

# 女子短大英文学科におけるプログラミング教育

——論理的思考能力・問題解決能力などの向上を目指して——

青 山 広

## On Teaching Computer Programming in the English Department of a Two-year Women's College

——Strengthening the abilities of logical thinking and solving problems——

Hiroshi Aoyama

### 1. はじめに

ここ数年、コンピュータの普及には目覚ましいものがある。それに伴い4年制大学や短期大学における情報教育・コンピュータリテラシ教育もかなり普及し、文科系学部や学科においてもパソコンを使ったワープロ教育などは必修に近い形で行われつつあるように思われる。当短大英文学科でも平成3年度よりパソコンを使ったワープロ教育（日本語ワープロおよび英文ワープロ）を始め、平成7年度からは「情報基礎論」という通年科目を設け、コンピュータのハードウェアやソフトウェアについて教育するようになった。その中で、MS-DOSというOSのもとでBASICというプログラミング言語を使った簡単なコンピュータプログラムの作成を指導した。そして、平成8年度からはMicrosoft Windows 95というOSのもとで作動するMicrosoft Visual Basic Ver. 4.0というプログラミング言語を使ってプログラム作りを指導してきた。ただ、プログラムと言っても、英語学習ソフトウェアという、パソコン上で英語の学習ができるプログラム作りを専門に指導してきた。

本稿では、女子短期大学英文学科におけるプログラミング教育の実践とその問題点について考えるとともに、プログラミング教育そのものの可能性や教育効果などについて考察したい。具体的には、まず、筆者が初めてプログラミング教育を行った平成7年度の通年科目「情報基礎論」の内容に簡単にふれ、その後、平成8年度後期に筆者のゼミに入り、筆者担当の半期科目「基礎演習B」と平成9年度通年科目「英語演習」および「卒業課題研究」を受講した20名の学生に対する1年半の間に及ぶ指導をもとに考察する。

## 2. BASIC を使ったプログラミング教育

当英文学科では、「情報処理」という科目を開講しパソコンを使った日本語ワープロ・英文ワープロの教育を行っていたが、社会におけるコンピュータ利用の普及に伴い更なるコンピュータ教育の充実が求められ、平成7年度から「情報処理」を「情報基礎演習」に改めるとともに新たに「情報基礎論」という科目を設け、コンピュータのハードウェア・ソフトウェア両面について基礎的知識・技能を教育するということを始めた。筆者の研究分野は論理学と言語哲学であり、コンピュータ教育とは全く関わりがなかったが、この年から「情報基礎論」を担当することとなった。教育内容および方法とも何かと試行錯誤的色彩が強かったが、平成7年度におけるこの科目の授業内容を紹介するとともに、その中のプログラミング教育について述べる。プログラミング教育を実施した理由は、コンピュータで何らかの仕事させるプログラムを作成することにより、問題を分析したり総合したりする能力が養われると同時に論理的思考能力が培われ、また、利用者にとって少しでも使いやすいプログラムを考えることから、これからの社会で必要となるコンピュータを使った表現能力なども養われると考えたからである。

授業で使用したパソコンはスタンドアロンで設置された61台の「NEC 9801 DX」（HDDなし）であった。OSとしてはMS-DOS Ver. 3であった。プログラミング言語としては、BASIC/98という構造化プログラミングに対応したBASIC言語を使い、プログラミングの初歩を指導した。年間の授業スケジュールは以下のようなものであった：

4月：コンピュータの機器構成とそれらの基本的な働き。タイピング練習。

5月：MS-DOSの基礎

6月：BASIC言語によるプログラミングの実習。フローチャートの作成。

7月：同上

10月：同上

11月：アプリケーションソフト（ワープロソフトなど）の簡単な実習と

パーソナルコンピュータ利用技術認定試験（4級）の指導。

12月：BASIC言語によるプログラミングの実習。

1月：同上

6月から始めて1月までに約14回のプログラミング指導をした。テキストは市販のものを使い、次のような項目について指導した：

1. 四則計算とPRINT文を使った計算結果のディスプレイ表示。
2. REM文, END文, LET文, INPUT文の使い方。変数およびその型について。
3. フローチャート（プログラム・フローチャート）の必要性とその基本的作成法。
4. 条件分岐（IF-THEN文）、繰り返し処理（ループ）。

5. READ 文, DATA 文, GO~TO 文。
6. PRINT~USING 文, LPRINT~USING 文。
7. サブルーチン。

前期末までに上記3のフローチャートの作成までを指導し、レポート課題として「身長と体重をキーボードから入力し、標準体重と肥満度を計算し、その結果をディスプレイ上に表示するプログラムをフローチャートとともに作成しなさい。」という問題を出した。また、後期に入ってから、穴埋め問題や選択問題から構成される英語の学習プログラムを作成させた。ここで英語学習プログラム作成を教材として選んだ理由は、当時、英語学習ソフトがいくつか市販されており、筆者にとり非常に興味深かったことと、英単語の和訳を答えさせたり選択問題のようなプログラムであれば初心者でも簡単に作成できること、また、英語の問題を英文学科の学生自身に作らせることにより従来と異なった視点から英語の学習ができるのではないかと考えたこと、などを挙げるができる。もちろん、初心者にも簡単にプログラムを作ることができると言っても、英文学科の女子学生であり、コンピュータを専門的に学ぶために入学したわけでもなく、また高校時代にコンピュータを扱ったことのある学生は、受講生27名の内3名ほどであった。ただ、中には理解力のすぐれた学生も2割程度おり、課題を難なくこなしていた。これは後にも議論するが、コード（ソースコード）を文法規則に従って正しく書くということがやはり最も難しいようで、多くの学生がコードを書きながら、「今、自分が何を書いているのかわからない。」という感想をもらしていた。この点は筆者の力量不足でもあるのであるが、数学における計算式が続いているようなコードを考えることは文科系の学生にはなかなか困難なことであると思われる。

