# 席替えアプリの作成

## アプリの使い方

　Excel VBAの応用例として席替えアプリを開発しました。本章ではそのプログラムを読んでみましょう。

　席替えアプリは生徒を乱数で配置するだけでなく、以下の機能があります。

* 特別席を設定する。特別席に配置すべき生徒を指定することができる。
* 席をお互いに離す生徒を指定できる

　

図8.1　名簿の例（前半部分）

　名簿の例を図8.1に示します。DE列はプログラムが使用するので空けておきます。F列より右側が条件を指定する場所です。

　特別席に配置すべき生徒の場所に1を入れます。

　お互いの席同士を離す生徒のペアに1を入れます。席何個分離すかは別途指定します。この例では、1006加藤、1013須藤、1017千田の3名はお互いに離します。また、1006加藤と1011佐藤も離します。左から順番に見ていって「離す」のペアが見つからない列を発見した場合、そこから右側は無視します。



図8.2　席配置シートに席を配置したところ

　席は手作業で配置します。配置後の例を図8.2に示します。席は4種類あり、「男子普通席」「女子普通席」「男子特別席」「女子特別席」です。セルの範囲を指定した後、左下の8個のボタンのいずれかを押すことで、その席の属性を設定します。男女の並びは1つ1つ設定するのは大変なので「市松模様」「縞模様」「性別反転」のボタンを使ってだいたいの指定をした後、1個ずつ手作業で設定します。

　席を配置できる場所は3～11行目、A～H列の範囲です。最大で8×8=64個の席を配置できます。

　「席を離したい生徒同士の席を最低席何個分離すのか」「どの名簿シートを使うのか」をそれぞれH1, J1に記入します。席間距離は図8.3のように数えます。



図8.3　席間距離の数え方

　配置が終わったら、「席数確認」のボタンを押して、配置した座席数と名簿の人数を比較します。結果が右上のテキストボックス内に表示されます。以下の4つの条件を満たしていることを確認します。

* 男子用の席数（普通席＋特別席）＝ 男子生徒の人数
* 女子用の席数（普通席＋特別席）＝ 女子生徒の人数
* 男子用特別席の数 ≧ 特別席に配置する男子生徒の人数
* 女子用特別席の数 ≧ 特別席に配置する女子生徒の人数

　上の4条件を満たさない場合は、席を修正した後、もう一度「席数確認」を押します。条件を満たすまで繰り返します。

　条件を満たしたら、「席替え」のボタンを押して下さい。席替えが完了します。条件が厳しすぎて満たせなかった場合は「10回試行しましたが条件を満足できませんでした」と表示して終了します。離す生徒の最小席間距離を小さくするか、手作業で並べ換えて下さい。

　上下逆の席配置表が必要なときは、シート「上下逆」を選択し、「作成」ボタンを押して下さい。

　では、プログラムの内部を見ていきましょう。

## 定数

　プログラムの先頭を見て下さい。VBA Editorで一番上までスクロールするか、右上のリストボックスを操作して (Declarations) を選択して下さい。

　Option Explicitと書いています。これにより変数の宣言を強制します。今回のようなある程度長いプログラムを書くときは、変数名のタイプミスによるバグを避けるため、Option Explicitを書いた方がよいです。その後に、定数を定義しています。「全てのプロシージャで共有すべき値」かつ「更新する必要がない値」は、定数として宣言します。Constをつけると定数となります。Constで宣言した変数にプロシージャ中で値を代入すると、エラーが発生します。

　座席の色、シート「席配置」において使用する範囲、名簿シートにおける列の役割、男女を指定するための文字などを指定します。

## 関数yrand

　最初に書いてあるのが関数yrand(i, j) です。Excelのワークシート関数Worksheetfunction.Randbetween(i, j) は発生させる乱数に偏りがあるように思います。乱数の質が悪いように思われるので、同等の関数をyrand(i, j) という名前で作成し、こちらを使います。

　1～nまでの乱数を得るための数式は以下のように書きます。

 1 + Int(Rnd() \* n)

