

# Visual Basic によるプログラミング言語教育

## Education of Programming with Visual Basic

野澤 智\*  
Satoshi Nozawa

概要：大学・短大における情報教育について述べる。はじめに、情報リテラシー教育の現状を述べる。さらに、インターネットの急速な普及に伴い、情報リテラシー教育で今後何が重要視されるべきかについて議論する。次に、情報教育の一つであるプログラミング言語教育について解説する。特に、プログラミング言語教育の大学教育における存在意義について述べる。最後に、プログラミング言語教育の1事例として、短大のプログラミング演習を紹介する予定である。

### I. はじめに

近年、大学・短大における情報教育の重要性が益々高まってきている。これは、現代社会の情報通信化が急速に進み、理科系の卒業生のみならず、ほとんどすべての社会人にとって、情報に関する知識・技能が不可欠となっているからである。また、これを受けて小・中・高等学校でも情報教育が必修となり、2003年度入学生からは、全員が高等学校で情報に関する科目を履修してくる。したがって、当然のことながら、近い将来、大学・短大における情報教育のあり方もこれに対応することが不可欠となってくる。これからの情報教育に関する将来計画については、今後十分な検討を重ねた上で決定することが必要である<sup>1)</sup>。

さて、現在の大学・短大における情報教育には、次の3つの柱というべきものがあると考えられる。それらは、

- (1) 情報リテラシー教育
- (2) プログラミング言語教育
- (3) 情報処理に関する専門教育

である。本論文では、情報リテラシー教育とプログラミング言語教育についての発表を行う。情報処理に関する専門教育については、ここでは取り上げない。本論文の構成は次のとおりである。II節では、大学における情報リテラシー教育を取り扱う。また、III節では、プログラミング言語教育の存在意義を議論し、IV節では、Visual Basic によるプログラミング言語教育の事例を紹介する。最後に、V節で Visual Basic による作品例を紹介し、VI節においてまとめを行う。

---

\* 城西大学女子短期大学部

## II. 大学における情報リテラシー教育

大学における情報リテラシー教育の内容には、大まかに言って、(a)電子メール・インターネットの活用、(b)ワープロによる文書作成、(c)スプレッドシートによる表計算、(d)パソコンを用いたプレゼンテーションなどがある。これは、高度情報化の現代社会において、パソコンを電子文房具として十分に活用できるスキルを習得するためのものである。しかしながら、最近のインターネットの急速な普及に伴い、情報リテラシーの内容も少しずつ変化してきている。すなわち、これまでは上記の(b)、(c)などのスキルを習得させることに重点が置かれていたが、インターネットの急速な普及に伴いインターネット上でのトラブルや犯罪が増加していることに対応すべく、著作権の問題、すなわちインターネット上の他人の著作物の取り扱いの問題やネチケット\*などに関する情報倫理教育<sup>2)</sup>の重要性が指摘されるようになってきた。つまり、道具として使いこなすだけでなく、その使い方は是非や正当性、安全性についても同時に留意することが重要になってきている。

情報リテラシー教育は、「自動車の運転の教習」にたとえられることがある。つまり、電子メールやインターネット、ワープロなどは、ボタンの操作方法や入力のやり方を一度覚えてしまえば、だれでも簡単に使うことができる。これは、自動車の運転と同じである。しかしながら、情報発信が簡単にだれでもが出来るようになった現在では、操作方法だけの教育では片手落ちな教育である。すなわち、インターネット上における著作物の取り扱いの問題、ネチケットやセキュリティなどに関する情報倫理教育が非常に重要である。これは、自動車の運転では道路交通法など法令を守らなければならないのと同様に、インターネット上でのルールやネチケット、その他の諸規則を守ることが必要であるからである。今後は情報リテラシー教育における情報倫理教育の比重が益々大きくなると考えられる。

## III. プログラミング言語教育の意義

大学・短大でのプログラミング教育は、従来、理科系の学部・学科を中心に行われてきた。その代表的なプログラミング言語はFORTRAN (FORmula TRANslator) である。FORTRAN の歴史は古く、1956年に開発された。その後1977年 (FORTRAN-77) と1990年 (FORTRAN-90) に大幅にバージョン・アップされてきた。本来、FORTRAN は大型計算機用に開発された科学技術の数値計算を行うためのプログラミング言語である。言語としては、最も古くに開発されたものであるが、40年以上にわたって蓄積された資産 (プログラム・ライブラリ群) があり、今でも多くの教育・研究機関で使用されている。