平成8年度に当短大ではLANが導入され、それに伴い61台のパソコンも更新され、さらに新たに2つの情報教室が設けられ、各情報教室のパソコンからインターネットにアクセスできるようになった。各教室のパソコンの構成は以下のとおりである：

情報教室1：Hitachi Flora DM1 61台（内教師用1台）

CPU : Pentium 75MHz  
メモリ：24MB  
HDD : 1 GB  
OS : Microsoft Windows 95

情報教室2：Hitachi Flora DM1 31台（内教師用1台）

ハードウェア構成・OSは上記教室1と同じ

情報教室3：Power Macintosh 7200 31台（内教師用1台）

CPU : PowerPC 601 90MHz  
メモリ：16MB  
HDD : 500MB

OS : 漢字 Talk7.5

### 3. Visual Basic を使ったプログラミング教育

平成8年度後期に筆者の「基礎演習B」を受講した20人の学生に対して、情報教室2において Visual Basic Ver. 4.0 を使ったプログラミング教育を始めた。同じ学生達が、平成9年度の筆者担当の「英語演習」と「卒業課題研究」を受講し、Visual Basic による学習を継続した。この節を進めるに当たって、これら3つの科目での指導内容をもとにして、平成9年度末（平成10年1月20日）に実施した下記のような無記名のアンケート調査の結果を盛り込みながら議論をする。

青山ゼミ アンケート調査 98/1/20

1 英語学習ソフトの作成について次の質問に答えて下さい。

ア Visual Basic の使い方はむずかしいですか？：

むずかしい 少しむずかしい どちらとも言えない 少しやさしい やさしい

イ プログラム作りで何がむずかしかったですか？ むずかしい順に 1, 2, 3と番号をつけてください：

英語の問題を考える

オブジェクト（ラベルやイメージなど）のプロパティを設定する

コードを考える

ウ プログラムを作ることが好きになりましたか？：

好きになった きらいになった どちらでもない

エ 上の質問ウに対する回答の理由を書いて下さい：

オ 卒業後もプログラム作りをしたいですか？：

続けたい たまにはしたい したくない

カ 英語学習ソフトを作って、英語の学習がますます好きになりましたか？

はい いいえ 以前と変わらない

キ 英文学科の学生が英語学習ソフトをつくることはどう思いますか？

したほうがよい (理由: )

しないほうがよい (理由: )

どちらでもよい (理由: )

ク 自分の作った学習ソフトは、対象としている学習者にとって役に立つと思いますか？

役立つ 少しは役に立つ 何とも言えない あまり役に立たない 役に立たない

ケ 英文学科の学生がコンピュータを習う場合、どんなことを習うとよいと思いますか？

つぎの中から習う（使う）とよいものに○をつけて下さい（一つ以上に○をつけて下さい）。

ワープロ 表計算（エクセル） データベース インターネット（ホームページから情報を得る） メールのやり取り プログラミング ホームページ作り 翻訳ソフトの使い方 英語学習ソフトを使う 他の学校（国内・国外）の学生との交流（テレビ会議やメールのやり取り） メールを使った英作文 インターネットを使ったリスニング訓練 写真や絵の加工や編集

- 2 この1年間を振り返って、授業内容について、全体的な印象と評価を書いてください。
- 3 講義内容・授業方法について良くなかった点や改善すべき点があったら書いてください。
- 4 授業で（もっと）教えてほしかったことがあったら書いてください。
- 5 短大に入学する前にコンピュータを使ったことがありましたか？： はい いいえ  
「はい」と答えた場合、何に使いましたか？：

ワープロ プログラミング インターネット その他

- 6 自宅でコンピュータを使っていますか？ 自宅でコンピュータを使っている人は次の質問に答えてください。

パソコンのタイプ：                    デスクトップ                    ノート

パソコンを購入した年月：

パソコンのメーカー名：

インターネットを使っていますか？：

はい いいえ インターネットに接続する方法がわからない

使っているワープロソフト： Word 一太郎 その他 使っていない

パソコンに満足している： はい いいえ（不満の理由：                    )

アンケート項目5の回答を見ると、

短大入学前にコンピュータを使ったことがある：13名

（ワープロ：11名 プログラミング：1名 インターネット：0名 その他：2名）

短大入学前にコンピュータを使ったことがない：7名

という結果であった。コンピュータを使ったことがあると言っても、ワープロを使うといった程度がほとんどで、あまり高度な利用はしていなかったことがわかる。

また、質問項目6の結果は、パソコンを所有している学生が20名中16名（80%）おり、その内デスクトップタイプのパソコンを所有している者が13名、ノートタイプのパソコンを所有している者が3名であった。この数字は英文学科の他のゼミと比べて高いものと思われる。また、購入時期は16名中9名（約60%）が平成8年の前半に購入しており、4名が平成7年に購入している。また、パソコン所有者の内インターネットを使っている者は16名中わずか1人であっ

た。所有しているパソコンに満足している学生は16名中12名（75%）で、そうでない者は4名であった。この4名中2名は使い方がよくわからないと答えている。

さて、「基礎演習B」で使用したテキストは次のものである：

『Windows 95 版 Visual Basic 4 オフィシャルコースウェア』

Michael Halvorson 著（豊田孝訳） アスキー出版局 1996

半年間の授業でカバーした項目は次のとおりである：

- 1) Visual Basic におけるプログラミングの方法
- 2) 各種コントロールの使い方
- 3) メニューの設定とダイアログボックスの利用
- 4) 変数と演算子
- 5) 条件分岐とループ