　Rnd() は0～0.999999...の範囲の乱数を発生させる関数です。これにnをかけると0～(n×0.99999...) の範囲の乱数になります。上限はn未満で最大の数です。Int関数を使って整数部のみを取り出すと0～(n-1)の範囲の整数になります。これに1を足すと1～nまでの乱数となります。ちょっとトリッキーですが、1～nの乱数を得るときの定番表現です。

　i～jまでの乱数を得てyrandに入れるには以下のように書きます。

 n = j - i + 1

 yrand = i + Int(Rnd() \* n)

　yrandの最初にRandomizeという命令を引数なしで使っています。引数なしでRandomizeを呼ぶと、システムタイマーの値を使って疑似乱数の種を初期化します。yrandを呼ぶたびにシステムタイマーの値で初期化するので、乱数の質が向上します。

## 席の属性を設定するプロシージャ

　set\_male\_normal() 以降は、席の属性（塗りつぶし色）を設定するためのプロシージャです。

 Selection.Interior.color = 色を表す数値（定数）

と書くことで、色を設定します。「男子普通席」（プロシージャ名set\_male\_normal()）「女子普通席」（set\_female\_mormal()）の場合は単色で塗りつぶしますが、「特別席」seat\_special()）の場合、

* 男子普通席→男子特別席
* 女子普通席→女子特別席
* それ以外の席（男子特別席と女子特別席）はそのまま

という処理を行います。「普通席」（seat\_normal()）はその逆です。seat\_normal() seat\_special() はオブジェクト変数を使ったFor Eachを使っています。この技は○.○節で学習しました。ここではDim c As Rangeとしていますが、Dim c As Objectでもプログラムは問題なく動きますDim c As Rangeとすると、Range以外のオブジェクトを代入するとエラーになります。この方がプログラムの誤りに気がつきやすいのでRange型として宣言しています。

　「性別反転」（invert\_sex()）は

* 男子普通席→女子普通席
* 男子特別席→女子特別席
* 女子普通席→男子普通席
* 女子特別席→男子特別席

という変換を行います。

市松模様にするプロシージャ

　プログラムを以下に示します。

 istart = Selection(1).row

 iend = Selection(Selection.count).row

 jstart = Selection(1).column

 jend = Selection(Selection.count).column

 For i = istart To iend

 sex = i Mod 2

 For j = jstart To jend

 If sex = 0 Then

 Call set\_seat\_male(i, j)

 Else

 Call set\_seat\_female(i, j)

 End If

 sex = (sex + 1) Mod 2

 Next j

 Next i

　Selection(1)は選択した選択領域の左上のセルを表すオブジェクトです。Selection(1).rowで行番号を取り出し、Selection(1).columnで列番号を取り出します。

　Selection.countは選択したセルの個数です。Selection(Selection.count)は選択領域の右下のセルです。

　選んだ範囲の最初の行をistart, 最後の行をiendに入れ、左端列をjstart, 右端列をjendに入れます。

　i Mod 2はiを2で割ったときの余りをとります。sex は0か1です。左端列は0か1が交互に並びます。そこから右へ向かっては、sex = (sex + 1) Mod 2を実行するので、sexは0, 1, 0, 1, 0, .... のように0と1を交互にとります。すなわち男女の席が交互に配置されます。

　このプロシージャから、set\_seat\_male(i, j) とset\_seat\_female(i, j) を呼びます。set\_seat\_male(i, j) はi行j列の席を男子席にします。以下の変換を行います。

* 女子普通席 → 男子普通席
* 女子特別席 → 男子特別席
* それ以外の席（男子特別席と女子特別席）はそのまま

　set\_seat\_female(i, j) も同様です。

## シートの確認

　kakunin() はシート「席配置」の席が、J1で指定された名簿シートの席替えに使用できるか否かを確認するためのプロシージャです。内部でプロシージャcount\_seat() を呼び出しています。変数smeiboに名簿のシート名を入れてcount\_seat() を呼ぶと、シート「席配置」と名簿シートsmeiboにおいて、以下の4つの個数をカウントして返します。名簿シートにおいては「個数」を「人数」に置き換えて下さい。

