

VBAを用いたプログラミング指導実践

愛知県 金城学院高等学校教諭 加藤 和幸

1. はじめに

情報教育においてプログラミングの学習は、計算機の本質に触れさせたり、いくつかの事象をモデル化しシミュレーションをさせたりする場面で利用できる重要な内容である。また生徒にとっても、自分の計画通り動く作品が最後に残せる「ものづくり的」要素をもつ内容として、やりがいのある題材の1つであろう。しかし、指導の方法によっては抽象的な考えができずに、「数学嫌い」と同じような「情報嫌い」を作る可能性のある内容でもあり、指導する側も生徒の関心や興味をそらさないように心がけながらの指導が必要となるデリケートな内容でもある。

筆者は、高等学校の新カリキュラムで教科「情報」がスタートする以前から数学科の選択授業として、高校生2年生を対象にBASIC言語を用いて、数学A・B・C（旧カリキュラム）の内容を指導してきた。そこでの指導経験を生かし、新しい「情報」の授業内でも体系的なプログラミングの指導をどこかで取り入れられないかと、既存の必修教科「情報A」の中で試してみた。しかし、時間的に余裕がないことや、全員履修の科目ゆえ生徒の理解度に差があることなどから、なかなかカリキュラムとして定着させるには至らなかった。

そこで、2004年度より始めた高校2年・3年の選択教科の時間に「情報」という科目を設け、選択教科内での学習内容として計画を立てた。選択教科内で位置づけたことにより、時間的にも内容的にも余裕をもった取り組みができ、「情報ABC」の教科書の内容を少し発展させ、柔軟なカリキュラム設計が可能となった。

2. カリキュラムの実際

本校では、高校2年生より内部推薦で系列大学へ行くクラスと他大学進学クラスに分かれる。ここでの対象クラスは内部推薦のクラスで、数学が得意な理系進学の生徒は含まれないが、系列大学の情報文化学科や生活環境情報学科への内部推薦を希望している生徒は多く含まれる。

表1にそのカリキュラムの概略を示す。

表1 カリキュラムの図

	学習内容
1	プログラミングとは
2	順次的な処理
3	選択的な処理
4	確率的なモデル
5	アルゴリズム
6	繰り返し処理
7	自由作品の作成
8	評価と作品の修正

指導時間：16時間（連続2時間×8週）

プログラミング言語は「情報A」でなじみの深い表計算ソフトのマクロプログラムであるVBAを用い、操作段階で抵抗が少ない環境で学習させることとした。

3. プログラミングの内容

(1) 順次的な処理をさせる例題

最初の題材として「三角形の面積の計算」を扱った。まず、底辺と高さを入力させ、実行ボタンを押すことにより、セル上に面積の値を出力させるものを作成させた。

底辺や高さの値を入力させる場面では、Input-Boxによる方法を知らせ、値の入力の意味を実感

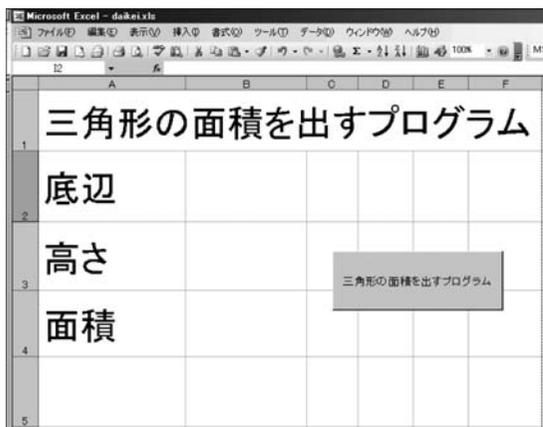


図1 「三角形の面積の計算」の画面

リスト1 「三角形の面積の計算」のプログラム

```

Sub sankaku()
    Dim a As Integer
    Dim h As Integer
    Dim s As Single
    a = Application.InputBox("底辺は")
    Cells(2,2).Value = a
    h = Application.InputBox("高さは")
    Cells(3,2).Value = h
    s = a * h/2
    Cells(4,2).Value = s
End sub