ただし、条件分岐とループについてはごく簡単にふれただけであった。下に Visual Basic Ver. 4.0 のスタート画面を示すが、従来のテキストベースのプログラミング環境とは全く異なり、グラフィカルなインターフェイスとなっている。

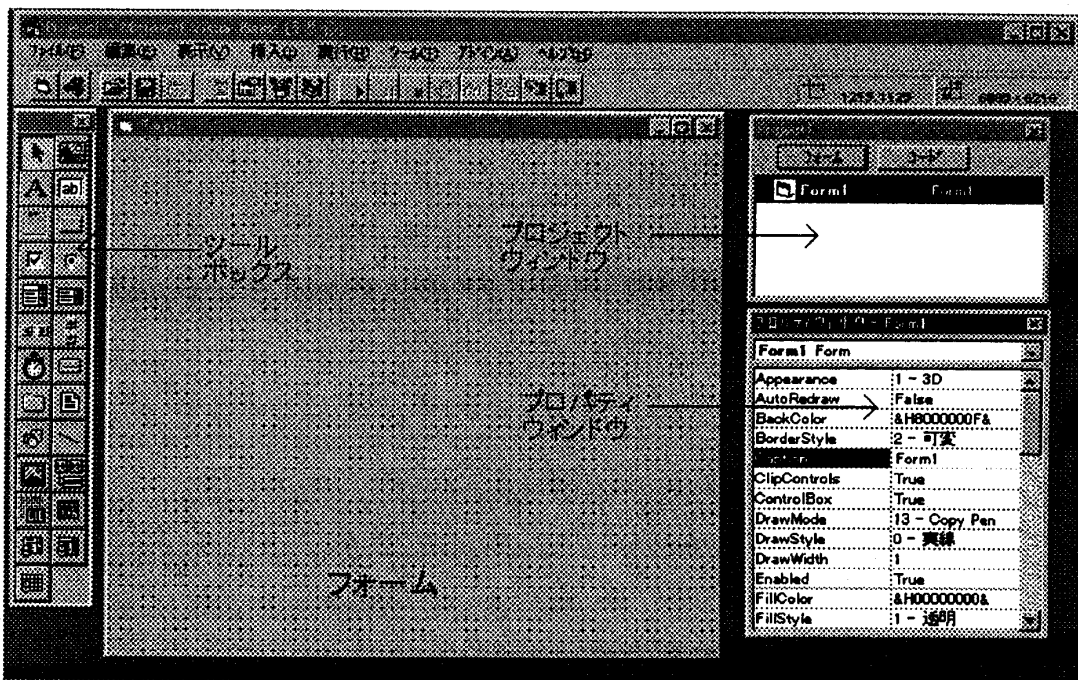


図 1 : Visual Basic 4.0 の起動画面

Visual Basic Ver. 4.0 でのプログラム作りは、大まかに次のような流れで進む：

1. ツールボックスの中にある各種コントロールを選択し、それを中央のフォーム上に配置する。

2. フォーム上のコントロールはオブジェクトと呼ばれ、それらのオブジェクトに適当なプロパティ（色、形、キャプション、位置など）をプロパティ・ウィンドウの中で設定する。
3. オブジェクトに対して何らかの働きをさせたい場合（例えば、マウスでクリックされた時、フォーム上に何かのメッセージを表示する）、そのオブジェクトをマウスでダブルクリックするとコードウィンドウが開くので、そこに必要なコードをキーボード入力する。これにより1つのサブルーチン（“プロシージャ”と呼ばれる）ができる。
4. 上記の1から3を繰り返す。

つまり、オブジェクト（フォームも1つのオブジェクトとみなされる）の設定とそれに関するプロシージャの記述によりプログラム作りが進むことになる。ここで、先のアンケート調査項目1-Aの結果をあげると

Visual Basic の使い方はむずかしいですか？：

むずかしい＝2名、少しむずかしい＝13名、どちらとも言えない＝2名、

少しやさしい＝3名、やさしい＝0名

となっている。この結果は、1年半 Visual Basic を使ってきて卒業間際の時点でのアンケート結果ということで、「基礎演習B」終了時、即ち、半年経過後では、もっと異なっていたと考えられる。その時点では、アンケート調査は実施していないが、ほとんどの学生がむずかしいと答えたであろう。実際、「最初はとてもむずかしかったが今は簡単」とコメントをつけた学生もいた。

また、先のアンケート調査項目1-Iの結果をあげると

「英語の問題を考える」を最もむずかしいと答えた学生＝1名

「オブジェクトのプロパティを設定する」を最もむずかしいと答えた学生＝0名

「コードを考える」を最もむずかしいと答えた学生＝19名

また、「英語の問題を考える」を3項目中最も簡単と答えた学生が12名いた。この12名は「オブジェクトのプロパティを設定する」を2番目に簡単（2番目にむずかしい）とも答えている。やはり学生にとってコードを考えることが最も難しいことが理解できる。プログラミング作成においてコードを考えることが最も難しいということは、よく理解できるのであるが、従来の Basic 言語と比較して Visual Basic の良い点は、コードで記述していた部分の多くを GUI (Graphical User Interface) を使ってある程度視覚的に設定できることである。フォーム上にマウスを使ってコントロールを配置し、プロパティウィンドウを使ってオブジェクトのプロパティを設定するということは、どことなくおもちゃのプラモデルを部品から組み立てていく過程と似通っており、手先が器用といわれる日本人にはとても使いやすいプログラミング言語であると言える。実際、Visual Basic の利用者は増えており、また、ソフトウェア開発業界においても Visual Basic を使える人の需要が高まっているようである。

