

## パズルを利用したVBAプログラミング入門

兵庫県立神戸甲北高等学校教諭 山上 通恵

### 1. はじめに

学習指導要領では、「情報の科学的な理解」について理論的な扱いが「数理的、技術的な内容に深入りしない」とされ、また「コンピュータによる情報の表し方」「コンピュータによる情報の処理」の項目でも、「基礎的な内容にとどめ、プログラミング言語によらない方法」が勧められている。ここでは、パズルを利用することで、若干複雑な「プログラミング」に立ち入りながら、生徒がどんどんはまり込んでいく題材があるので紹介する。

### 2. パズルのルール

利用するのは「ナンバープレース」「数独」などと呼ばれるパズルである。

1	6			2			5	3
4	7		1		9		2	6
		8				1		
	4		3	2			6	
3				6				2
	9		7	5			4	
		4				7		
9	3		5		1		8	4
7	5			8			3	9

9行9列のマス目に数値を埋め込んでいくが、ルールは、

- ・各列に1～9までの数字が重複なく入る
- ・各行に1～9までの数字が重複なく入る
- ・各ブロック（太線で囲まれた3×3の領域）に1～9までの数字が重複なく入る

というたってシンプルなものである。ここでは、Microsoft Excelと付属のプログラミング言語 Visual Basic for Applications (VBA) を利用する。

### 3. 前提となる知識

プログラミングの知識はまったくない状態の生徒を想定するが、以下のような内容を事前に押さえておく必要がある。

- ・VBAエディタの起動と実行
- ・For～Nextの構文
- ・If～Then～Else～End Ifの構文
- ・Cells (y, x)valueでセルの数値を扱える
- ・Cells (y, x)Interior.ColorIndexでセルの背景の色を扱える

セルの座標表現では、数学の座標表現 (x, y) と混乱しないように注意する必要がある。

### 4. 解法の探求

まず、プログラミングの前に、必ずパズルを紙と鉛筆で解く体験が必要である。この体験を通じて解法の流れを確立し、それを具体的にプログラミングするという過程が必要である。

実際にパズルを解く過程で、このパズルは、各セルに入る数字を書き入れていき、1つしか入らないセルをその数字に決定する  
ある数字が入らないセルを消していき、決定できるセルを見つける

という二つの考え方で解くことがわかる。いずれも根拠を持って決定できる数字を探す手続きであり、推論は入り込まない（難易度によっては推論が必要なパターンもある）。

主に の考えで解法を構築する。まず、すでに数字の入っているセルには当然ほかの数字は入らないので、次のようにセルの背景をグレーにする。

```

For X = 1 To 9
  For Y = 1 To 9
    Cells(Y, X).Interior.ColorIndex = 0
    If Cells(Y, X) <> "" Then
      Cells(Y, X).Interior.ColorIndex = 15
    End If
  Next Y
Next X

```

ColorIndexの値15はグレーを表す。

1	6			2			5	3
4	7		1		9		2	6
		8				1		
	4		3	2		6		
3				6				2
	9		7	5		4		
		4				7		
9	3		5	1		8	4	
7	5			8			3	9

次に、例えば「1」について、すでに「1」が入っている行を探し、その行をグレーにして「1は入れない」ことを表すと、次のようになる。

```

For X = 1 To 9
  For Y = 1 To 9
    If Cells(Y, X) = 1 Then
      For i = 1 To 9
        Cells(Y, i).Interior.ColorIndex = 15
      Next i
    End If
  Next Y
Next X

```

同様に、すでに「1」が入っている列を探し、その列をグレーにして「1は入れない」ことを表すと、次のようになる。

```

For X = 1 To 9
  For Y = 1 To 9
    If Cells(Y, X) = 1 Then
      For j = 2 To 10
        Cells(j, X).Interior.ColorIndex = 15
      Next j
    End If
  Next Y
Next X

```

最後に、すでに「1」が入っているブロックを探し、そのブロック全体をグレーにして「1は入れない」ことを表すと、次のようになる。

```

For X = 1 To 9
  For Y = 1 To 9
    If Cells(Y, X) = 1 Then
      For i = Int((X - 1)/3)*3 + 1 To Int((X - 1)/3)*3 + 3
        For j = Int((Y - 1)/3)*3 + 1 To Int((Y - 1)/3)*3 + 3
          Cells(j, i).Interior.ColorIndex = 15
        Next j
      Next i
    End If
  Next Y
Next X

```

行、列の処理に比べて若干複雑であるが、例えば「 $i = \text{Int}((X - 1)/3) * 3 + 1$ 」は「見つけたセルが属するブロックの左上隅のセルの列番号」を計算している。

以上の処理の結果、「1」が入らないセルは、次のようになる。

1	6			2			5	3
4	7		1	9		2	6	
		8				1		
	4		3	2		6		
3				6				2
	9		7	5		4		
		4				7		
9	3		5	1		8	4	
7	5			8			3	9

ここで、最下行に注目する。ルールに従えば、最下行のどこかに「1」が入らなければならないが入る可能性があるのは左から3列目しかない。したがって、このセルが「1」に決定できる。この過程をコーディングすると、次のようになる。

```

For Y = 1 To 9
  Counter = 0
  For X = 1 To 9
    If Cells(Y, X).Interior.ColorIndex = 15 Then
      Counter = Counter + 1
    End If
  Next X
  If Counter = 8 Then
    For X = 1 To 9
      If Cells(Y, X).Interior.ColorIndex < > 15 Then
        Cells(Y, X) = 1
        Cells(Y, X).Interior.ColorIndex = 6
      End If
    Next X
  End If
Next Y

```

ここでは「1」の決定と同時に、背景を黄色 (ColorIndex = 6) にして、どのセルが決定できたかわかるようにしている。

同様の処理を列及びブロックに対しても行う。

以上の処理全体を「2」～「9」に対しても行う。

## 5. 発展

実教出版のサイト (www.jikkyo.co.jp) に、問題の提示からさまざまに工夫していく過程のファイルを置くので参考にさせていただきたい。いずれも生徒の発案から具体化していったものである。

問題を提示する

「1」が入れない場所をグレーにする

「1」が決定できる場所に「1」を入れる

「1」を「N」に拡張【擬似変数】

チェックに引き続き決定を自動化

ボタンで数字を選べる

ボタンで数字を選べる【引数の利用】

ボタンで数字を選べる【再帰】

解決済みのボタンを使用不可にする

全自動

(いずれも、問題をA1～I9ではなくB2～J10に配置したので、変数に加工が必要である)

## 6. まとめ

途中からパズルを解くおもしろさは消滅してしまう懸念はあるが、プログラミングのおもしろさに転化すると考えられる。生徒は行きづまると気分転換に紙と鉛筆でパズルを解き、改めてプログラミングに取り組むなどの姿勢が見られた。また、生徒の力で実現可能かどうかは別にして、次々とアイデアが提案され、問題点を細分化することで、可能な範囲で解決していったが、結果として「引数」や「再帰処理」の概念まで引き出せた。「おもしろさ」は「むずかしさ」を克服できると実感した。

現在、コンピュータ部の一部の生徒が、発展学習として「推論」や「バックトラック」の概念を導入し、今回紹介した手続きだけでは解けないパターンに挑戦している。

この課題を通じて、問題解決における分割統治的な考え方や、プログラミングのコーディングの基礎的な内容が指導できた。今後、科学的な理解の比重が高まる中で、パズルの解法のコーディングは、垣根の低い課題として有効であることが確かめられた。