

情報科教育法 II 2003 # 13

久野 靖*

2003.2.4

はじめに

半年に渡っておつき合い頂いた「情報科教育法 II」も今回で最終回となりました。今回は最初に「ディスカッション」を取り上げ、前回追加課題で出してもらったテーマで少し掲示板での議論を体験してみたいと思います。その後「教具としてのソフトウェア」を取り上げ、その部分で前回ほとんどやれなかった Java 言語も扱っておきます。

1 ディスカッションとグループコミュニケーション

1.1 ディスカッション

「情報」の授業の中でディスカッションを行うこと、グループ単位で意見を取りまとめることは多く出てくる。議論を通じて考え方を洗練して行くのは「情報」の内容の一環だとも言えるが、より広い「生きる力」にも通じるという意味で重要なことだと思う。

しかし、ただ「ではディスカッションしましょう」だけでうまく進むのか？ それはかなり疑問である。どうということが問題になると思うか？ 「日本人は議論が苦手」とよく言われるがそれはなぜか？

- 「全員一致で異論がないのが美しい」という文化 (?) がある。
- 「異論を述べる」とそれだけで「喧嘩を売っている」と思うような土壌がある。
- そのため、テーマを設定しても何も異論が出ずに議論にならない。
- 一方で、異論が出ると感情的な対立 (喧嘩) になりやすい。
- その原因としては、意見/見解とそれを述べている人間とを区別して見る訓練ができていないことが挙げられる。
- このため、複数の対立意見に基づいて合意点を見出すといった作業がうまく行かない。

ではどうしたらうまく議論が進められるようになるか？ 簡単な回答はないが、たとえば次のようなことはヒントになるかと思う。

- テーマとして、論点をはっきりしているもの、賛否両論が (誰の目にも) 存在するようなものをうまく選ぶ。
- 開始時に、誰はどのような主張かを明確にしてから始める。
- 議論により多くの論点を見出すことが目的であり、相手をやりこめることは目的ではないことを宣言する。
- 論点について反論し、それを述べている人の行為等を問題にしないよう釘を刺す。

あと、うまい議論を実際に見ることもよいきっかけとなる。その意味では、活発に意見を述べることができ、上記のような態度ができていない人を選んで「パネルディスカッション」をやるのもいいかも。

*筑波大学大学院経営システム科学専攻

1.2 グループコミュニケーションの活用

ディスカッションを「生で」やってもらうのもいいが、コンピュータを活用して(グループコミュニケーションツールを使って)やってもらう方法がある。具体的な手段としては(前にも挙げたが)次のものがある。

- メール、メーリングリスト
- 掲示板
- チャット

このうち、メールは普段皆が使っているのでソフトに対する習熟がでているという利点があるが、メッセージが長めなので打つのが大変という問題がある。いっぽう、チャットは口数が勝負なのでまじめな意見を述べるのはやりにくいという問題がある。こうして比べてみると、掲示板システムが利用できれば、まじめな議論にはいちばんいいかと思う。

そもそも、コンピュータ上のツールを使うことで、どのような利点があるだろうか？

- 誰でも(口下手とかに比較的關係なく)打ち込めば議論に参加できる。
- 多数の人が同時に発言してもそれなりに議論が進む。
- 「議事録」が自動的に残るので後から発言を検討できる。

とくに最後の点を活用して、たとえばしばらく議論してもらった後で、その議論経過をまとめる演習をする、喧嘩などがあつたらその原因はどこにあつたか皆で検討する、などの活用方法はどうか。

もちろん、弱点についても注意を払う必要がある：

- タイピングが遅い人が不利になりやすい → 時間になるべく余裕を持たせ、速い人がフルスピードで打たないように場の進行を調整する
- 何もしないで読むだけの人が見過ごされがち → 誰がどれだけ発言しているか注意を払う、人数が多すぎる場合はいくつかのグループに分けて議論を行わせる
- 「ふざけ」「いたずら」「おちゃらけ」などが出やすい → そのようなものが見られたら生で注意を促す