しかしながら、プログラミングにおいてコード記述は必要であり、とくにプログラム中の重要な部分においてコードを書くということは避けて通れないし、これから先、どんなに GUI 化が進んでも何らかの形でコードを書くということは避けられない。また、コード記述の準備段階であるアルゴリズムを考えるという点についてもなかなか困難な点である。これはある程度、数学的な論理的思考能力が必要であり、一般的に文科系の学生には苦手な部分かもしれない。アルゴリズムを考える場合、ある目標を達成するにはどのような手段・手続きを行えば良いのかをまず論理的に筋道立てて考えないといけない。問題を分析的にもまた総合的にも考える必要がある。例えば、商店で何かしら商品を購入する場合を考えてみる。購入したい物を決定し、必要なお金を用意し、その品物を販売している商店へ行き、求める商品を購入する。品物がなければ注文をし、必要なら申し込み金を支払って帰宅するというアルゴリズムが一例として考えられよう。これをフローチャートにしてみると下図のようになるだろう：

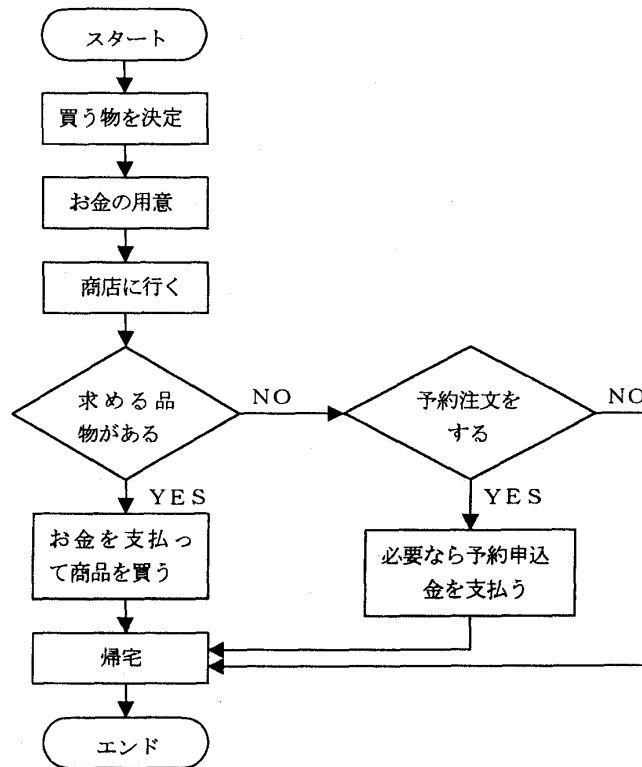


図 2：買い物をする時のフローチャートの例

ここで、長円形はフローチャートの始まりと終わりを示し、長方形は何らかの処理・作業を示し、ひし形は判断処理を示す。この例は非常に簡単なケースであるが、このような問題解決のアルゴリズムを考え、そしてそれをフローチャート化するという作業はおおよそありとあらゆる問題解決に対して応用できる。問題を考える時にこのようにフローチャートを書いて問題解決を視覚化しながら行うことにより、問題を分析する能力や論理的な思考力が養われる。



ここでプログラミング作業の流れを教科書的に書けば次のようになる：

1. 問題提起
2. 問題の分析
3. 問題解決の方法（アルゴリズム）の決定と流れ図（フローチャートやPAD）の作成
4. プログラムのコーディング
5. プログラムの試験運用とデバッグ
6. マニュアルの作成

もちろん、ほんの数十行のプログラムを書くのにフローチャートの作成は必要ないと思われるが、プログラムが大きなものになればなるほどその必要性は高まる。ただし、今回の指導の中では、フローチャート作成は特に指導しなかった。これは、学生に作るように指導したプログラムの形式をある程度統一したからである。それをフローチャートで表せば、ほぼ次のようになる：

