

# Excel のレポート課題における学業不正防止システムの提案

今福啓<sup>†1</sup>

**概要:** 本研究では、講義における Excel のレポート課題において、ファイルコピーのような学業不正による提出を防ぐ方法を提案する。提案するシステムは Excel のシートと VBA (Visual Basic for Applications) により構築され、他人のファイルをコピーしてそのまま提出するとただちに判明する。また、受講者ごとに異なる課題を自動生成するため、自らが必ず課題を作成する必要がある。提案システムを用いることで、レポート課題における学業不正の防止にたいして有効な手段となる。

**キーワード:** Excel、学業不正、VBA

## Proposition of A System to Prevent Academic Misconduct for Excel Report Task

KEI IMAFUKU<sup>†1</sup>

**Abstract:** In this paper, we propose a new system which prevents academic misconduct such as submission by a file copy for an Excel report task. The proposed system is created by Excel sheet and VBA (Visual Basic for Applications). If we use this system, when the file is having copied and submitted, it will be revealed immediately. Furthermore, this system generates different tasks for every student, therefore everyone should have to solve by oneself. This system is effective to prevent the academic misconduct for the report task.

**Keywords:** Excel, Academic misconduct, VBA

### 1. はじめに

Microsoft 社の表計算ソフトの Excel は、データを計算式や関数で集計したグラフ作成、必要な情報だけを取り出すフィルタ、データの見出しから再集計したピボットテーブル、ソルバーを使った線形計画問題の解決、分析ツールを使った t 検定による母集団との平均値の比較といった幅広い用途に利用できる。

このように Excel では基礎から応用にいたるまで様々な処理が可能であり、企業においてよく使われることから、多くの大学では Excel を利用した問題解決方法についての講義が行われている。

通常、講義担当者は受講生が学んだ内容を正しく身につけているかを評価する。評価には試験やレポートが用いられるが、試験では時間制限のために講義で扱う内容の一部しか評価対象とすることができない。そのため、時間を限定せずにファイル作成ができる、レポート課題を評価に使用することがしばしば行われる。

レポート課題の有用性は、大学生に対する調査結果[1]から判断できる。この結果では「PC スキルは必要だと思うが、自信はある」と回答した割合は 23.6%であるのに対し、「PC スキルは必要だと思うが、自信はない」と回答した割合が 73.2%と非常に多い。これは、講義でさまざまな問題解決方

法を学んだとしても、多くの大学生がそれを使いこなすところまで習得できていないことを示している。

大学の講義で目指すのは、計算式やグラフ作成といった操作が容易な機能の習得ではない。目的は、さまざまな機能を組み合わせて問題を解決する能力を身につけることである。そのためには、時間をかけて不足する知識を補いつつ作成できる、レポート課題が望ましいと考えられる。

ソフトウェアを使って作成するレポート課題で大きな問題となるのは、課題を自分で作成せず、他人が作成したファイルをコピーしてそのまま提出するという、学業不正が容易となる点である。ドイツの 4 つの大学で調査した学業不正に関する研究結果では、調査時点から過去 6 ヶ月以内に何らかの学業不正を行った学生が 75%に達することが示されている[2]。そのため何も対策がなされないならば、監督者のいないところで作成されるレポート課題では、学業不正の生じる可能性が高まることは容易に想像できる。

学業不正が行われると、他の受講者との評価において公平性を欠くだけでなく、それを行った者にとっても知識の修得を妨げる、極めて望ましくない行為となる。そのため、他人が作成したファイルを自らが作成したものとしてコピーさせないことや、コピーを行ったとしてもすぐに判明するように対策を講じておくことは必須である。しかし、ソフトウェアを使ったレポート課題における学業不正をどう

<sup>†1</sup> 獨協大学  
Dokkyo University

防ぐのかという具体的な方法については、あまり積極的に議論されてこなかったといえる。

そこで本研究では、多くの講義で行われている Excel のレポート課題に着目し、「ファイルをコピーして提出すると容易に判明する」「自らが課題を作成しなければならない」という2点を実現するシステムを提案する。提案システムは、Excel のシートと VBA (Visual Basic for Applications) を組み合わせることで作成される。そしてファイルコピーによる提出という学業不正を防ぐと同時に、個人ごとに異なるレポート課題が自動生成されるため、他人が回答済みの内容を丸写しできないファイルを構築できる。

