

# 情報科教育法II 2003 # 12

久野 靖\*

2003.1.28

## はじめに

前回、センター試験「情報関係基礎」をやってみてもらいましたが、皆様の感想を見ると結構楽しまれたようです。ただ、楽しむのが目的ではなくどういう問題がいいかを考えて欲しいですね。今回は「試験と評価」の続きと、あと今回からちよつとだけ Java 言語をやります。

## 1 教科「情報」における試験と評価 (2)

### 1.1 「考える問題」について

前回のセンター試験「情報関係基礎」の大問1の問3(「おふだ」の問題)について、「これが『情報』の試験なのか?」という疑問を持った人もいるかも知れない。しかし考えて欲しいのだけど、もしそれより前の穴埋め問題が中心だったとすると、知識を持っているかどうかの比重があまりに高くなってしまいませんか? 数学などの試験でも、単に計算ができるというより、数学の原理を身につけてさまざまな形で応用しつつ問題を解けるかどうかを見ることが重要だと思いますよね? だとすれば、教科「情報」でもそのような方向での問題をぜひ取り入れて欲しい。

具体的には、そのような「考える」ことを要求する部分としては次の箇所がある。

- 問題解決 — 情報 A、情報 B
- アルゴリズム — 情報 B
- 情報の表現 — 情報 A、情報 C
- 情報の伝達 — 情報 A、情報 C

たとえば、模擬授業の中に「暗号の仕組み」があつが、試験問題としてシーザ暗号よりは高度な「特定の暗号化/複号のしくみ」を説明したあと、平文を提示して暗号文を回答させる、暗号文を提示して平文を回答させる、などのテーマであれば、かなり「考えてもらう」ように構成できるはず。今回模擬授業の「情報の圧縮」なども同様。

また、このように特定のテーマを取り上げるのでないとしても、問題文である枠組みについて説明して、それを解く方法を考える問題(センター試験のオートマトンの問題のようなもの)も、考える力を見るという点ではあつてよいはず(もともと試験とは考える力を見るものだという姿勢)。

## 2 問題をやる

皆様に宿題として試験問題を考えて来て頂いたので、実際に相互に試験を実施して頂く。

---

\*筑波大学大学院経営システム科学専攻

### 3 模擬授業

今回の模擬授業は「情報 C」 p36～57 だそうです。担当の方はよろしく。

### 4 具体的な試験問題案と検討

#### 4.1 試作問題とは…

久野ほか有志で「もしセンター試験に教科「情報」の問題があるとしたら、どんな問題になるだろうか」仮に作ってみたもの。目標としては、選択肢や穴埋め問題であっても「単に知識を問う」のではなく「ものごとの原理から理解していないとできない」「実際に実習を行った体験に根ざしていないと分からない」ようにすること。その目標は成功したでしょうか？ ちょっと試してみてください。

#### 4.2 (試作問題) 電子メール

電子メールに関して述べている次の文章のうちから、間違っただ内容のものを 3 つ選んで回答欄に記入せよ。

- A) 電子メールの利点の 1 つは、受け手がコンピュータの前に座っていなくてもメッセージを送れることであるが、受け手のコンピュータの電源が入っていないと、送ったメッセージがエラーになって返送されてしまう。
- B) 電子メールのメッセージがエラーで返送されてくるのは、送った相手が引っ越すなどしてそのメールアドレスが有効でなくなってしまった場合に限られる。
- C) 電子メールもコンピュータの上では単なるファイルとして格納されているので、メールサーバの管理者などがたまたまメッセージの内容を読んでしまうことはあり得る。
- D) 正しい電子メールのアドレスでは「@」が 2 回現れることはない。
- E) 電子メールアドレスの「@」より右側の部分はドメインアドレスといい、必ず自分が使っているコンピュータを表す名前になっている。
- F) 他人に自分のメールパスワードが知られてしまうと、その人に自分あての電子メールメッセージを読まれてしまうおそれがある。

#### 4.3 解説

この問題の場合、正解できるためには、電子メールの仕組みがどうであるかという原理を知っていること、実際にメールを使っていることが必要であるように工夫したつもり。どうでしょうか？

#### 4.4 (試作問題)HTML ファイルの作成方法

WWW ページの内容を記述する HTML ファイルを作成するとき、テキストエディタ等のソフトウェアで直接 HTML の指定を打ち込む方法と、WWW ページ作成ソフトウェアを使って自動的に HTML を生成させる方法とがある。これらの方法の得失について述べた以下の記述のうちから、間違っているものを 3 つ選び回答書に記号を記入せよ。

- A) WWW ページ作成ソフトウェアを使えば、どのコンピュータで見ても同じ配置で見えるページが作成できる。
- B) HTML を直接打ち込む方法では、打ち込む指定が間違っていると、ブラウザで見てもまったく何も表示されないページができてしまうことがある。