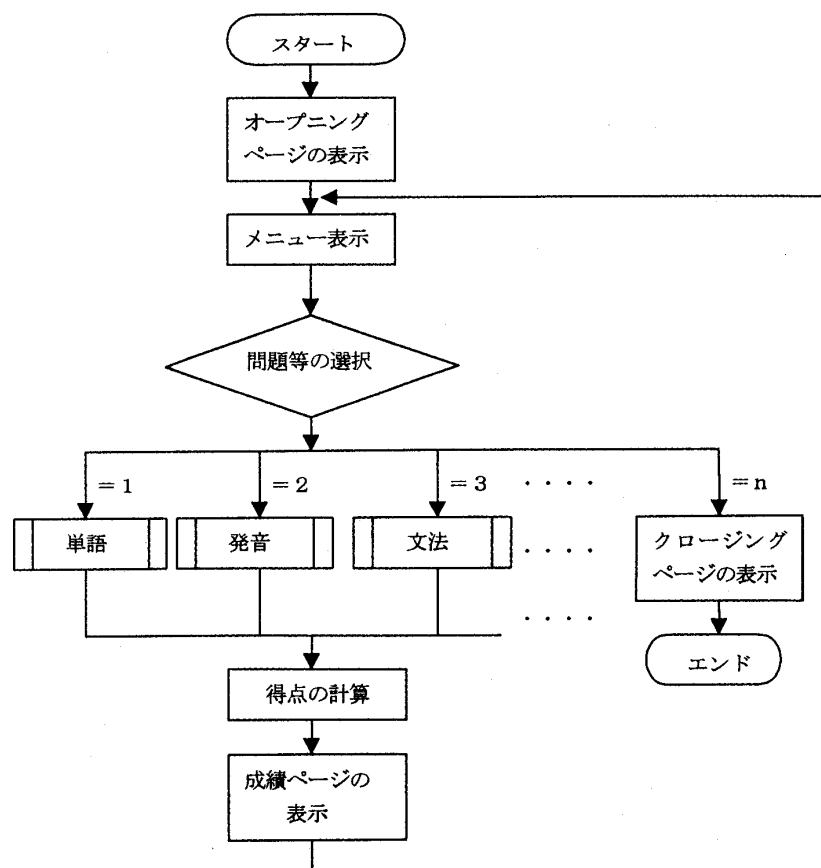


図3：英語学習ソフトのフローチャート

単語、発音、文法といった各サブルーチンの中身については学生にまかせた。また、結果的に、全ての学生がこのフローチャートのおりのプログラムを作成したというわけではなく、単語、発音、などの問題の切り分けやプログラムの終了方法などは学生各自の考えにまかせた。

ただ、必要最小限の要求として次の項目を提示した：

1. メニューで問題を選択する。
2. 成績を表示する。
3. フォームは最低10枚作る。
4. フォーム上のオブジェクトが自動的に移動したり、マウスで移動させたりできるようにする。
5. フォームやフォーム上のオブジェクトのデザインや色づかいに配慮する。
6. プログラムの中で音・音楽ができるようにする。

ただし、最後の音・音楽に関しては指導した時期がかなり遅くなってしまったため、ほんの少数の学生しか取り入れることができなかった。フローチャート作成の指導を特にしなかった理由のもう一つは、以前にも述べたように、Visual Basic によるプログラミングは GUI 化が進んでおり、今回の英語学習ソフトのようなプログラムであれば、フォーム上にコントロールを適当に配置し、プロパティウィンドウでそれらのプロパティを指定すれば基本的な部分は出来上がってしまうからである。ただ今から思うと、将来大きなプログラムを作成する可能性も考えてアルゴリズムとフローチャートについてしっかり指導すべきであった。

さて、次にコード生成（コーディング）について考える。1年半の期間の内、最初の1年間は Visual Basic の基礎的項目についてテキストを使いながら、また筆者の用意したサンプルプログラムおよびそのコードを学生に提示しながら授業を進めた。例えば、平成9年6月には動きのあるフォームを作るということで次のような「もぐらたたきゲーム」に類似したプログラムを作成させた：

□

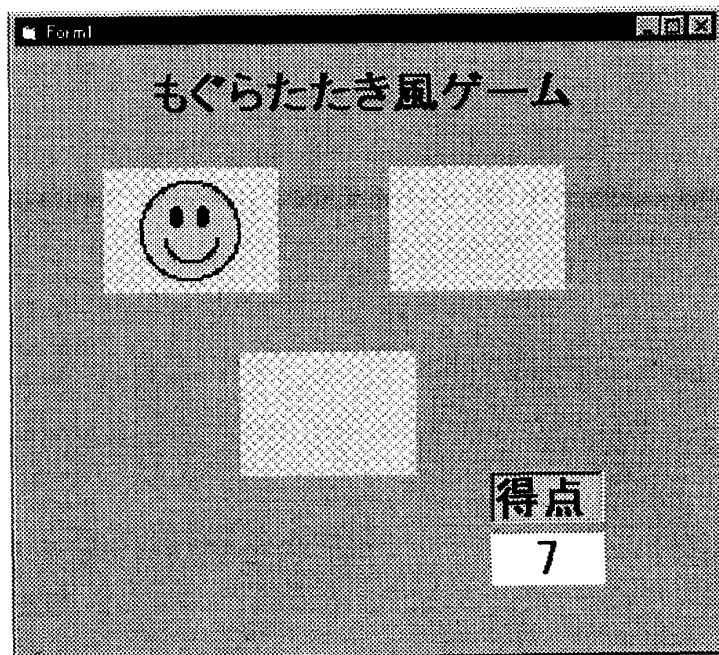


図4：もぐらたたき風ゲームの実行画面

このゲームはフォーム上に3個の長方形（3個のラベルを使用）があり、そのいずれかにランダムに人の笑った顔（通称、にこちゃんマーク。これは配列で生成された3個のイメージ）が現れる。現れたらすかさずマウスのポインターで顔をクリックする。うまくクリックできたら得点が1点ずつ加算される。クリックが外れた場合はビープ音が鳴る。このプログラムのソースコードは次のようなものである：

---

#### Option Explicit

```
Dim mogura, score As Integer      ' Option Explicit で変数の宣言をすると、そのフ
                                   ' ォームに関連する全てのプロシージャでその変数
                                   ' を同じ変数として使用できる。
```

#### Private Sub Form\_Load()

```
Timer1.Enabled = True           ' タイマーを起動する。インターバルは1秒弱。
Randomize                        ' 乱数生成の初期化。
```

End Sub

#### Private Sub Image1\_Click(Index As Integer)

```
If mogura = Index Then          ' マウスでクリックされたイメージの Index 番号
    Beep                          ' とにこちゃんマークの表示されているイメージの
    score = score + 1             ' Index 番号が一致している時だけビープ音を鳴ら
    Label16.Caption = score       ' し、得点を1加え、にこちゃんマークの絵を消す。
    Image1(Index).Visible = False ' 間違ってもクリックしても何も起きない。
```

End If

End Sub

#### Private Sub Timer1\_Timer()

```
Image1(0).Visible = False       ' タイマーが起動するたびに3個の
Image1(1).Visible = False       ' イメージ上のにこちゃんマークは
Image1(2).Visible = False       ' 見えなくする。
mogura = Int(Rnd * 3)           ' 変数 mogura に0,1,2のいずれかの数字が入る。
Image1(mogura).Visible = True   ' にこちゃんマークのでのイメージを表示する。
Image1(mogura).Picture = Image2.Picture ' にこちゃんマークの表示。
```