---

\* インターネット上での守るべきエチケットのことをさす。

大型計算機からエンジニアリング・ワークステーション (EWS) へとダウンサイジングするにつれ、UNIX をオペレーティング・システム (OS) とする EWS で C 言語が使われるようになってきた。C 言語は、1972 年に開発され、その後、C++へと拡張され、現在では、多くのパソコンや EWS で使用されている。また、最近ではグラフィカル・ユーザー・インターフェース (GUI) をもった Visual C++が Windows プログラミング用の言語として開発され、多くのユーザーに用いられている。

これに対して、文科系の学部・学科におけるプログラミング教育の言語としては、BASIC (Beginner's All purpose Symbolic Instruction Code) が広く使われてきた。BASIC は 1965 年に、初心者でも習得が容易なプログラミング言語として開発された。MS-DOS を OS とするパソコン上で動作する言語であり、その後多くのアプリケーション・プログラムが BASIC を用いて教育・研究用に開発されてきた。その後、1990 年代に入り、Windows プログラミング用の言語として Visual Basic が開発され、Delphi その他の Windows プログラミング言語とともに、現在広く使われている。

さて、ここで大学・短大の情報教育におけるプログラミング教育の位置付けを考えてみたい。大型計算機の時代あるいは初期のパソコンの時代は、市販のアプリケーション・ソフトはほとんどなく、何をするにも、自分で開発するほかはなかった。したがって、教材用のソフトや研究テーマの計算などは、すべて FORTRAN や BASIC などを用いて開発する必要があり、プログラミング言語教育が中心的課題であった。

しかしながら、現在の状況は一変しているといえる。すなわち、非常に多くのアプリケーション・ソフトが市場にあふれている。このため、ある仕事をするために苦勞して FORTRAN, BASIC や COBOL でわざわざプログラミングしなくても、大抵の計算は EXCEL などの表計算ソフトで可能である。以前であれば、苦勞して計算結果をグラフ化していたものが、EXCEL を用いれば、きれいなグラフを一瞬のうちにプロット出来てしまう。帳簿管理やデータ・ベースの作成なども同様である。では、このように市販のアプリケーション・ソフトが豊富な現在、大学教育におけるプログラミング教育の存在意義はあるのだろうか。もしあるとすれば、それは何であろうか。

近年、就職に有利であるとのことから、各種の検定の資格取得が強調されている。本学女子短期大学の情報教育においても、ワープロ技能、表計算技能などの各種の検定試験がおこなわれ、上級の資格取得をめざし、学生は頑張っている。筆者は過去数年間にわたり、女子短期大学のプログラミング言語教育にたずさわっている。学生から、「この授業を受けるとどんな資格が取れますか。」とよく聞かれる。資格取得に直結した授業は、その目的がはっきりしているが、プログラミングのように直接資格と結びつかない授業は敬遠されがちである。筆者は、こんな時に「プログラミングの授業は創造性と考える力を養う授業である。」と説明することにしている。II 節で、情報リテラシー教育を「自動車の運転の教習」に例えたが、プログラミング言語教育を「自動車の製作

実習」に例えることが出来る。すなわち、さまざまな自動車部品を使って、一台の自動車を組み立てるが、プログラミングでは、フォームやコマンド・ボタンなどの種々のオブジェクトを使って一つの作品を作るのである。一つひとつのオブジェクトの名称や用途を知り、それらを組み合わせて課題を制作する。全体のデザインの設計から組み立て、デバッグと一連の作業を通して作品が完成する。もちろん、市販の専用ソフトに比べれば見劣りがするが、立派な手作りのソフトが出来上がる。トヨタや日産などの車にはない、手作りのよさがそこにはある。また、プログラム中に隠れているエラーを長い時間をかけてやっと見つけた時の達成感が大きい。

さて、大学教育の本質的な部分は単なる知識の暗記ではなく、思考する能力を養うことにある。その意味では、プログラミング言語教育は、作品の制作を通して「創造力と考える力」を養い、その存在意義は現在も非常に大きいと考えている。