- C) 市販の WWW ページ作成ソフトウェアは、HTML の標準規格に適合したファイルを作成するかどうかの公的検定を受けているので安心である。
- D) 適切な知識があれば、HTML を直接打ち込む方法で、WWW ページ作成ソフトウェアが提供していないような機能を用いたページを作成することができる。
- E) WWW ページ作成ソフトウェアを使えば、リンク先が正しく指定されていなくてたどれない、という不具合は起きなくなる。
- F) ある WWW ページ作成ソフトウェアで用意した HTML ファイルを別の WWW ページ作成ソフトウェアに読み込ませようとしたとき、思った通りの形で読み込まれてくれないことがある。

#### 4.5 解説

これも WWW のページを実際に作ったことがあるという経験、および WWW の原理について理解していることを要求しているつもり。

#### 4.6 (試作問題) 言語規則の帰納

220X 年、宇宙探検隊員のまこと君がとある惑星に降りてみると、そこの原住民は「ほげほげ語」を喋っていた。まこと君は、話されている「ほげほげ語」を聞いて、その規則を作ろうとしてみた。その惑星の人間が話す正しい「ほげほげ語」の会話には、以下のような例があることがわかった。

- A 「ほげほごほご。」
- B 「ほげ。」
- A 「ほげほげほげほご。」
- B 「ほげほご?」
- A 「ほげほげほげほげほげほげほごほご。」
- B 「ほげほごほごほご。」
- A 「ほげほげ。」
- B 「ほげほげ。」

次の選択肢のうちから、「規則」として誤った推定になっているものを 3 つ選び、回答欄に記入せよ。

- 規則 1) 必ず「ほげ」で始まる。
- 規則 2) 「ほげ」の後には「ほげ」か「ほご」が来る。
- 規則 3) 必ず「ほご」で終る。
- 規則 4) 「ほげ」と「ほご」は同じ個数だけ現れる。
- 規則 5) 「ほごほご」のあとは「。」か「ほげ」が来る。
- 規則 6) 文は必ず「。」か「?」で終わらなければならない。

#### 4.7 解説

これは「考える問題」としては簡単な方だと思うが「規則」というメタなものを問うという点では一応思考力を見ているつもり。分かりましたか?

#### 4.8 (試作問題) 問題解決とは?

「問題を解決する」とはさまざまな定義がありえるが、ここでは次のように定義する。

困った(望ましくない)状態が確実になくなるようにすること。

ただし、問題解決を行った結果、新たな問題が発生することもまた当然ありえる。さて、次のような問題(困った状態)があるとする。

学校の屋根つき駐輪場が狭くて、遅れて来ると入れることができず、雨が降ると自転車が濡れてチェーンが錆びてしまう。

この問題に対する(上の定義にかなった)解決方法でないもの下の選択肢から3つ選び、回答欄に記入せよ。

- A) 自転車で通学するのをやめて歩く。
- B) 駐輪場の増設を学校に要望する。
- C) 自転車にグリースを塗る。
- D) 学校の近くの空き家の車庫にこっそり入れる。
- E) 天気予報を聴いて降水確率が0でない日は自転車通学しない。
- F) ゴムベルト駆動の自転車に買い替える。

#### 4.9 解説

問題文は一見常識を問うているようであるが、実は論理を問うている。つまり普段自分が持っている常識から離れて与えられた前提条件に従った判断ができるかどうかを見ている。

#### 4.10 (試作問題) 情報社会の意味

「現代は情報社会である」と言われている。以下に、近年社会で起きているさまざまなことがらを列挙した。これらの中から、「現代が情報社会であることと直接的に関連している事例」として適切なものを3つ選び、回答欄に記入せよ。

- A) 自動車に小児を乗せるときには、チャイルドシートの着用が義務づけられるようになった。
- B) 海外に住んでいても、その日の日本語の新聞を読むことができるようになった。
- C) JRの全国どこの列車の指定席でも、もよりの駅の「みどりの窓口」で購入することができる。
- D) 段ボールや牛乳パックなど、従来はふうのごみとして出していたものを分けて回収するようになった。
- E) 身体にハンディキャップを持つ人でも、パソコンを使って自宅で仕事ができるようになった。
- F) バスや路面電車などで、床が低くて高齢者でも乗り降りの楽なものが増えてきている。

#### 4.11 解説

これは前問と少し似ているが、「情報社会」の定義は示さずに考えさせるところが違っている。つまり「情報社会」とはどういう意味かについてきちんと考えたことがある人はすぐに取り掛かれる、という趣旨。

