1_Change()    'スピンの値をラベルに表示する
    Label1.Caption = SpinButton1.Value
End Sub

Private Sub SpinButton2_Change()
    Label2.Caption = SpinButton2.Value
End Sub

Private Sub SpinButton3_Change()
    Label3.Caption = SpinButton3.Value
End Sub

Private Sub SpinButton4_Change()
    Label4.Caption = SpinButton4.Value
End Sub
```

9. 資料「円形名簿」マクロコード（Excel2003 以前）

(ア) 標準モジュール"Kasa"

Option Explicit

Sub Kasa()

'氏名データを順にワードアートに代入し円形に並べる

'変数の型宣言

Dim Ninzu As Integer	'データの件数
Dim KakuInc As Double	'角度の増分
Dim Kaku As Double	'そのデータの回転角度
Dim Hankei As Integer	'文字配置の元になる円形の半径
Dim i As Integer	'カウンター
Dim Namae As String	'氏名データ
Dim Komoku As String	'項目データ
Dim rad As Double	'ラジアン
Dim x As Double	'ワードアートのX座標
Dim y As Double	'ワードアートのY座標

```

Dim xPos As Single      ' 中心位置を加えた X 座標
Dim yPos As Single      ' 中心位置を加えた Y 座標
Dim cell_x As Integer   ' 氏名欄のセルの Row 値
Dim Gosu As Integer     ' 氏名の語数

On Error GoTo Err1      ' 名前が入力していない場合のエラー処理

Range("M5", Range("M5").End(xlDown)).Select      ' 入力した件数に応じてセルを選択する
Ninzu = Selection.Count                          ' 氏名データの件数を数えて代入
KakuInc = 360 / Ninzu                            ' 角度の増分を算出
cell_x = 5                                        ' 氏名欄の最初のセルの Row 値
Kaku = 0                                          ' 角度変数の初期化
Hankei = Range("I6")                             ' 半径の初期化

For i = 1 To Ninzu
    Namae = Cells(cell_x, 13)                    ' セルの値を順に Namae 変数に代入
    Komoku = Cells(cell_x, 14)

    Gosu = Len(Namae + Komoku)                    ' 氏名の語数
    Hankei = Hankei + (Gosu - 7) * 6              ' 語数に合わせて半径を増やす

    rad = Kaku * 3.1415 / 180                    ' ラジアンを算出
    x = Hankei * Math.Cos(rad)                   ' ワードアートの元の x 座標
    y = Hankei * Math.Sin(rad)                   ' ワードアートの元の y 座標

    xPos = x + 190                               ' 中心をずらしたワードアートの x 座標
    yPos = y + 300                               ' 中心をずらしたワードアートの y 座標

    xPos = xPos - (Gosu - 7) * 6                 ' 標準を 7 個として語数の差に
    定数 6 を掛けて x 座標をずらす

    ActiveSheet.Shapes.AddTextEffect(msoTextEffect1, Namae & " " & Komoku, "MS 明朝",
12#, _
        msoFalse, msoFalse, xPos, yPos).Select    ' ワードアートの描画
    Selection.ShapeRange.Fill.ForeColor.SchemeColor = 8 ' 文字の色を設定
    Selection.ShapeRange.Line.Visible = msoFalse   ' 文字の縁取りをしない
    ' Selection.ShapeRange.TextEffect.ToggleVerticalText
    Selection.ShapeRange.IncrementRotation Kaku    ' 回転角度の増分

```

```

    Kaku = Kaku + KakuInc           '角度を増やす
    cell_x = cell_x + 1           'セルの Row 値を増やす
    Hankei = Range("I6")         '半径の初期化
    DoEvents                       '画面を描画し直す
Next

Range("M5").Select
Exit Sub

```

Err1:

```

    MsgBox ("名前を入力して下さい")
    Range("M5").Select
End Sub

```

(イ) 標準モジュール”Rnd”

Option Explicit

```

Dim Moto(50, 3) As String         '並び替え前のリストを入れる配列変数
Dim Namae(50, 3) As String       '氏名データを入れる配列変数
Dim Ninzu As Integer             '氏名の件数を入れる変数
Dim i As Integer                 'カウンター
Dim j As Integer                 'セルの Row 値を入れる変数
Dim temp1 As String              '一時保存
Dim temp2 As String
Dim r As Integer                 'ランダムに生成した整数値

```

Sub Rnd()