#### IV. Visual Basic によるプログラミング言語教育の事例

さて、プログラミング言語教育の目的と意義についてはⅢ節で述べた。ここでは、筆者が短期大学部で実践しているプログラミング演習を例にとり、その内容について述べたい。はじめに、シラバスを以下に示した。

● 授業名：プログラミング演習 通年，2 単位

● 授業の目的・目標：

この授業では、情報処理の基礎となるプログラミング言語を学習する。BASIC および、Visual Basic（以下 VB とする）についての詳しい学習および演習を行う。前期は基礎的な Windows プログラミング、後期は、より高度な Windows プログラミングをめざす。

● 授業の内容：

前 期

回	項 目	内 容
1	プログラミングの基礎知識 1	アルゴリズムとフローチャートの概念
2	プログラミングの基礎知識 2	フローチャートの作成
3	プログラミングの基礎知識 3	フローチャートの課題作成
4	VB の基礎知識 1	VB の起動と画面の構成
5	VB の基礎知識 2	BASIC と VB の違い (VB の特徴)
6	VB の基本操作 1	プロジェクトのロードとフォームの作成
7	VB の基本操作 2	各種コントロールの学習
8	VB の基本操作 3	プロパティの設定
9	基礎的プログラミング 1	作品の立案とフォームの作成
10	基礎的プログラミング 2	コーディング
11	基礎的プログラミング 3	デバッグとコンパイル
12	課題作成 1	タイマーコントロールを使った課題作成
13	課題作成 2	課題の完成，提出

## 後 期

回	項 目	内 容
14	基礎的プログラミング 4	簡単な MDI プログラムの作成
15	基礎的プログラミング 5	MDI インターフェイスの使い方
16	基礎的プログラミング 6	ファイルメニューの作成
17	課題の作成 1	指定された課題の作成
18	課題の作成 2	課題の作成と提出
19	実践的プログラミング 1	シミュレーションプログラムの作成
20	実践的プログラミング 2	乱数の発生のおもしろさ
21	実践的プログラミング 3	アニメーションの作成
22	課題の作成	指定された課題の作成と提出
23	応用プログラミング 1	自由テーマの課題の立案と設計
24	応用プログラミング 2	フォームの作成とコーディング 1
25	応用プログラミング 3	コーディング 2
26	課題の作成	自由課題の提出

はじめに、アルゴリズムとフローチャートについて学習する。これらは、プログラミングの基本であり、プログラミング言語として何を採用するかとは独立で普遍的な概念であり、もっとも重要なテーマの一つといえる。

次に Visual Basic の基本構成について学習する<sup>3)</sup>。Visual Basic では、BASIC や FORTRAN、COBOL などと異なり、グラフィカル・ユーザー・インターフェース (GUI) に対応した Windows のためのプログラミング言語であり、これまでの言語とはかなり違った特徴を持っている。すなわち、Visual Basic のプログラムは次の 3 つの部分から成り立っている。

- (1) フォームと呼ばれる GUI とフォーム上に配置された種々のコントロール群 (これらはオブジェクトと呼ばれる)。
- (2) それらのオブジェクトに対する、プロパティと呼ばれるパラメータ値群。
- (3) それらのオブジェクトごとに記述された BASIC コード (プログラム)。

従来の BASIC、FORTRAN、COBOL では、ソースコードと呼ばれる部分 ((3) の部分) だけから成り立っており、これらの基本構成に関する概念は Visual Basic でプログラミングを行う上での最も基本的なものである。

さて、個々のコントロールに関する名称やその特徴と用途を理解した上で、例題を通して、実際のプログラミングを行う。Visual Basic によるプログラミングは次のような 5 つのステップで行われる。

- (1) 仕様の立案と設計……紙の上で、デザイン的设计。
- (2) フォームの作成……Visual Basic のフォーム上で各種コントロールを配置。
- (3) プロパティの設定……配置したオブジェクトのプロパティを設定。
- (4) コーディング……オブジェクトごとに BASIC コードを記述。
- (5) デバッキング……プログラムを実行し、プログラム・エラー (バグ) の修正。