End Sub

---

プログラム1：もぐらたたき風ゲームのソースコード

これを見てわかる通り、記号'で始まるコメントをなるべく多く書き記して学生に各行の意味がよく理解できるようにした。Visual Basicの文法的項目、例えば、関数、メソッド、イベントプロシージャなどはできる限り、練習問題を実行させてその中でそれらの働きを理解させないといけない。

次に、デバッグについて簡単に述べる。Visual Basicでは各種のデバッグ機能が存在し、比較的容易にプログラム中のバグ（誤り）が見つけれられる。例えば、

1. キーボードからコードを打ち込んでいる時
2. プログラムの実行時
3. exeファイルの作成時

といった時にバグがあれば、メッセージなどで知らせてくれる。ただ、よく指摘されるように、単語の綴りの間違いといった簡単なバグであれば問題はないのであるが、論理的な間違いやVisual Basicの内部的な仕様にもとづく不具合などはなかなか面倒である。例えば、フォーム上に複数のオプションボタンを配列で作った時、Index番号0のオプションボタンがプログラム実行時に常に選択された状態になってしまうが、これなどは、仕様のなものでしかたないのかもしれないが、何とか改善してほしいものである。Visual Basic Ver. 5でもこれは改善されていない。

いずれにしても学生の作るプログラムには、問題処理のアルゴリズムの誤りやIf~then構文や繰り返し構文などの文法規則の誤解などに起因するバグがかなり見られた。そうしたものをなくすにはある程度の経験や慣れも必要であるが、基本的文法項目をしっかり理解させることが必要である。

ソフトウェア開発の最後の過程であるマニュアルの作成に関しては、今回は特に作成するように指導はしなかった。しかし、多くの学生がプログラムの最初のところでその学習ソフトの内容や使い方を説明するフォームを作成した。現在のソフトウェア開発の流れとして、マニュアルのオンライン化が推進されているが、学生の作成したものはそれほどの本格的なものではない。ただ、今後はそうしたオンラインマニュアルを作成させるべきであると考えている。

#### 4. 学生の作品例

ここで学生の作品を紹介したい。鈴木由香と長瀬美枝の作品である。この2人の作品は平成10年3月7日に東京都港区で開かれた第15回パソコン利用技術研究発表会（社団法人パーソナルコンピュータユーザ利用技術協会主催）で研究発表したものである。

鈴木作品は中学生・高校生向けの英語学習ソフトで3つのステップから構成されている。最初の2つのステップは単語、文法、作文、読解の4種類の問題群からなり、3番目のステップは総合問題となっている。各ステップ毎に得点の表示がされ、最後のステップの得点表示ベ

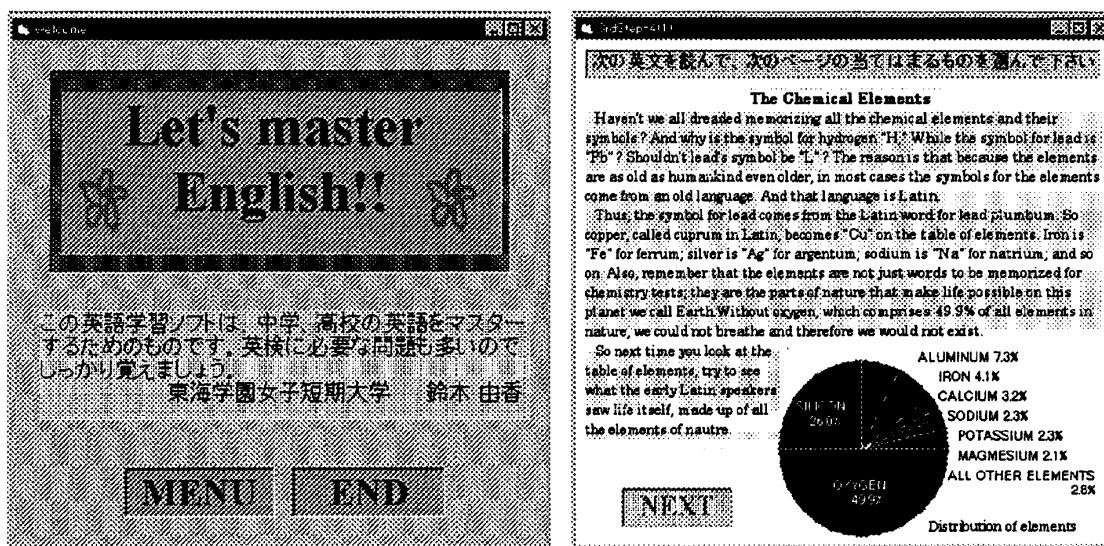


図 5：鈴木由香の作品から

ページには3つのステップの合計得点も表示されるようになっている。また、プログラム実行時にCD-ROMドライブに音楽用CDを入れておくとプログラムの実行中はCDの音楽が流れるようになっている。彼女は私が授業中に指導したことのほとんど全てを自分のプログラムの中に取り入れた唯一人の学生である。