* 男子席の個数（普通席＋特別席）
* 女子席の個数（普通席＋特別席）
* 男子特別席の個数
* 女子特別席の個数

　count\_seat() においては、指定された名簿シートがなかったときにエラーを出す処理があります。以下の部分です。

 find = 0

 For i = 1 To Worksheets.count

 If smeibo = Sheets(i).name Then

 find = 1

 Exit For

 End If

 Next i

 If find = 0 Then

 MsgBox ("シート「" & smeibo & "」がありません。")

 End

 End If

　ここで変数findは「フラグ変数」と呼ばれます。フラグ (flag) は「旗」の意味です。フラグ変数はyesかnoの二者択一の値をとります。この場合は、指定された名前のシートが見つかったとき1、そうでないとき0です。

　最初に0を入れています。Worksheets.countはxlsmファイルが含むワークシートの個数です。各シートは Sheet(1) ～ Sheets(Worksheet.count) でアクセスできます。シートの名前はSheets(i).nameで得られます。3行目のIf文は「smeiboで与えられたシートが見つかったなら、findに1を入れ、Forループから脱出する」ことを表しています。Forループを終わった時点でfind = 0ならシートが見つからなかったことを意味します。そのときは、シートが見つからないというメッセージボックスを表示してプログラムの実行を停止します。Endはプログラムの実行を停止する命令です。

　次のget\_last\_rowは「シート名」「列名」を指定して呼ぶと、指定されたシートの指定された列において、データが入っている最後の行の番号を返します。ここではB列を指定していますが、A列, C列でも構いません。わずか1行のプロシージャなので、わざわざプロシージャとして独立させる必要はないように思われますが、call get\_last\_rowと書く方が処理内容が明示されて分かりやすいと思います。

　席数や人数をカウントした結果をテキストボックスに表示します。テキストボックスに表示するのは以下の部分です。

 Sheets(shaiti).Shapes.Range(Array("TextBox 1")). \_

 TextFrame2.TextRange.Characters.text = text

　1行目の行末の \_ は継続行を表します。次の行に続いていることを示します。この部分は、自分で「マクロの記録」を実行し、テキストボックスの中に文字を書き込んでみて、命令の書き方を取得しました。テキストボックスの中に文字 "abc" を書き込む行為を「マクロの記録」を行うと以下のようになります。

 ActiveSheet.Shapes.Range(Array("TextBox 1")).Select

 Selection.ShapeRange(1).TextFrame2.TextRange.Characters.text = "abc"

　　　　　　　　（以下略）

　ここから必要な部分のみを取りだしました。

## シートのクリア

　clear\_nameはシート「席配置」の名前、名簿シートの「行, 列」の列、メッセージ表示用テキストボックスの中をクリアするプロシージャです。特に説明を要する部分はないでしょう。

## 席替えの実行

　名簿シート名の取得、シートのクリアを行った後、まず「男子特別席」に対して以下の処理を行います。

1. 男子特別席の席数を数える。

2. 男子で特別席に座る人を男子用特別席に配置する。

　プログラムは以下のようになっています。

 n = 0

 For i = imin To imax

 For j = jmin To jmax

 If Cells(i, j).Interior.color = c\_male\_special Then

 n = n + 1

 row(n) = i

 column(n) = j

 End If

 Next j

 Next i

 For i = 1 To last

 If Sheets(smeibo).Range(csex & i) = male\_mark And \_

 Sheets(smeibo).Range(cspecial & i) = 1 Then

 If n <= 0 Then

 MsgBox ("男子特別席の数が不足しています")

 Exit Sub

 End If

 name = Sheets(smeibo).Range(cname & i)

 r = yrand(1, n)

 Cells(row(r), column(r)) = name

 Sheets(smeibo).Range(ci & i) = row(r)

 Sheets(smeibo).Range(cj & i) = column(r)

 n = n - 1

 For j = r To n

 row(j) = row(j + 1)

 column(j) = column(j + 1)