## 2. 提案するシステム構成

本研究で提案する Excel のシステム構成は、他人のファイルをコピーして提出するとただちに判明する「ファイルコピー対策」と、学生ごとに異なる「課題の自動生成」から構成される。システムは、学生の情報を記録しコピー対策に使用するためにあらかじめ作成した Excel のシートと、課題を自動生成する VBA(Visual Basic for Applications)のプログラムにより構成される。VBA のプログラムは付録 A. に掲載している。

### 2.1 システムの全体像

提案システムでは、初めてファイルを開いたときのみ VBA により「学籍番号」「氏名」を入力するウィンドウを表示し、入力された内容を作成済みのシート（以下「学生情報シート」とよぶ）に記録する。記録した後、ランダムに発生させた値をパスワードとしてシートを保護し、記録された内容を変更できないようにすることが重要な点である。この操作により、シートを再編集するためのパスワードは不明となり、シートからの読み出しはできるが、第三者による学籍番号、氏名の書き換えは困難となる。

学生情報シートに記録した後、ファイルごとに異なる課題を自動生成し、学生情報シートから学籍番号と氏名を読み出して課題シートの1行目にコピーする。この操作は必ずしも必要ではないが、シートを見るだけでどの学生のファイルであるのかが分かり、教員が採点する際の利便性が高まる。

ファイルを閉じる際には、学生情報シート以外をすべて非表示にし、パスワードによりブックを保護してシート構成を変更できないようにする。そのためファイルを開く際は、プログラム中に書き込まれているパスワードを使って保護を解除し、非表示のシートを再表示して編集できるようにする。マクロが無効としてファイルを開いても課題のシートが表示されず、課題作成に必要なシートを不用意に削除させないことが可能となる。

### 2.2 VBA のプログラム

本システムにおける VBA のプログラムは、3つの定数と5つのプロシージャから構成される。

定数は以下のとおりである（1～3行）[a]。

- a. Const 問題数 = 1
- b. Const パスワード = "123"
- c. Const 学生情報 = "学生情報"

a.の「問題数」は、自動生成する問題の数である。本研究では1つの課題のみを生成するが、作成したい課題の数に応じて変更する。

b.の「パスワード」は、ファイルを開く、あるいは閉じる際のブック保護に使用する。ここで示すパスワードは単純だが、実際には予測できない複雑なものを使わなければならない。

c.の「学生情報」は、受講者が入力した学籍番号および氏名を記録するために使用する「学生情報シート」の名称である。先に述べたとおり、このシートは事前に作成しておく必要がある。

プロシージャは次の5つである。

- a. Workbook\_Open (6～43行)
- b. Workbook\_BeforeClose (46～57行)
- c. 初期化 (60～83行)
- d. SUMIF の課題 (86～170行)
- e. 罫線 (173～180行)

Excel では、ブックを開く際に a.の「Workbook\_Open」が実行される。提案システムでは、初めてブックを開いた（11行の条件が成り立つ）ときのみ、学籍番号と氏名を Inputbox に入力して学生情報シートに記録する（13～19行）。学生情報シートは、ランダムに発生した値をパスワードとしてシートを保護する（22行）。これにより、記録された情報を読み出せるが、変更は困難となる。

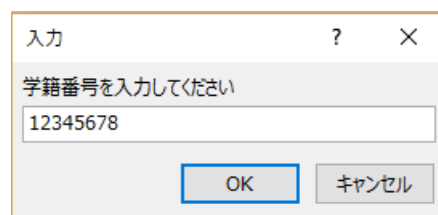


図 1 InputBox による学籍番号の入力例

Figure 1 Example of Entering Student ID with InputBox.

課題は、新たなシートを作成して自動生成する（25～30行）。本研究では課題数が1つであるが、さらに課題を作成する場合には、27～30行のうち28行のプロシージャを作成する課題に応じて変更し、31行の後に追加する。

本システムでは、マクロが無効の際にシートを編集でき

a) カッコ内は、付録 A の行番号をあらわす。

ないようにするため、ファイルを閉じる際に行われる b. の「Workbook\_BeforeClose」で課題シートを非表示にして (50~53 行)、定数「パスワード」を使ってブックを保護する (56 行)。そのため、ファイルを開く際には課題を作成できるよう、シートを再表示する必要がある (36~39 行)。あわせて定数「パスワード」を使ったブック保護により、学生情報シートや課題を作成したシートを不用意に削除できないようにする (42 行)。

c. の「初期化」では、課題を作成するシートの内容をすべて空にして列幅を初期状態に戻した上で (63, 64 行)、学生情報シートから読み込んだ学籍番号と氏名を 1~2 行目に記入する (67~82 行)。どの課題を自動生成する場合でも、事前にこのプロシージャを実行することで、課題を作成するシートのフォーマットを統一できる。

d. の「SUMIF の課題」では、Randomize 命令を使うことで、ファイルごとに異なる課題を自動生成する。ここでは、SUMIF 関数を使用して入力した条件を満たす内容だけを合計する課題を作成している。最初に c. の「初期化」を実行してシートのフォーマットを初期状態とし (87 行)、見出しとなるセルを入力する (90~120 行)。たとえば 19 行で使用している e. の「罫線」は、指定したセル範囲に罫線を引くプロシージャであり、後ほど説明する。ここで作成される課題は、具体的には以下の 3 つである。