```

させる形にした。出力は表計算ソフト上での1つのセルにさせ、出力させる文字のデザインは表計算ソフト上で設定をさせた。また、実行の処理はプログラムとリンクさせた実行ボタンで行なわせ、ボタンのデザインも自由にさせてみた。面積を計算させるプログラムは、台形の面積や円の円周や面積計算にも発展させることができ、プログラミングの最初の例題としてその方法を定着させることができた。図1とリスト1がその画面とプログラムである。

プログラムを書かせるときに、いくつかの文法の指導は必要である。ただ、文法の指導が先になると、どうしてもそのことを必要以上に気にする生徒がみられ、プログラム作成時に意欲的でなくなる場面も見られる。体系的な文法の指導やいく



図2 「2次方程式の解の計算」の画面

つかの関数の指導は、プログラミングに少し習熟した後に行なうものとし、ここでは必要最小限なものだけを後追いさせる形で指導した。

また、多くの表計算ソフトには自動でマクロを記述する機能もあるが、その利用は最後に補足で行なうこととし、プログラムのコードは自分で記述することを原則とさせた。

(2) 選択的な処理をさせる例題

題材として「2次方程式の解の公式を利用した解法」を取り上げた。また、判別式の正負により解の判別もさせた。選択処理の必要性を感じさせる方法として、最初は普通の実数解が見つかる数値を利用するだけでの実行をさせ、途中で「解なし」の時にどうするのかという疑問を持たせた。そして、新しい処理手順として、if文による選択処理の導入をした。問題文は「 $ax^2+bx+c=0$ の形の2次方程式の解をa, b, cの値を入力させ、解の公式で計算させる」。図2にその画面、リスト2でそのプログラムを示す。

(3) 確率的なモデルの例題

次に「乱数により今日の運勢を5段階の数で占うゲームの作成」という例題を与えた。確率的なモデルの方法として指導が必要となる内容が乱数を用いる題材である。数学の題材を扱うより、コンピュータにしかできない手法を知らせることにより、プログラミングの楽しさに触れることができる題材で、最後の作品作りへの動機づけにもなる例題である。図3に画面、リスト3でプログラムを示す。

リスト2 「2次方程式の解の計算」のプログラム

```

Sub nijihou()
    Dim a As Integer
    Dim b As Integer
    Dim c As Integer
    Dim D As Single
    Dim x1 As Single
    Dim x2 As Single
    a = Application.InputBox("a=")
    Cells(2,2).Value = a
    b = Application.InputBox("b=")
    Cells(2,5).Value = b
    c = Application.InputBox("c=")
    Cells(2,8).Value = c
    D = b^2-4*a*c
    If D<0 Then
        MsgBox("解はなし")
    Else
        x1=(-1)*b + Sqr(D)/(2*a)
        Cells(5,4).Value = x1
        x2=(-1)*b - Sqr(D)/(2*a)
        Cells(6,4).Value = x2
    End If
End Sub

```

(4) 繰り返し処理をさせる例題

コンピュータの処理手順の指導の中で、最も時間をかけて指導すべき内容は、「繰り返し」の処理である。生徒にとっては前述の「順次的な処理」や「選択的な処理」は数学の思考とさほど変わらないため抵抗は少ないが、「繰り返し」の処理はプログラミングではじめて出会う考え方で、定着しにくい内容の1つである。ここでは、アルゴリズム指導の基本の例題である交換法のソートの例題を与え、じっくり取り組ませてみた。

図4にその画面、リスト4でそのプログラムを示す。指導をしてみると、配列の定義の段階で何をしているのかを見失う生徒が多く、本題のソートへなかなか理解が進まないことがわかった。

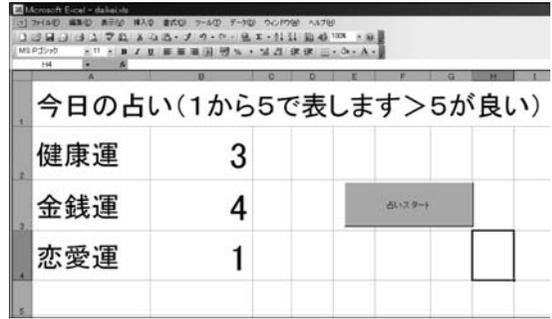


図3 「乱数によるゲーム」の画面

リスト3 「乱数によるゲーム」のプログラム

```

Sub uranai()
    Dim a As Integer
    Dim b As Integer
    Dim c As Integer
    Randomize
    a = Int(5*Rnd()+1)
    b = Int(5*Rnd()+1)
    c = Int(5*Rnd()+1)
    Cells(2,2).Value = a
    Cells(3,2).Value = b
    Cells(4,2).Value = c
End Sub