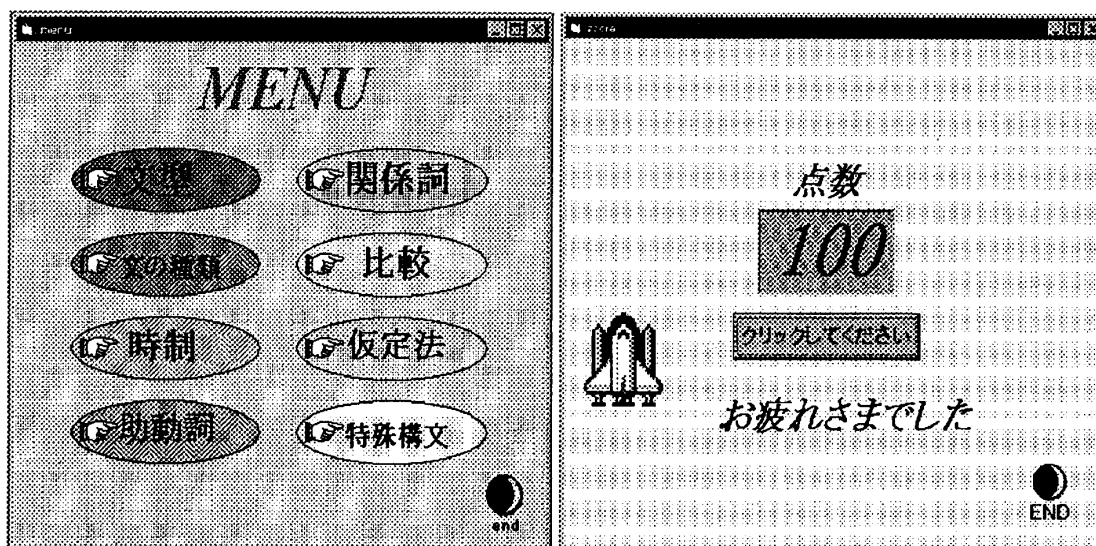


図 6：長瀬美枝の作品から

長瀬美枝の作品は高校生向けのプログラムであり、主に文法の学習に力点が置かれている。問題は8つの文法項目に分けられており、各項目は複数のページ（フォーム）にわたって問題が作られている。各項目毎に得点が表示されるが総合得点は表示されない。長瀬の作品の特徴

は各ページのデザインが優れていることである。各種のアイコンなどの小物を上手に使っており、また、各問題ページに正解チェック、ヒント、やり直し、次ページへ進む、メニュー、終了といったボタンが配置され、学習しやすさに配慮されている。更に動きのあるオブジェクトも特徴があり、各問題項目の最後の得点表示ページではロケットが飛んでいくのであるが、すこしずつロケットが小さくなっていくという細かい演出をしている。楽しく学習のできるソフトウェアとなっている。

さて、ここで再びアンケートの結果について見てみたい。項目1-ウの「プログラム作りが好きになりましたか」という質問については、10名の学生が「好きになった」、10名が「どちらでもない」と答えた。「好きになった」学生が50%おり、「きらいになった」学生が皆無であったことは筆者にとって非常に喜ばしい結果である。その理由を尋ねた項目1-エの回答では、「好きになった」と答えた学生は「プログラムを作るのが楽しい」、「自分の考えたプログラムが思うように動くのが楽しい」などと回答している。また、「どちらでもない」と答えた学生でも「コードを少し間違えただけで動かないのがきらい、でも、楽しい時もあった」とか「むずかしくて大変だったけど、楽しかった」というように「面倒・むずかしい、でも楽しい」といった意見が多かった。また、項目1-オの「卒業後もプログラム作りをしたいですか」という質問には、1名が「続けたい」、14名が「たまにはしたい」と答えている。項目1-ウで「どちらでもない」と答えた10名の学生の内5名が「たまにはしたい」と答えている。

次に、項目1-カから1-ケについて見る。1-カの質問は、「プログラム作成を通じて、英語の学習がますます好きになりましたか」というものであるが、「はい」と答えた学生が2名、「以前と変わらない」が18名、「いいえ」が0名となった。筆者は当初、それまで英語の問題を与えられて解くだけであった学生に英語学習ソフトを作らせることにより、異なった視点から英語をみつめてもらい、英語学習にさらに興味を抱いてほしいというほのかな期待を抱いていた。しかし、この質問項目の回答を見る限り、その期待はあまりかなえられていない。考えられる理由の1つに、授業での指導の中心が Visual Basic の使い方であり、英語の問題作りや英語そのものの指導が十分できなかったということと関係しているかもしれない。また、そもそも英語学習ソフト作りと英語学習が好きになると言うこととはあまり関係がないのかもしれない。この点については、もう少し指導方法などを変えたりして両者の関連を探ってみたい。

質問項目1-キは、「英文学科の学生が英語学習ソフトを作ることをどう思うか」という質問であるが、「したほうがよい」が15名、「どちらでもよい」が5名であった。「したほうがよい」の理由のなかで最も多かったのが「英語の勉強にもなる」あるいはそれと同趣旨の答えを書いた学生が6名いた。この点は、前項目の結果とのからみで言えば、英語学習ソフトを作ることは、英語の勉強にはなるがそれでもって英語学習がますます好きになるということはない、ということであろう。また、「したほうがよい」と答えた者のなかで、「英語を学習している人のほうが問題を作りやすい」と書いた学生がいた。これはある意味で当然のことで、プログラ

ミングを職業にしているいわゆるプログラマーであるからといって英語学習ソフトを作成できるということはない。市販の英語学習ソフトでは必ずと言ってよいほど英語の教師が制作に参加している。また参加しなければまともなソフトはできないであろう。

質問項目1-クは、「自分の作ったソフトは役に立つと思いますか」というものだが、「役立つ」1名、「少しは役に立つ」11名、「何とも言えない」6名、「あまり役に立たない」2名であった。この結果は多少控え目な自己採点のようにも思われるが、まずまず妥当なものであろう。