 Next j

 End If

 Next i

　nは特別席の席数をカウントするための変数です。最初の二重ループは席が存在する範囲をサーチし、男子特別席を見つけたなら、その行と列をrow(n), column(n) に入れます。二重ループが終わった時点で、変数の内容は以下のようになっています。nは男子特別席の個数です。

|  |  |  |
| --- | --- | --- |
| 添字 | row( ) | column( ) |
| 1 | 1個目の男子特別席の行番号 | 1個目の男子特別席の列番号 |
| 2 | 2個目の男子特別席の行番号 | 2個目の男子特別席の列番号 |
| --- | ---- | ---- |
| n | n個目の男子特別席の行番号 | n個目の男子特別席の列番号 |

　次に一人ずつ特別席に配置してゆきます。名簿を上から順にみてゆき、男子で特別席に配置すべき生徒を見つけたなら、以下の処理を行います。

　特別席の残りがn個あるとき、1～nの乱数を発生させて、それがrであったなら、row(r), column(r)のセルにその生徒を配置します。次にnの値を1減らし、row( ), column( ) の添字r以降の値を、一つ上に詰めます。この処理を行っているのが、プログラムのn = n - 1より下の部分です。

　男子特別席に配置すべき生徒の処理が終わった後、女子特別席に配置すべき女子、残りの男子、残りの女子、に対しても同様の処理を行います。

## 席の交換

　前節での配置では離すべき生徒のことは考慮しませんでした。一旦、ランダムに席を配置した後、離すべき生徒同士が、条件を満たしているか否かを調べます。条件を満たしていない場合は、どちらかの生徒を別の生徒と交換します。

　この操作を条件が満たされるまで繰り返します。その処理を行うのがプロシージャrelocate() です。

　最初にmake\_work\_sheetを呼びます。make\_work\_sheetはシート「work」に離すべき生徒のリストを作成します。リスト作成後、シートworkは図8.4のようになります。



図8.4　シート「work」の初期状態

　A列は条件の番号です。1から順番に大きくなります。B列とC列に離すべき2人の生徒が、名簿シートの何行目かを入れます。この例では、離すべき生徒として、名簿シートの7, 14, 18行目に1が入っていました。3人以上に1が入っている場合、そのうち2人をピックアップした場合のあらゆる組み合わせに対して、指定距離以上の席間距離を確保します。7, 14, 18の3人の場合、「7と14」「7と18」「14と18」の3個の条件に分解します。4人なら、6個の条件に分解します。この処理をmake\_work\_sheetで行います。

　アルゴリズムは以下の通りです。

1. 1つの列を上から下まで見る。1の個数が0のとき、無限ループを抜け、make\_work\_sheetを終了する。

 そうでないとき、1つの列に記入されている1の個数がnのとき、list(1), list(2), ...., list(n) に行番号が入る。

2. 以下の2重ループにより、全ての組み合わせが得られるので、シートworkの2, 3列目に逐次記入する。

 i = 1 To n-1

 j = i + 1 To n

 （省略）

 Next j

 Next i

　離すべき生徒の表をシート「work」に書き込んだ後、以下の処理を行います。

1. シートworkに記載された生徒間の距離を計算する。プロシージャcalc\_distanceがその処理を行う。

2. 全ての条件が満たされていたなら終了

3. そうでない場合、条件を満たしていない2人について、どちらか一人を別の席に座っている人と交換する。

　calc\_distance() はシートworkのD列に各々の生徒のペアの席間距離を記入します。名簿シートのD列とE列にその生徒が座っている場所（行と列）が入っているので、席間距離の計算は簡単です。席間距離はAbs(i1 - i2) + Abs(j1 - j2) で定義します。絶対値をとる関数Abs() を使用しています。

　全ての条件を満足したか否かの判定をしているのが以下の部分です。

 satisfy = 1 ' sasitfy = 1 は全ての条件を満足したとき

 For i = 2 To njyoken + 1

 kyori = Sheets(swork).Cells(i, 4)