- i. 指定された金額以上の売り上げを合計 (127, 128 行)
- ii. 指定された店舗の売り上げを合計 (130, 131 行)
- iii. 指定の日付まで、あるいは指定の日付以降の金額を合計 (133~138 行)

各課題で集計に使うデータは、141~155 行でランダムに作成される。

プロシージャの最後では、セル内の文字列が正しく表示されるように列幅を調整し (158~161 行)、課題を作成するセルのみを編集可能として (164~167 行)、定数「パスワード」を使ってシートを保護する (169 行)。

e. の「罫線」は、引数で指定したセルの周囲に罫線を引くためのプロシージャである。d. の「課題作成」のプロシージャで頻繁に罫線を引くことから、プログラム全体を簡略化するために使用している。

図 2 は、d. のプロシージャ「SUMIF の課題」を使用して課題を自動生成した例である。課題は「144 万円以上の売上を合計する」「獨協大学前の売上のみを合計する」「8 月 1 日からの売上を合計する」という 3 種類となっている。提案システムでは、このような課題がファイルごとに異なる形で自動生成される。

A	B	C	D	E	F	G	H
1	学籍番号		12345678				
2	氏名		獨協 一郎				
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							

図 2 プロシージャ「SUMIF の課題」による課題の自動生成例

Figure 2 Example of Automatic Generation by Procedure “Problem of SUMIF”

提案システムの作成手順をまとめると、次のようになる。

1. Excel を起動し、ブックを新規作成する。以下ではファイル名が「VBA テンプレート.xlsx」であるとして説明する。ファイルを保存する際、形式を「Excel マクロ有効ブック (.xlsm)」として、マクロの内容を保存できるようにする。
2. シートを 1 つ作成し、シート名を付録 A. の定数「学生情報」で設定した名称とする。
3. タブ「校閲」の「ブックの保護」で、定数「パスワード」と同じパスワードを設定する。
4. タブ「開発」の「Visual Basic」を左クリック後、開いているファイルの「VBAProject (VBA テンプレート.xlsx)」のタブ「Microsoft Excel Objects」にある「ThisWorkbook」をダブルクリックし、付録 A. の VBA のコードを入力する。
5. VBA のツール → VBA Project のプロパティ → 保護 → 「プロジェクトのロック」で「プロジェクトを表示用にロックする」にチェックを入れ、「プロジェクトのプロパティ表示のためのパスワード」に任意のパスワードを設定する。この設定により、ブック保護のパスワードが入力されている VBA のプログラムを他人に見られないようにする。

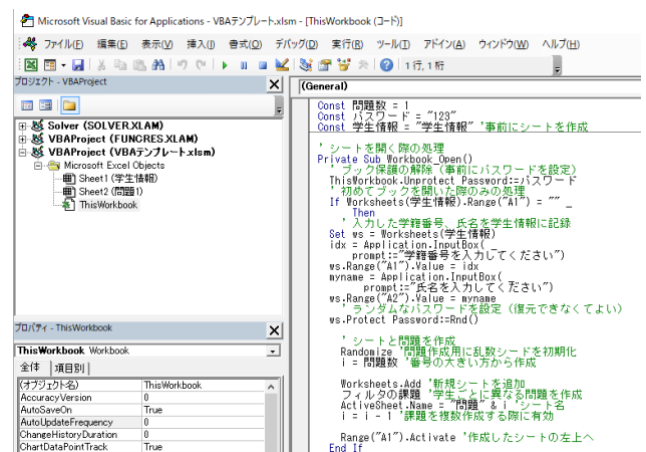


図 3 VBA の作成例

Figure 3 Example of creating VBA.