特に、(4) のコーディングに際しては、プログラム全体の流れを把握するために、あらかじめ前述

のフローチャートを作成し、それに基づきプログラミングを行う必要がある。

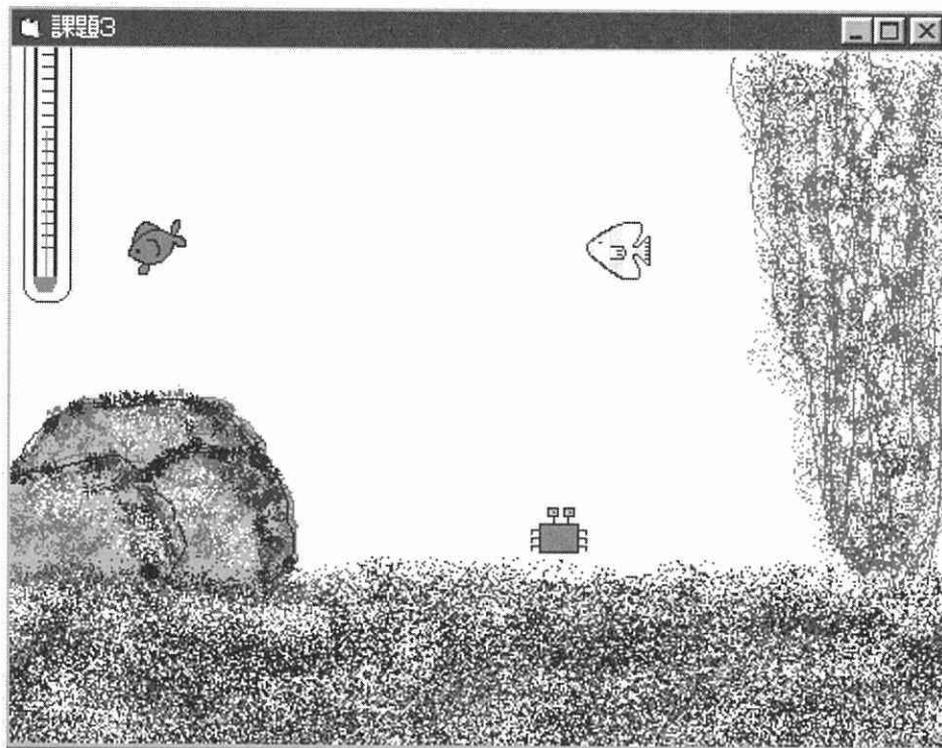
## V. Visual Basic による作品例

本節では、実際に行われた具体的な作品の例を紹介する。プログラミング演習では、課題作品の制作を通して授業を行っている。例年、前期に3つ、後期に3つの課題を制作している。本節で紹介する課題は後期の最後の課題であり、1年間で学んだプログラミング学習の総まとめ的なものである。プログラミングの内容は、かなり高度なものになっており、ソースコードのサンプルをプリントとして、学生に配布しておく。各オブジェクトの機能などの基本的なことを説明した上で、作業に取り掛かる。学生はIV節で述べた5つのステップに従って作業を行う。フォームのデザインや設計など、多くの所で学生一人ひとりの個性があらわれる。絵を描くことが得意な人とそうでない人。また新しいアイデアを考えて作る人、サンプルどおり作る人など。千差万別なものが出来上がる。