いずれも、本来の掲示板なら難しいけれど、教室で一同が会して掲示板を使っているのなら教師が巡回したり教室全体に呼びかければ済む。

あと書き込みを匿名にするかどうかもちよつと迷うところ。

- 匿名の方が生徒の本音が聞けるかも知れない
- その半面、いたずらが出やすい。実名の方がまじめにやってもらえる。
- 折衷案として、ニックネームを決めさせるとか？

1.3 実習: 掲示板での議論

では実際に、前回の追加課題としてやってもらった「情報の授業はどうあるのがいいか？」をテーマとして、このクラスの掲示板上で議論をやってみていただきます。書き込み内容が直接は表示されない(リンクを選択したときに見える)タイプの掲示板なので、タイトルを適切につけるように注意してやってください。

2 教具としてのソフトウェア

2.1 教具とは？

教具というのは要するに「教えるのに使う道具」で、たとえば数学であれば黒板で使う巨大な三角定規やコンパスとか、あと小学校であれば生徒にそれぞれ持ってもらう「算数セット」(おはじきとか一杯詰まった奴?)なども相当するかと思う。英語だったらネイティブが教材を読み上げているテープや CD(と、それを聴かせるためのプレーヤ)、社会科だったら巨大な地図とかだろうか。

で、物理や化学などの教師になると、物理実験や化学実験のための仕掛けを製作したり、それを生徒に使わせても大丈夫なように工夫したり、実験の手引を作ったりするのも重要な仕事になるわけである。ガラス管をバーナーで熱して曲げたり加工することもあるだろうし…

では、教科「情報」では何が教具に相当するだろう？ もちろん、コンピュータやネットワークを使って実習をさせるのだから、コンピュータのハードウェアや、ネットワークの接続機器が「教具」だと言えなくはない。しかしそれは一度設置してきちんと管理すれば、何にでも使えるし授業のたびに工夫して手直しすることもないので、あまり教具らしくはない。

実は、教科「情報」で一番教具らしいのはソフトウェアだ、というのが自分の考えである。具体的には…

- 非常にさまざまな種類のものがあり、
- 教える内容に応じて使い分ける必要があり、
- 1つのソフトウェアでもどのように使うか使い方を教える内容ごとに工夫しなければならず、
- 場合によっては特定内容を教えるためのソフトが専用にあることもあり、
- さらに場合によっては教員がそのようなソフトを作ることもある。

もう少しこのさまざまな場合について検討してみよう(「情報科教育法」の第3章も読んでおいて頂きたい)。

2.2 教具としてのソフトウェアの分類

ソフトウェアを「情報」などの科目で使う場合は、まず次の4通りに分類できる。

- (1) 教材提示・学習用ソフトウェアによる学習(CAI、CMI)
- (2) 汎用のソフトウェアを使う
- (3) 専用のソフトウェアを使う
- (4) 生徒にソフトウェアを作らせる/修正・加工させる

これらについてそれぞれ見てみよう。

2.3 教材提示・学習用ソフトウェア

これは一般に CAI(computer assisted instruction)、CMI(computer mediated instruction)と呼ばれる。意味は「コンピュータによる授業補助」で、先生が授業する代わりにコンピュータにやらせておいて先生は机間巡視でもしていればいい、という感じ。

このような教材を作るソフトウェアもいろいろ研究されている。内容的にはスライドにより教材を順番に提示し、学習確認のため選択肢問題が表示され、OKだと先へ行くがNOだと復習や補足説明が出てくるといった感じ。しかし、やっぱり先生が教えてこそその授業だという意見も多く、あまり普及していない。(どう思いますか?)