 If kyori < min\_dist Then

 satisfy = 0

 End If

 Next i

　satisfyはフラグ変数です。全ての条件を満足したなら1、そうでないなら0です。最初1を入れておいて、条件を満たさなかったら、0にします。min\_distは最小席間距離です。Cells(i, 4) は4列目に距離が入っています。

　1件でも条件を満たさない、あるいは10回再配置を試みても条件を満足できなかった場合、Exit Doして無限ループから抜けて、relocate() を終了します。

　それ以外の場合は、再配置が必要です。1行ずつシート「work」から席間距離を取りだし、満たしていないペアを再配置します。再配置はプロシージャmove\_manage() に任せます。1組再配置をすると、その時に動かした人の影響により、既に満たしていた条件が満たされなくなる場合があります。1組を再配置するたびに、calc\_distance() を実行します。

## 席の交換

　プロシージャmove\_manage() が入れ替えを行います。条件を満たさない2人のうち、片方の生徒を別の場所の生徒と交換します。以下のようなアルゴリズムを使います。

1. 条件を満たしていない2人それぞれについて、「座っている場所（行・列）」「特別席対象者か否か」「性別」を取得する。

2. 片方が特別席のときは普通席の人を移動させる。両方とも特別席あるいは両方とも普通席の場合は乱数で決める。

3. 動かす人の交換先となる席をピックアップする。交換先となる席が満たす条件は以下の通り。

 動かす人が特別席対象者の場合は、以下の条件を全て満たすこと。(a) 動かさない生徒の席からmin\_dist以上離れている。(b) 同性用の特別席である。

 動かす人が普通席対象者の場合は、以下の条件を全て満たすこと。(a) 動かさない生徒の席からmin\_dist以上離れている。(c) 同性用の普通席である。

　移動先候補となる席の行と列をrow( ), column( ) に入れていきます。候補となる席の数はnkouhoです。

　乱数でそのうち1個を選び、選んだ場所に座っている生徒と席の交換を行います。get\_rowは場所 (i, j) に座っている生徒が名簿の何行目にいるかを返すプロシージャです。get\_rowで移動先に座っている生徒の行番号を取得し、プロシージャexchange()で2人の席を交換します。

## 手動での再配置

　乱数で自動的に席を決めた後、手作業で席を入れ替えたい場合もあります。そのような要求に応えるため、「手動交換」のボタンを用意しました。manual\_exchante() が手動交換を実現します。○.○で学習したオブジェクトのループがあります。複数のセルを選択した状態で、以下のプログラムを実行すると、セルの個数だけループを回ります。cはRange(" ") オブジェクトであり、選択しているセルから1個ずつ取りだしてcに入れます。

 i = 0

 For Each c In Selection ' c には 1 個のセルが入る

 i = i + 1

 Next c

　このプログラムではiにセルの個数が入ります。2個以外の場合は、エラーを出すよう処理します。2個の場合、選んだ2人の名簿シート上での行番号を取りだし、exchange() で交換を行います。cがRangeオブジェクトのとき、行、列、セルの内容は以下のように、取り出します。

 gyou = c.row

 retu = c.column

 namae = c.value

## その他の処理

　「席の周囲に罫線を描く」「罫線を除去する」の2つのプロシージャがあります。罫線を引く・消すにはどうすれば良いか？　は「マクロの記録」で調べました。

　上下逆の座席表を作成する機能があります。セルの値をコピーするだけでなく、周囲の罫線と色もコピーします。

　どのプログラムも行っていることは単純なので、プログラムを見れば理解できるでしょう。

おわりに

　長いExcel VBAの講座を最後まで読んでいただきありがとうございました。アイデア次第でExcelは色々な使い道がありそうです。新しい使い方がまだまだ沢山眠っていると思います。「ちょっと面倒だな」と思っても、「少し頑張ってみよう」と思ってトライすることで、Excel VBAの腕は徐々に上達することでしょう。私のWebサイト

http://denki.nara-edu.ac.jp/~yabu/soft/excel-vba.html

もご活用下さい。それでは、皆さんがExcel VBAを楽しく活用されることを祈って、本書を終わりたいと思います。