質問項目1-ケは、英文学科の学生がコンピュータを使って何を習ったら良いかという質問であるが、

ワープロ：14名                      インターネット：10名                      表計算：9名

翻訳ソフトの使い方：9名      プログラミング：8名                      他校の学生との交流：8名

メールを使った英作文：7名      メールのやり取り：6名      英語学習ソフトを使う：3名

などとなっている。ワープロ、インターネット、表計算、翻訳ソフトに対する関心が高いのは、個人的な興味（例えば、インターネット）ということもあるであろうが、大学での勉学上必要であったり、就職を意識した結果であろう。プログラミングを学習すると良いと答えた学生が8名いたことは、質問項目1-キの結果とも関連しているであろうが、興味深い。

質問項目2では、筆者の授業の評価をさせたものだが、結果としては

楽しかった（良かった）：5名

むずかしかったけれども楽しかった（良かった）：11名

むずかしかった：3名

無記名：1名

となった。16名（80%）の学生が楽しかった（良かった）と答えてくれたということは、幾分お世辞も入っているであろうが、筆者としては喜ばしい結果である。また、「ほかのゼミでは学ぶことが出来ないことを学べて良かった」、「最初はこのゼミを選んで失敗したと思ったけれど、終わってみると自分のためになったと思うようになった」という回答もあり、心強く感じた。

質問項目4は、授業でもっと取り上げてほしかったことをたずねたものだが、12名が無記名または「特になし」と答えた。また、3名が表計算ソフトの使い方を習いたかったと答え、2名がインターネットの使い方、2名が、Visual Basicのコードの作成についてもっと教えてほしいと答えた。

## 5. ま と め

本稿では、当英文学科の20名の女子学生に対する1年半の期間にわたるVisual Basicを使ったプログラミング教育について述べ、その可能性や効果などについて考察してきた。英文学科

におけるプログラミング教育と英語学習ソフトの作成という2点からまとめてみたい。

(1) 英文学科においてプログラミング教育をすることについて

- ア. 従来プログラミングは工学部や情報関連学部・学科の学生に教えるもので文系、特に英語関連学部・学科の学生とは関係のないものと考えられてきたし、今でも、ほとんどの人はそう考えているものと思われる。しかし、文系の学生にプログラミング能力がないとは言えないことが筆者の試みからも結論できる。また、プログラミングがどのようなものか知らなくても、学習することにより、プログラミングが好きになる学生が多くいるということは、4年制大学や短大の文系学部・学科でもプログラミング教育を導入することに何ら問題はないし、高度に発達したコンピュータ社会で活躍することになるこれからの学生にとっても有意義な学習経験となるであろう。
- イ. プログラミングを習うことにより、普段使うワープロソフトなどのアプリケーションソフトの仕組みなどが多少なりとも理解でき、コンピュータリテラシ教育にもなる。コンピュータを単なるブラックボックスとさせないためにも有効な教育であると考えられる。
- ウ. プログラミングでは論理的思考能力が必要不可欠である。問題解決のアルゴリズムを考えたり、フローチャートを書いたりすることにより問題に対する分析能力や論理的思考能力が養われる。また、自分で様々な解決方法を考え、それを実現していくということで問題解決能力や創造性が培われる。
- エ. 文系の学生にもプログラミング能力は十分あるが、苦手な学生がいるのも事実である。そこで、数学的な記号の羅列をなるべく避けるような工夫が必要かもしれない。Visual Basicはその点、従来のBasicやCといったプログラミング言語と異なり、プログラミングにおけるGUI化が進んでおり初心者の学生にとって使いやすい言語となっている。
- オ. プログラムが出来上がって思い通りに動いた時の満足感や達成感が大きく、学生にやる気を起こさせる。

(2) 英文学科の学生が英語学習ソフトを作成することについて

- ア. ソフトを作りながら英語の勉強ができる。
- イ. 英語学習ソフトを作るのは英語を日々学習している人のほうが良い。また、そうでなければよい英語学習ソフトはできない。
- ウ. CAI(Computer Assisted Instruction) ソフト全般について言えるが、学習者が楽しく学習できるようにプログラムの流れやインターフェイス部分のデザインに工夫を凝らすなどの努力をするため、表現力が磨かれる。
- エ. 他人が自分の作ったソフトで英語の勉強をしてくれるということで、人のために多少なりとも役立つことをしたという喜びや満足感が得られる。



以上のようにまとめることができる。今回は、英文学科の学生が英語学習ソフトを作るという試みを考えたが、これは文科系の他の学部・学科でもその分野のCAIソフトを学生に作らせることにより同様の効果や成果が得られると思われる。これからの社会は言うまでもなく情報化社会である。そうした社会で必要不可欠なコンピュータリテラシ教育をするのにプログラミング教育は非常に有効かつ有意義な手段であろう。

#### 参考文献

- [1] 浅岡奈美・内山理恵・青山広「Visual Basic Ver. 4.0を使った英語学習ソフト」『第14回パソコン利用技術研究発表会講演論文集』Pp. 124-126, 1997
- [2] 浅岡奈美・内山理恵・青山広「短大英文学科生による英語学習ソフトウェアの作成」パソコンリテラシ Vol. 23 No. 7 Pp. 27-30, 1998
- [3] 生田茂『情報科学の基礎』講談社 1993
- [4] 小川真一『JIS フローチャートと構造化チャート』工学社 1992
- [5] 河西朝雄『Visual Basic Ver. 4.0 初級プログラミング入門 (上)』技術評論社 1997
- [6] 鈴木由香・長瀬美枝・青山広「女子短大生による英語学習ソフトウェアの作成」『第15回パソコン利用技術研究発表会講演論文集』Pp. 79-81, 1998