```

'氏名欄に入力された氏名をランダムに並び替える
Math.Randomize                   '乱数の種を初期化
'Range("M5", Range("M5").End(xlDown)).Select '入力した件数に応じてセルを選択する
'Ninzu = Selection.count         '選択したデータのあるセルの個数を数え変数に代入
j = 5                             '一番上の氏名欄の Row 値

```

```

Call Hikae                       'Hikae を呼び出す

```

```

For i = 1 To Ninzu               'Namae 配列変数の氏名データをランダムに並び替える
    r = Int(Ninzu * Math.Rnd + 1)
    temp1 = Namae(i, 1)
    Namae(i, 1) = Namae(r, 1)
    Namae(r, 1) = temp1

```

```

        temp2 = Namae(i, 2)
        Namae(i, 2) = Namae(r, 2)
        Namae(r, 2) = temp2
    Next

    j = 5                                ' 氏名欄のセルの Row 値を初期化
    For i = 1 To Ninzu                    ' ランダム化した配列変数の値を氏名欄に書き出す
        Cells(j, 13) = Namae(i, 1)
        Cells(j, 14) = Namae(i, 2)
        j = j + 1
    Next
    Range("M5").Select
End Sub

Sub 元にもどす()
    j = 5                                ' 一番上の氏名欄の Row 値
    For i = 1 To Ninzu
        Cells(j, 13) = Moto(i, 1)        ' 元のリストをセルに書きこむ
        Cells(j, 14) = Moto(i, 2)
        j = j + 1
    Next
End Sub

Sub クリア()
    ' 氏名欄のデータを消去する
    Call Hikae

    Range("M5", Range("N5").End(xlDown)).Select
    Selection.Clear                       ' 値のみ消去は ClearContents
    Range("M5").Select
End Sub

Sub Hikae()
    ' 氏名欄のデータを Moto 配列変数に控えをとる
    Range("M5", Range("M5").End(xlDown)).Select    ' 入力した件数に応じてセルを選択する
    Ninzu = Selection.count                        ' 選択したデータのあるセルの個数を数え変数に代入
    j = 5
    For i = 1 To Ninzu                            ' Namae 配列変数に氏名データを代入

```

```

        Namae(i, 1) = Cells(j, 13)
        Namae(i, 2) = Cells(j, 14)
        j = j + 1
    Next

```

```

For i = 1 To Ninzu
    Moto(i, 1) = Namae(i, 1)
    Moto(i, 2) = Namae(i, 2)
Next
End Sub

```

(ウ) 標準モジュール”Sakujo”

Option Explicit

Sub ワードアート削除()

```

    Dim s As Object
    'オブジェクト変数
    '描画オブジェクトの内、ワードアートオブジェクトを全て順に選択して削除する
    For Each s In ActiveSheet.Shapes
        If s.Type = msoTextEffect Then s.Delete
    Next
    Range("M5").Select
End Sub

```

10. 資料「円形名簿」マクロコード (Excel2007)

(ア) ワードアートで描画せず四角形を描いてその中にテキストを代入する

```

For i = 1 To Ninzu
    Namae = Cells(cell_x, 13)
    Komoku = Cells(cell_x, 14)

    rad = Kaku * 3.1415 / 180
    x = Hankei * Math.Cos(rad)
    y = Hankei * Math.Sin(rad)

    xPos = x + 165
    yPos = y + 300

```

Shape オブジェクトが機能向上しているのでテキストボックスを回転させるとテキストも回転するようになった。また左上を中心に回転するので位置補正がいらぬ。

```

ActiveSheet.Shapes.AddShape(msoShapeRectangle, xPos, yPos, 150, 20).Select
Selection.ShapeRange.TextEffect.Text = Namae & " " & Komoku
Selection.ShapeRange.Fill.Visible = msoFalse

```

```

Selection.ShapeRange.Line.Visible = msoFalse      '四角形の枠を消す
Selection.Characters.Font.ColorIndex = 0          '文字の色の設定
Selection.HorizontalAlignment = xlHAlignLeft      '文字列を左に寄せる
Selection.ShapeRange.IncrementRotation Kaku      '回転角度