2.4 汎用ソフトウェア

教科「情報」では汎用のソフトウェアを活用することは沢山ある。

- ブラウザ→情報の取得、検索などに
- プレゼンソフト→プレゼンテーションの作成と実施
- ワードプロ→文書の作成、取りまとめ、レポートなど
- 表計算→簡単なシミュレーションやデータ操作、分析など
- ペイントソフト→画像や色の原理など
- ドローソフト→図形の表現など
- サウンドソフト→音の表現など

汎用ソフトはさまざまな目的に使えるし、それなりに機能もしっかりしているのでいいのだが、汎用だけに演習の目的以外にも使えてしまう、そのためにきちんと演習の筋立てを確立させてから使わないとなんとなく操作しただけで終わってしまう、という問題がある。

3 模擬授業

今回の模擬授業は「情報 C」 p84~85 だそうです。担当の方はよろしく。

4 教具としてのソフトウェア (2)

4.1 専用ソフトウェア

専用ソフトウェアとは、ある特定のテーマについて学習するために、そのための機能のみを持つソフトを使用すること。そのようなものはできあいで売っている (またはフリーで配られている) こともあるが、自分のクラスで使うのに「ぴったり」しているかどうかが問題。その点で、一番いいのは担当教員が作って用意すれば、その授業のニーズにぴったり (できればそれくらいのソフトウェア作成能力を持った人に「情報」教員になって欲しい)。

たとえば、「文字はコンピュータの中では数値 (2 進数) で表わされている」ことを学ばせたいとする。そのことをただ説明するより、普段使っているテキストファイルの中身を 2 進数で表示させるプログラムを使わせてみたらどうだろう? 簡単な Java プログラムを示す。

```
import java.awt.*;
import java.awt.event.*;

public class Sample03 {
    public static void main(String[] args) throws Exception {
        int ch;
        while((ch = System.in.read()) >= 0) {
            for(int i = 0; i < 8; ++i) {
                int mask = 1 << (7-i);
                System.out.print(((ch&mask)==0)?'0':'1');
            }
            System.out.println();
        }
    }
}
```

これを使って「abcde」というファイルの内容を表示させてみる。

```
% cat test.txt
abcde
% javac Sample03.java
% java Sample03 <test.txt
01100001
01100010
01100011
01100100
01100101
00001010
%
```

では日本語の「あいうえお」だとうどうだろう？

```
% cat test1.txt
あいうえお
% java Sample03 <test1.txt
10100100
10100010
10100100
10100100
10100100
10100110
10100100
10101000
10100100
10101010
00001010
%
```

これは EUC コードだけど SJIS だとまた違ったようになる。さて、これで何が分かると思うか？ どういう演習課題を出すのがいいか？

4.2 生徒にプログラミングをさせる

先の回にやったように、生徒にプログラミングをさせることは、コンピュータの動作原理に対する理解を養う上では極めて効果的な方法。ただし、専門教科レベルでかなり時間を使わないと難しいという面がある。

別の方法として、ゼロから書くというより、とりあえず動くものを与えて手直しさせる程度の方がつまづきがなく安心)、という方法もある。実はこのクラスではできるだけそういう方法を採用している (皆様の中には HTML にせよプログラミングにせよ得意な人も不得意な人もいるので、手直ししてみる程度でないときつい人も多いから)。そういう方法もそれなりに効果はあるでしょう？

この「手直しさせる方法」の場合は、ある特定のテーマについて学ぶための「教材プログラム」をそのつど用意して、それをいじってもらい、という形になることが多い。うまく用意すれば、プログラムを動かすだけでもそれなりに楽しいので興味を持たせることができる。

4.3 例: マウス操作の原理

「プログラムを直させて実習する」例として、今回はマウス入力とそのプログラムでの扱いの原理を学んでもらうようなものを取り上げる。つまり、「普段我々がマウスを操作すると何か起きるのは、プログラムでそうしてくれているのだ」ということを知ってもらうための題材。この例題は