```

そこで、ソートさせる数を一般化せず最初から5つと決め、そこでの配列への初期値の代入方法も1つ1つ代入する形を取った。そのことにより、「繰り返し処理」が交換ソートの部分だけになり、アルゴリズムの定着もしやすいことがわかった。

また、生徒の中にはなぜ並べ替えを表計算ソフトの機能でやらせないかという単純な疑問を持つ生徒も多くいるため、表計算の機能が想起しにくい横並びの設定とした。

(5) 自由作品「占いゲーム」の作成

プログラムを作ることの醍醐味は、自分が計画をしたことが計画通りに、画面上で実現されることである。

いくつかの例題の学習の後、最終の課題としてオリジナルの「占いゲーム」を自由に作成させた。

	1	2	3	4	5	6	7	8	9	10	11
1	好きな整数を5つ下に入れなさい。										
2	3	5	2	7	1						
3	3	2	5	1	7						
4	2	3	1	5	7						
5	2	1	3	5	7						
6	1	2	3	5	7						
7											

図4 「ソートの例題」の画面

リスト4 「ソートの例題」のプログラム

```

Sub sort1()
    Dim D(5) As Integer
    Dim temp As Integer
    For i=2 To 5
        D(1)=Cells(i,1).Value
        D(2)=Cells(i,2).Value
        D(3)=Cells(i,3).Value
        D(4)=Cells(i,4).Value
        D(5)=Cells(i,5).Value
        For j=1 To 4
            If D(j)>D(j+1) Then
                temp = D(j)
                D(j)=D(j+1)
                D(j+1)=temp
            End If
        Next j
        Cells(i+1,1).Value = D(1)
        Cells(i+1,2).Value = D(2)
        Cells(i+1,3).Value = D(3)
        Cells(i+1,4).Value = D(4)
        Cells(i+1,5).Value = D(5)
    Next i
End Sub

```

ランダムに与えられる数字を用いて、それを画面上で表示させることやデザインの工夫などを通し、自分の作品として、プログラムを正確に作り上げることを目的とした。また、作成後は作品を共通フォルダに置かせ、それぞれ友達作品を試し合うことで簡単な相互評価もやらせてみた。その過程では友人が想定していない値を入力するな

どしてエラーが生じ、プログラムの修正を何度もしなければならぬ場面も多くみられた。

生徒の感想では、1つの完成作品としてプログラムを作ることは楽しかったというもので、作成するうちにあれもこれもと機能が広がっていき、時間切れになってしまったというものもいくつか出された。教師側の時間配分として、作品の作成時の最初の設計段階で時間を取らせることや、デバッグから運用テストの時間も十分に確保したカリキュラム作りが必要だと思われた。

4. おわりに

「手順的な自動処理」を体験させる手段として、プログラミングの指導は情報教育では不可欠なものである。ただ、その具体的な方法については千差万別で、どの方法が初等教育に適しているか、また、どれが生徒の関心をひき、将来の高等教育でのプログラミングにつながっていくかなどまだまだ手探りの状態であるといえる。

また、プログラミングの基本概念としていくつかの「アルゴリズム」の指導が必要だが、その指導をどこまで掘り下げているのかは大変難しい問題であることを実感した。より習熟度の高いプログラミング指導には「アルゴリズム」そのものを生徒自らが深め、構築させていけるような教材づくりが必要となる。

プログラミングを「モノ作り」の観点からとらえさせ、自分のオリジナルな作品作りをさせることは有用である。生徒には将来のプログラミング学習への強い動機づけにもなり、関心意欲を育てる題材としては適していることが実感できた。

今後の生徒らの関心は、表計算ソフト上でのプログラミングから、Web上でのプログラミングに移りつつある。そのためにはWebをプラットフォームにしたプログラミングにも取り組ませる必要がある、「JavaScript」などのHTML内で記述する言語なども学習する体験をさせ、それらも今後の実践課題の1つとしていきたい。