Kaku = Kaku + KakuInc                             '角度を増やす
cell_x = cell_x + 1                               'セルの Row 値を増やす
Hankei = Range("I6")                             '半径の初期化
DoEvents                                          '画面を描画し直す
Next

```

11. 資料「日報」マクロコード

(ア) “ThisDocument”オブジェクト

```
Private Sub Document_Open()
```

```
Selection.Find.ClearFormatting
```

```
With Selection.Find
```

```
.Text = "年" & Str(Month(Now())) & "月" & Str(Day(Now())) & "日"
```

```
.Replacement.Text = ""
```

```
.Forward = True
```

```
End With
```

```
Selection.Find.Execute
```

```
Dim Cal As New Class1
```

```
Cal.Calendar
```

```
End Sub
```

ファイルを開いた時に今日の日付が入った文字を検索して選択することで該当ページへジャンプする。

Class1 クラスから Cal インスタンスを生成し、Calendar メソッドを実行する。

(イ) クラスモジュール”Class1”

```
Sub Calendar()
```

```
CalendarForm.Show                                'カレンダーコントロールを起動する
```

```
CalendarForm.Calendar1.Day = Day(Now())          'カレンダーのデフォルトの日付を本日にする
```

```
CalendarForm.Calendar1.Month = Month(Now())
```

```
CalendarForm.Calendar1.Year = Year(Now())
```

```
End Sub
```

(ウ) 標準モジュール”PrintCurrentPage”

```
Sub PrintCurrentPage()
```

```
Application.PrintOut FileName:="", Range:=wdPrintCurrentPage 'カーソルのあるページのみを印刷する
```

```
End Sub
```

12. 資料「月間行事」「年間行事」

(ア) 標準モジュールサブルーチン”DateFill”

Sub DateFill() '1日の日付 Serial データを元に月末までの日付を記入する。

```
n = Range("A3")
nYear = Year(n)
nMonth = Month(n)
nDay = Day(n)
m = DateSerial(nYear, nMonth + 1, nDay)
maxDate = Day(m - 1)
Range("A3").Select
For i = 2 To maxDate
    ActiveCell.Offset(1, 0).Select
    n = n + 1
    ActiveCell = n
Next
End Sub
```

A3セルの日付 Serial からいったん年、月、日のデータを取り出し、月だけを加算してよく月を求め、その上で DateSerial で Serial データに戻す。そこから 1 を減算して日付を取り出し、当該月の月末の日付を求める。この数値を元に大の月、小の月を判断して月間行事に日付を記入していく。

(イ) 「年間行事」: 関数による年間行事表

西暦から各月の日付と曜日の一覧表を自動的に作成するように関数を設定します。その一覧表を参照することで年間行事表の日付セルに日付と曜日を同時に表示するようにしています。二つのデータを縦に配置するには間で改行をする必要があります。それは、=DAY(AA3)&CHAR(10)&AB3 というように 2 つのセル参照を&で CHARA(10)を結び付けることで実現しています。CHARA(10)は改行コードです。

13. 資料「百マス計算ワークシート」: ワークシート関数

(ア) VBA を使わない方法

VBA で書いた方が簡単なような気もしたのですが、参考にした本のワークシート関数の使い方が面白かったのでまねさせていただきました。セルに RAND()関数を並べて RANK()関数で順位付けして 0 から 10 までの乱数を得ています。何でも VBA ではなく開発効率の良い方法を組み合わせればよいでしょう。

(イ) 余りのある割り算

この手のシートで余りのある割り算があるものは見つかりませんでした。やっていることは簡単です。=INT(M4/O4)&"..."&MOD(M4,O4)として INT()関数で商の整数部分を取り出し、MOD()関数で余りを求めて結び付けているだけです。

*注: VBA では、モジュールごとに自動的にコードを分散して記述するため、書くときは簡便なのですが、総合的に読む場合は全体を捉えにくい所があります。思わぬ所にコードが隠れている場合があります。ファイルからマクロコードを読んでくださる場合は各モジュールを全てチェックしてみてください。