### ● 水槽の中で泳ぐ魚たちのシミュレーションソフトの作成

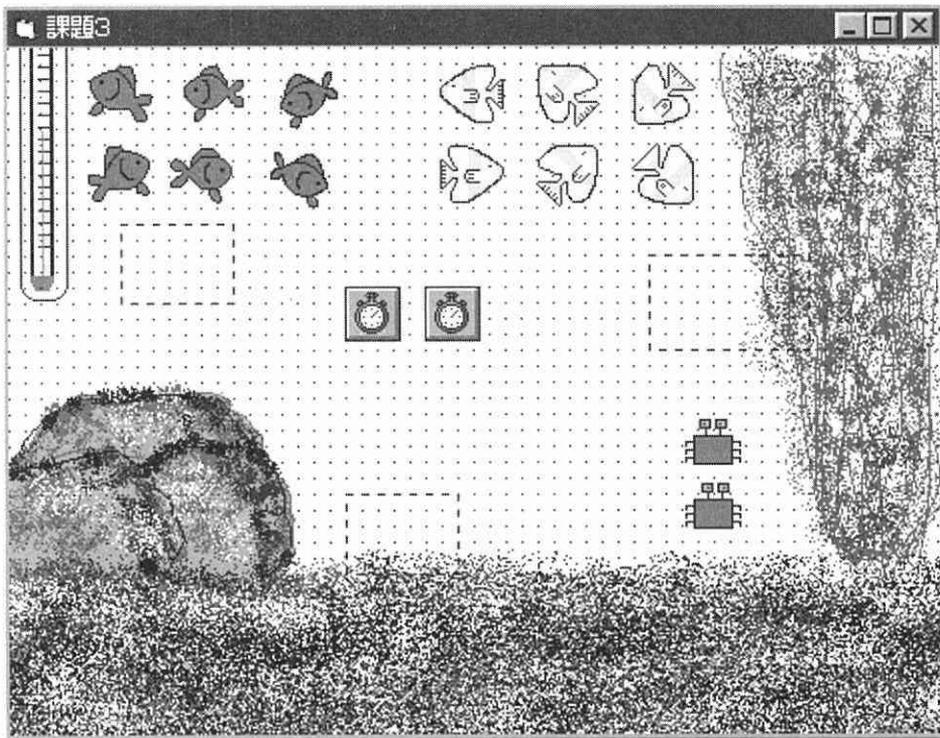
ここでは、完成例のような水槽の中で魚と蟹が自由に泳ぐ様子をシミュレートしたプログラムを作成する。

完成例



## (1) フォーム上のコントロールの配置

このプログラムでは擬似乱数を発生させ、その乱数に応じて、それぞれの魚は6通りの方向、蟹は左右の2通りの方向に動く。これによって、水槽の中の魚や蟹の動きをシミュレートしたプログラムである。フォーム上では、下のように、水槽の絵があり、それぞれの魚の動きにあわせたピクチャー・コントロールが14個(=6+6+2)配置されている。また、動く魚を格納するためのピクチャー・コントロールが3個(点線の四角)配置されている。さらに、2つのタイマー・コントロールが配置されている。水槽は、ペイントで作成したビットマップを貼り付け、魚や蟹は、アイコンエディタで作成したアイコンを貼り付けている。



## (2) フォームのコードの内容

```
Private Sub Form_Load( )
    Width=463*Screen.TwipsPerPixelX+Width-ScaleWidth
    Height=345*Screen.TwipsPerPixelY+Height-ScaleHeight
    kingyright.Visible=False
    kingyrightup.Visible=False
    kingyrightdown.Visible=False
    kingyoleft.Visible=False
```

```
kingyoleftup.Visible=False
kingyleftdown.Visible=False
  sakanaright.Visible=False
  sakanarightup.Visible=False
  sakanarightdown.Visible=False
  sakanaleft.Visible=False
  sakanaleftup.Visible=False
  sakanaleftdown.Visible=False
kaniright.Visible=False
kanileft.Visible=False
  timerrnd.Interval=800
  timerkingyo.Interval=80
kingyo.Picture=kingyleft.Picture
sakana.Picture=sakanaright.Picture
kani.Picture=kaniright.Picture
flag=0
Randomize
End Sub
```