## 5 Java 言語

### 5.1 Java と JavaScript

前回やった JavaScript のことを「Java」だと思っている人がたまに (実はしばしば) いるが、この 2 つは全然別の言語である。そもそもこんなに似た名前なのがいけないのだが、次のような歴史的事情がある。

- WWW は 1990 年に「発明」されたが、最初はテキスト表示が専門で、全然知られていなかった。ところが、1993 年にグラフィクスが表示できるブラウザ「Mosaic」が公開されると急速に利用が増えた。
- しかし当時のページは HTML のみで「動かない」ページだった。そこに Sun が「アプレット」つまりページの中で絵が動く機構を発明したので (1994 ごろ)、一大センセーションが巻き起こった。アプレットは Java という言語で記述するようになっていたので、「Java ブーム」になった。
- Netscape はそのころ Mosaic 開発チームがスピアウトして興した企業で、急成長していたが、そのブラウザにアプレットを取り込んだのでいっそう急速に Java が普及した。
- Netscape 社はさらに、アプレットとは別に「言語でブラウザを直接制御する」ことを考え、LiveScript という言語を作って公開した。その直後、Java ブームにあやかるために Sun と契約して Java という名前を使うことにして JavaScript という名前に変更した。

両者はどう違うかまとめておく。まず JavaScript は。

- ブラウザに内蔵されて (組み込まれて) いて、ブラウザをいろいろ制御できる (たとえば別のページに飛ばすとか)。
- ページの中身も DOM を使って自由に扱うことができる。
- ただし、各種動作は基本的に「ブラウザを操っている」ので、自分で自由にグラフィクス画面を描くといったことはできない。
- 言語としては、型チェックがなく、小規模なプログラムをささっと書いて動かす/いろいろな部品を組み合わせ活用する「スクリプト言語」。

一方 Java は。

- 特定目的の言語ではなく、汎用の言語。C や C++ のようなもの。
- ブラウザ上で動くけれど、動く領域は「アプレット」と呼ばれあらかじめ指定された四角い領域。
- むしろ、サーバ上でのデータ処理など「見えない」部分でよく使われる。「見える」プログラムでもブラウザとは無関係に普通のプログラムとして扱うことが多い。
- 言語としては、型チェックがあり、大規模なソフトウェア開発に対応するように設計されている。

### 5.2 ごく簡単な例題

とりあえず、お話ばかりでは面白くないので、なるべく簡単ながら実際に動く例を見てみることにする。

```
import javax.swing.*;
import java.awt.*;

public class Sample01 extends JFrame {
    public static void main(String[] args) { // ここから起動
        new Sample01().setVisible(true);
    }
    public Sample01() { // 初期設定
```

```

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); setSize(400, 400);
}
public void paint(Graphics g) { // 画面描画
    g.setColor(new Color(255, 0, 0)); //★
    g.fillRect(100, 100, 60, 40);    //★
}
}
}

```

説明はややこしいができるだけ簡潔にする。

- 最初の `import` は C でいうと `include` のようなもの。
- Java ではすべてのプログラムは「クラス」という単位で用意する。
- クラスの中に「`public static void main(String[] args) { ... }`」というメソッド (C でいえば関数) があると、そこからプログラムが実行される。
- この例題のクラス `Sample01` は「`extends JFrame`」と書かれているので、ウィンドウ (フレーム) を拡張したものになっている。
- `main()` ではその拡張したウィンドウを生成して (`new` の機能)、その生成したものを直ちに表示させている (`setVisible(true)`)。
- Java はオブジェクト指向言語であり、様々なオブジェクト (「もの」) を扱うことができる。オブジェクトにさまざまな動作をさせるには、そのオブジェクトの「メソッド」を呼び出す。その書き方は「オブジェクト.メソッド名 (パラメタ…)」のようになっている。オブジェクトの内側のメソッドでそのオブジェクトを指定するときは `this` と指定するか、オブジェクト指定を省略する。
- `new` の機能でオブジェクトを生成するとき、オブジェクトの初期設定を行う特別なメソッド (コンストラクタ) が実行される。コンストラクタは常にクラス名と同じ名前を持っている。上の例ではコンストラクタはそのオブジェクトである窓が閉じられたらプログラムも終るように設定し、続いて大きさを `400x400` に設定している。
- 具体的な「拡張」とは、この窓に何を表示するかを定めるメソッド `paint()` を追加定義していること。`paint()` は引数として `Graphics` オブジェクトを受け取る。このオブジェクトは画面に色々なものを描くための「ペン」だと思えばよい。
- ここではペンの色をまず「`255,0,0`」(つまり真っ赤) に設定し、続いて「座標 (`100,100`) の位置に幅 `60`、高さ `40` の長方形を塗りつぶして描く」ことを行う。