本研究で提案したシステムは、獨協大学の講義「コンピュータ入門 a, b」で実際に使用している。過去に1度、他人のファイルをコピーしたものが提出されたことがあるが、ただちにコピーであることが明白となった。そのため、提案システムは学業不正に対する高い有効性を持つといえる。

### 3. おわりに

本研究では、大学の講義で課される Excel のレポート課題において、他者の作成したファイルをそのままコピーして提出するといった学業不正を防止するためのシステムを提案した。提案システムは Excel のシートと VBA のプログラムにより作成され、他人のファイルをコピーして提出するとただちに判明する「ファイルコピー対策」と、学生ごとに異なる問題作成を自動的に実行する「課題の自動作成」から構成される。

提案システムを用いることで、他人の作成したファイルをコピーして提出することが不可能となる。そのため、講義の受講者は各自にあたえられたレポート課題を、自らが作成しなければならない。このようなシステムを構築することで、学業不正の防止に有効な手段となることが期待できる。

### 参考文献

- [1] “若者＝デジタルネイティブは本当？大学生の7割以上が、PCスキルに自信なし 採用担当者の約6割がPCスキル不足を感じていた！驚く新入社員のPC事情 PCを持ち歩く頻度が高い学生は、PCスキルが高い傾向に！” <http://nec-lavie.jp/common/release/ja/1702/0704.html> (参照 2018-02-20)
- [2] Justine, P., Sebastian, S., Floris, van V., Carola, G. and Stefan, F. Investigating the effect of academic procrastination on the frequency and variety of academic misconduct: a panel study. *Studies in Higher Education*. 2014, vol.40, pp.1014-1029.
- [3] 吉川順平. Excel VBA 最強リファレンス. SBクリエイティブ株式会社, 2014

### 付録

#### 付録 A. Excel VBA のプログラム

```

1. Const 問題数 = 1
2. Const パスワード = "123" 'ブック保護、解除に使う
3. Const 学生情報 = "学生情報" '事前にシートを作成
4.
5. ' シートを開く際の処理
6. Private Sub Workbook_Open()
7. ' ブック保護の解除
8. ThisWorkbook.Unprotect Password:=パスワード
9.
10. ' 初めてブックを開いた際のみの処理
11. If Worksheets(学生情報).Range("A1") = "" Then

```

```

12. ' 入力した学籍番号、氏名を学生情報シートに記録
13. Set ws = Worksheets(学生情報)
14. idx = Application.InputBox(_
15. prompt:="学籍番号を入力してください")
16. ws.Range("A1").Value = idx
17. myname = Application.InputBox(_
18. prompt:="氏名を入力してください")
19. ws.Range("A2").Value = myname
20. ' ランダムなパスワードでシートを保護
21. Randomize '乱数シードを初期化
22. ws.Protect Password:=Rnd()
23.
24. ' シートと問題を作成
25. i = 問題数 '番号の大きい方から作成
26.
27. Worksheets.Add '新規シートを追加
28. SUMIF の課題 '学生ごとに異なる問題を作成
29. ActiveSheet.Name = "問題" & i 'シート名
30. i = i - 1 '課題を複数作成する際に有効
31.
32. Range("A1").Activate '作成したシートの左上へ
33. End If
34.
35. ' 課題用シートを表示(閉じる際に非表示にするため)
36. For i = 1 To 問題数
37. Set ws = Worksheets("問題" & i)
38. ws.Visible = True
39. Next i
40.
41. ' パスワードでブック保護
42. ThisWorkbook.Protect Password:=パスワード
43. End Sub
44.
45. ' シートを閉じる際の処理
46. Private Sub Workbook_BeforeClose(Cancel As Boolean)
47. ThisWorkbook.Unprotect Password:=パスワード
48.
49. ' マクロ無効時にはシートを変更不可にする
50. For i = 1 To 問題数
51. Set ws = Worksheets("問題" & i)
52. ws.Visible = False 'シートを非表示にする
53. Next i
54.
55. ' パスワードでブックを保護
56. ThisWorkbook.Protect Password:=パスワード
57. End Sub
58.
59. ' シートを初期状態に戻し、学籍番号と氏名欄を作成