- (1) マウスボタンを押すと図形が緑になり、マウスの位置に移動
- (2) ドラグすると図形がついてくる
- (3) ボタンを離すともとの色に戻る

というもの。

```
import java.awt.*;
import java.awt.event.*;

public class Sample02 extends Frame {
    Color col = new Color(255, 0, 0);
    int x = 100, y = 100;
    public Sample02() {
        this.setSize(400, 400);
        this.addMouseListener(new MouseAdapter() {
            public void mousePressed(MouseEvent e) {
                col = new Color(0, 255, 0); x = e.getX(); y = e.getY(); repaint();
            }
            public void mouseReleased(MouseEvent e) {
                col = new Color(255, 0, 0); repaint();
            }
        });
        this.addMouseMotionListener(new MouseMotionAdapter() {
            public void mouseDragged(MouseEvent e) {
                x = e.getX(); y = e.getY(); repaint();
            }
        });
    }
    public void paint(Graphics g) {
        g.setColor(col);
        g.fillRect(x, y, 40, 40);
    }
    public static void main(String[] args) {(new Sample02()).setVisible(true);}
}
```

これらを実現するには、マウスの押し/離し/ドラグに対してそれを受け取って処理するオブジェクト (アダプタという) を設定する。そのオブジェクトの対応するメソッド中で上記の動作を行う。実際には、アダプタ側では変数の値を適切に書き換えた後で `repaint()` という再描画指示メソッドを呼ぶだけ。`paint()` では変数 `x/y` で指定された座標に、変数 `col` で指定された色で図形を描くので、これらが適切に設定されていれば上記の動作がうまく起きる。

演習 2 上の例題をコピーしてきてそのまま動かせ。

```
cp ~kuno/work/Sample02.java ←ファイルをコピー(最初の1回だけ)
javac Sample02.java ←コンパイル
java Sample02 ←実行、止めるときはControl+C
```

動いたら次のように直してみよ。

- a. マウスをクリックしたとき、その位置より「20ピクセル右」に矩形が移動するように直す。(ヒント: `x` を `e.getX()` より 20 大きい値に設定する。)
- b. ドラグしたとき、マウスの位置ではなくあさっての方にドラグされるようにする。(ヒント: マウスイベントの `getX()` と `getY()` の結果をそのまま使う代わりに入れ換えたり、適当な計算式で変換して `x` や `y` に設定する。)
- c. ドラグしたとき、位置 200 より右には移動できないようにする。(ヒント: `if` 文を使い、`e.getX()` が 200 より大きかったら何もせずに `return;` させる。)
- d. 矩形の上でクリックしたときだけドラグできるようにする。(ヒント: `e.getX()` が `x~x+40` の範囲、かつ `e.getY()` `y~y+40` の範囲にあるときだけ、`x` と `y` を設定する。)
- e. 「掴んだ位置」をそのまま維持してドラグできるようにする。(ヒント: `e.getX()` と `x` の差、`e.getY()` と `y` の差を覚えておいてそれだけずれた位置に移動させていく。)
- f. その他自分の好きな動き方の工夫。

演習 3 先の `Sample03.java` もコピーしてきてみて、適当なファイルを食べさせて動かしてみる。

5 チェックリストと自己評価

付録 A に「教科『情報』の教員に求められる事項チェックリスト」というものがあります。これは `kjm` メーリングリストに太田剛さんが投稿されたものを元に久野が改訂したものです。別紙として同じものを用意しましたので、こちらで自分がクリアしていると思う項目に「○」をつけて見てください(自己評価)。「○」が少なくても悲観する必要はないです。これが全部できる人は日本中で数名もいないと思いますから。なるべく多く「○」になるようにこの科目も含めて頑張りましょう、ということ。

6 最後に

以上で「情報科教育法 II」の内容はすべておしまいです。時間の都合上、取り上げられなかったことも沢山ありますが、とりあえず皆様に「情報の先生として授業をしたらこんな感じであり、それをうまくやるには自分がどういうことを学んで行く必要があるか」は分かって頂けたのではないかと思います。「情報の先生にならない人でも、このことはいろいろな形で役立ててもらえるのではないかと思います。では半年間のおつき合い、ありがとうございました。(おしまい)