### (3) タイマー・コントロール1のコードの内容

```
Private Sub Timerkingyo__Timer( )
  Static dx1 As Integer, dy1 As Integer
  Static dx2 As Integer, dy2 As Integer
  Static dx3 As Integer, dy3 As Integer
  If kingyo.Left < 0 Then n=1
  If kingyo.Left+kingyo.Width > ScaleWidth-800 Then n=3
  If kingyo.Top < 0 Then n=0
  If kingyo.Top+kingyo.Height > ScaleHeight-800 Then n=2
  If sakana.Left < 0 Then n=7
  If sakana.Left+sakana.Width > ScaleWidth-800 Then n=9
  If sakana.Top < 0 Then n=6
  If sakana.Top+sakana.Height > ScaleHeight-800 Then n=8
  If kani.Left < 0 Then n=12
```

```
If kani.Left+kani.Width > ScaleHeight-800 Then n=13
```

```
  If flag=0 Then
```

```
    dx1=-15: dy1=0
```

```
    dx2=20: dy2=0
```

```
    dx3=30: dy3=0
```

```
  Else
```

```
    Select Case n
```

```
      Case 0
```

```
        kingyo.Picture=kingyorightdown.Picture
```

```
        dx1=15: dy1=12
```

```
      Case 1
```

```
        kingyo.Picture=kingyright.Picture
```

```
        dx1=15: dy1=0
```

```
      Case 2
```

```
        kingyo.Picture=kingyoleftup.Picture
```

```
        dx1=-15: dy1=-12
```

```
      Case 3
```

```
        kingyo.Picture=kingyoleft.Picture
```

```
        dx1=-15: dy1=0
```

```
      Case 4
```

```
        kingyo.Picture=kingyorightup.Picture
```

```
        dx1=15: dy1=-12
```

```
      Case 5
```

```
        kingyo.Picture=kingyoleftdown.Picture
```

```
        dx1=-15: dy1=12
```

```
      Case 6
```

```
        sakana.Picture=sakanarightdown.Picture
```

```
        dx2=20: dy2=12
```

```
      Case 7
```

```
        sakana.Picture=sakanaright.Picture
```

```
        dx2=20: dy2=0
```

```
      Case 8
```

```
        sakana.Picture=sakanaleftup.Picture
```

```
dx2=-20: dy2=-10
```

```
Case 9
```

```
sakana.Picture=sakanaleft.Picture
```

```
dx2=-20: dy2=0
```

```
Case 10
```

```
sakana.Picture=sakanarightup.Picture
```

```
dx2=20: dy2=-10
```

```
Case 11
```

```
sakana.Picture=sakanaleftdown.Picture
```

```
dx2=-20: dy2=10
```

```
Case 12
```

```
kani.Picture=kaniright.Picture
```

```
dx3=30: dy3=0
```

```
Case 13
```

```
kani.Picture=kanileft.Picture
```

```
dx3=-30: dy3=0
```

```
End Select
```

```
End If
```

```
flag=1
```

```
kingyo.Move kingyo.Left+dx1, kingyo.Top+dy1
```

```
sakana.Move sakana.Left+dx2, sakana.Top+dy2
```

```
kani.Move kani.Left+dx3, kani.Top+dy3
```

```
End Sub
```

(4) タイマー・コントロール 2 のコードの内容

```
Private Sub timerrnd_timer( )
```

```
    n=Int(14*Rnd)
```

```
End Sub
```

(5) (General) (Declarations) のコードの内容

```
Dim n As Integer, flag As Integer
```

学生は、上のソースコードの内容をすべて理解しているわけではない。しかしながら、基本的な

事項, 例えばタイマー・コントロールの機能, シミュレーションあるいは擬似乱数の意味, イメージコントロールの概念などを理解している。課題制作では, それらのオブジェクトの意味を確認しながら制作を行う。

しかしながら, 入力したプログラムを実行すると大部分の学生はソースコードにエラーが発生する。そのエラーはコントロール名やコマンド名などのちょっとしたスペル間違いであることが多い。また, 学生のプログラムには出発点のアルゴリズムそのものに欠点があるものもある。このようなプログラム・エラーのデバッキングにかなりの時間と労力を費やす学生がほとんどである。プログラミングの授業は, こうした作業の繰り返しである。このプロセスの中で創造性と考える力を養うことができるのではないだろうか。

最後に, 完成した学生の作品の中には, こちらが予想していない独創的なものがあり, 感心させられることがある。その時は新しい発見であり, こちらも大変嬉しくなるものである。

## VI. ま と め

Ⅱ節では, 大学・短大における情報リテラシー教育の現状について考察した。情報教育の内容には大きく分けて, (1)情報リテラシー教育, (2)プログラミング言語教育, (3)情報処理に関する専門教育があげられる。Ⅲ節では, 特にプログラミング言語教育の重要性と意義について述べた。プログラミング言語教育は, 「創造性と思考する力」を養う意味で非常に存在意義の大きいものであることが述べられた。また, Ⅳ節では, 筆者のプログラミング演習を事例として, Visual Basic によるプログラミング言語教育の方法についての詳しい解説を行った。最後に Ⅴ節において, Visual Basic プログラミングの作品例として「水槽の中で泳ぐ魚たちのシミュレーションプログラム」の紹介を行った。

### 参考文献

- 1) 社団法人 私立大学情報教育協会編：短期大学情報教育モデル（1999年度版）
- 2) 社団法人 私立大学情報教育協会編：インターネットと情報倫理（1999年度版）
- 3) 川口輝久, 河野勉著：かんたんプログラミング Visual Basic, 技術評論社