```

```

60. Private Sub 初期化()
61. Randomize
62. ActiveSheet.Unprotect Password:=パスワード
63. Cells.Clear 'シートを全て空にする
64. Columns("A:XFDD").ColumnWidth = 8.25 '幅を調整
65.
66. ' 1~2 行目に学籍番号、氏名欄を作成して記入
67. Range("B1:C1").Merge 'セルを結合
68. Range("D1:E1").Merge
69. Range("B2:C2").Merge
70. Range("D2:E2").Merge
71.
72. Range("B1:B2").Interior.ColorIndex = 4 '緑色
73. Range("B1").Value = "学籍番号" '学生情報を記入
74. Range("B2").Value = "氏名"
75.
76. ' 学生情報シートから、課題シートに情報を書き込む
77. Range("D1").Value = _
78.     Worksheets(学生情報).Range("A1").Value
79. Range("D2").Value = _
80.     Worksheets(学生情報).Range("A2").Value
81. Range("D1").HorizontalAlignment = xlLeft
82. Range("D2").HorizontalAlignment = xlLeft
83. End Sub
84.
85. ' SUMIF の課題を自動生成
86. Sub SUMIF の課題()
87. 初期化 'シートを初期化
88.
89. ' 見出しを作成する
90. Range("B4:D4").Merge
91. 罫線 ("B4:D4") '罫線を引く
92. 罫線 ("B5:D5")
93.
94. Range("B4").Value = "売上高"
95. Range("B5").Value = "契約日"
96. Range("C5").Value = "店舗"
97. Range("D5").Value = "売上高"
98. Range("B4:D5").HorizontalAlignment = xlCenter
99.
100. Range("G9:H9").Merge
101. Range("G10").Value = "検索条件"
102. Range("H10").Value = "結果"
103. 罫線 ("G9:H11")
104. Range("G9:H10").HorizontalAlignment = xlCenter
105.
106. Range("G14:H14").Merge
107. Range("G15").Value = "検索条件"
108. Range("H15").Value = "結果"
109. 罫線 ("G14:H16")
110. Range("G14:H15").HorizontalAlignment = xlCenter
111.
112. Range("G19:H19").Merge
113. Range("G20").Value = "検索条件"
114. Range("H20").Value = "結果"
115. 罫線 ("G19:H21")
116. Range("G19:H20").HorizontalAlignment = xlCenter
117.
118. ' 見出しのセルを黄色で塗りつぶす
119. Range("B4,B5:D5,G9:H10").Interior.ColorIndex = 6
120. Range("G14:H15,G19:H20").Interior.ColorIndex = 6
121.
122. ' 課題作成に使用する値
123. stations = Array("新越谷", "蒲生", "新田", _
124.     "獨協大学前", "草加", "谷塚", "竹ノ塚", "西新井")
125.
126. ' 課題を自動で作成する
127. Range("G9") = "売上高が" & _
128.     (120 + Int(Rnd() * 30)) & "万円以上"
129.
130. i = Int(Rnd() * (UBound(stations) + 1)) '駅名
131. Range("G14") = stations(i) & "のみ"
132.
133. i = 7 + Int(Rnd() * 3) '7~9 月をランダムに指定
134. If Rnd() < 0.5 Then '課題は「~月まで」の合計
135.     Range("G19") = i & "月末までの合計"
136. Else '課題は「~月から」の合計
137.     Range("G19") = i & "月 1 日からの合計"
138. End If
139.
140. ' 表データをランダムに作成
141. d = DateValue("2017/6/1") '日付
142. lastline = 53 + Worksheets(学生情報).Range("A1") _
143.     Mod 30
144. For i = 6 To lastline
145.     Cells(i, 2).Value = Date 'セルを日付形式に
146.     Cells(i, 2).Value = d 'B 列に日付を入力
147.     d = d + Int(1 + Rnd * 6) '日付を 1~7 日ずつ増加
148.
149. ' セルに店名を入力
150.     s = Int(Rnd() * (UBound(stations) + 1))
151.     Cells(i, 3).Value = stations(s)
152.
153. ' 売り上げを入力
154.     Cells(i, 4).Value = 10000 * (90 + Int(Rnd * 370))
155. Next i

```

```
156.
157. ' セル幅を調整
158. Columns("A").ColumnWidth = 5
159. Columns("B:D").ColumnWidth = 15
160. Columns("E:F").ColumnWidth = 5
161. Columns("G:H").ColumnWidth = 20
162.
163. ' 編集できる範囲を限定しパスワードでシートを保護
164. Range("H5:H6").Locked = False
165. Range("G11:H11").Locked = False
166. Range("G16:H16").Locked = False
167. Range("G21:H21").Locked = False
168.
169. ActiveSheet.Protect Password:=パスワード
170. End Sub
171.
172. ' 指定したセルの外枠を罫線で囲む
173. Private Sub 罫線(s As String)
174. With Range(s)
175.     .Borders(xlEdgeTop).LineStyle = xlContinuous
176.     .Borders(xlEdgeBottom).LineStyle = xlContinuous
177.     .Borders(xlEdgeRight).LineStyle = xlContinuous
178.     .Borders(xlEdgeLeft).LineStyle = xlContinuous
179. End With
180. End Sub
```