このプログラムを実行するには次のようにする。

- (1) `cp /home/guests/kuno/work/Sample01.java Sample01.java`
- (2) `javac Sample01.java`
- (3) `java Sample01`

最初の (1) はもちろん例題をコピーしてくる。(2) はソースファイルをコンパイルする。Java では「ソースファイル名は中に入っているクラスの名前の後ろに『.java』をくっつけたものでなければならない」という規則があるので、勝手なファイル名をつけることはできない。(3) はプログラムの実行で、「`java` クラス名」という形で実行開始を指定する。

**演習 1** 上の例題をコピーしてきてそのまま動かせ。動いたら「//★」のところをいじって次のように直してみよ。

- a. 四角形の色や形を変更する。

- b. ループを使って複数の四角形を少しずつずらしながら描く。
- c. 上に加えて、色も少しずつ違った色にしてみる。
- d. 四角形以外の図形を描いてみる (次の説明を参照)。
- e. その他自分でかっこいいと思う絵を描くようにする。

### 5.3 Graphics クラスのさまざまなメソッド

上の演習 d. をやるために、変数 `g` に入っているグラフィクスオブジェクトに対して指定できるもののバリエーションをまとめておく。

- `g.setColor(new Color(赤, 緑, 青))` — 色を設定。RGB 値はそれぞれ 0~255。
- `g.setColor(new Color(赤, 緑, 青,  $\alpha$ ))` — 色を設定。RGB 値に加え、透明度を表す  $\alpha$  (これも 0~255) を指定できる。半透明な色で塗ると下の色が透けて見える。
- `g.fillRect(X, Y, 幅, 高さ)` — 長方形を塗りつぶす
- `g.fillOval(X, Y, 幅, 高さ)` — 楕円を塗りつぶす。この場合の楕円のは、長方形に内接する楕円となる (だからパラメタの意味は上と同じ)。
- `g.drawLine(X1, Y1, X2, Y2)` — 線を引く
- `g.fillPolygon(new int[] {X1, X2, X3}, new int[] {Y1, Y2, Y3}, 3)` — 三角形を塗りつぶす。もっと頂点数を多くしてもよい。

### 5.4 図形をドラグする例題

単に図形を表示するだけでもそれなりに「プログラムと画面の対応」は分かるが、やはりマウス操作などがあると「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」ということが分かりやすい。そのような例題を見てみよう。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 javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Sample02 extends JFrame {
    public static void main(String[] args) { // ここから起動
        new Sample02().setVisible(true);
    }
    Color col = new Color(255, 0, 0);
    int x = 100, y = 100;
    public Sample02() { // 初期設定
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); setSize(400, 400);
        addMouseListener(new MouseAdapter() {
            public void mousePressed(MouseEvent evt) { // ボタン押された
```

```

        x = evt.getX(); y = evt.getY();
        col = new Color(0, 255, 0); repaint();
    }
    public void mouseReleased(MouseEvent evt) { // ボタン離された
        col = new Color(255, 0, 0); repaint();
    }
});
addMouseListener(new MouseMotionAdapter() {
    public void mouseDragged(MouseEvent evt) { // ドラグされた
        x = evt.getX(); y = evt.getY(); repaint();
    }
});
}
public void paint(Graphics g) { // 画面描画
    g.setColor(getBackground()); // 背景色に設定し
    g.fillRect(0, 0, getWidth(), getHeight()); // 全体をクリア
    g.setColor(col); g.fillRect(x, y, 60, 40);
}
}

```

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。動いたら次のように直してみよ。

- a. クリックしたところの「少し右」に図形が移動するように。
- b. ドラグ時の X/Y 座標をもっと別な計算式で与える (たとえば `x` と `y` を入れ換えるとか)。
- c. 位置に応じて色が変わるようにする (ヒント: `new Color` で指定する数値を `x` や `y` に基づいて計算する)。
- 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。
- 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。
- f. その他自分の好きな動き方の工夫。

## 6 次回までの追加課題

次回でいよいよこの授業も最終回になります。なので「次回まで」の追加課題として、模擬授業の HTML の後に続けて次のテーマについて 200 文字程度で自分の考えを書いてください。

- テーマ名: 教科「情報」の授業はどのようにするのがよいか
- 内容: たとえば「コンピュータの操作をじっくり教えるのがよい」「操作よりはプログラミングやコンピュータの仕組みなどの基本原理に通じるものを教えるのがよい」「コンピュータよりも情報社会やその影響などに重点を置くのがよい」など、「情報」の授業のあり方について自分の立場を決め、それをまず冒頭で簡潔に明記する (「私は、～であると考えている」などと)。続いて、なぜそう思うのかについて理由や根拠や考えの筋道などを説明する。