

JAVA を使った物理教育プログラミング

高嶋隆一

京都教育大学 理学科 takasima@kyokyo-u.ac.jp

キーワード：JAVA、オブジェクト指向言語、物理教育、GUI

(受付け：2000 年 5 月 1 日)

I. はじめに

理科教育の中の物理分野では、現象を数学的に取り扱うことがよく行われる。計算機は数学的な取り扱いが、数値計算によってより容易になるという利点がある。従って物理教育においては計算機の利用がよくおこなわれる。最初に計算機を使った物理教育によく利用されたのが BASIC 言語であった。

BASIC は文法が単純でわかりやすいために、簡単なプログラムを書いて、運動方程式を解き、結果を図示するというように、物理教育に役に立つ道具として使われて来た。コードも短く、サンプルプログラムをコピーして実行するには好適であった。ところが最近の計算機のプログラミング環境では、作成されたプログラムがグラフィックユーザーインターフェース(GUI)を通じて利用者とパラメーターをやりとりするようになっている。その結果プログラミングは複雑になってしまった。

さらに、オブジェクト指向といった新しいプログラミングスタイルが使われるようになったことも、今までの言語でプログラミングを行って来たものにとっては、手軽にプログラム作成に取り組みにくい結果となっている。

ここでは、いままで BASIC で書かれていたプログラムをインターネット環境で利用しやすい JAVA というプログラム言語に書き換える手法を紹介する。

JAVA はブラウザーに実行機能が付属しているので、作られた実行コードは小さく、ネットワークから転送して来て自分の計算機で実行するアプレットと呼ばれる仕組みが作られている。現在は開発環境が無料で提供されているのも良い点である。

II. JAVA とはどのようなものなのか

プログラムを書くために必要なのは Java Development Kit (JDK)とよばれるものである。これは Windows, Linux など用に無料で配布されている。Windows 用のものは sunsite 等からとって来ることができる。Linux 用のものもバイナリ

一のものが配布されている。これをとって来てコマンドパス(場合によっては環境変数 `JAVA_HOME` も)を設定するだけでよい。Linux 版では漢字が表示されないこともある。Solaris 版では漢字の表示は、開発元であるので問題は無い。今回使用したのは JDK1.1 であるが、現在ものは JDK1.2 である。次のセクションでソースコードを解説する `double_in3.java` の実行を例に説明する。バイトコードと呼ばれる `double_in3.class` の生成をコンパイルと呼ぶが以下のコマンドを実行する。

```
javac double_in3.java
```

場合によっては `double_in3.class` の他にいくつかのバイトコードができることもあるが気にしなくて良い。更にこれを実行するためには以下のようなファイルを作っておく必要がある。

```
1 <html>
2 <head>
3 <title>double_in3</title>
4 </head>
5 <body>
6 <hr>
7 <applet CODE="double_in3.class" WIDTH=400 HEIGHT=440>
8 </applet>
9 </body>
10 </html>
```

このファイル名を `double_in3.html` とすると、このファイルをブラウザから URL として呼び出せばプログラムが実行される。JDK には `appletviewer` というアプリケーションが付いて来るので以下のようにしてプログラムを実行しても良い。

```
appletviewer double_in3.html
```

III. GUI からのパラメータ入力

BASIC なら INPUT 文によって、プログラムの実行条件を入れることができたが JAVA でそれをしようとすると少し複雑な手続きを経ることになる。以下のプログラム `double_in3.java` はアプレットに与えられた領域を青色で塗りつぶすものであるが二つの入力欄を持っており、最初の欄に領域の割合を、次の欄に青色の明度を入力するようになっている。ボタンを押すと欄の値を使って描画が実行される。このプログラムは、例題のたくさんのっている教科書の例題に変更を加えたものである。(河西,1998) クラス `Applet` を継承して作られたクラス `double_in3` には関数 `init()`、`paint()` が最初から組み込まれているが、これを

上書き(オーバーライド)して自分の必要な動作をさせることができる。グラフィックユーザーインターフェースを使うプログラミングではこのような関数 `init()` の一部、`paint()` を書き換える操作が主になる。

```
1  import java.applet.Applet;
2  import java.awt.*;
3  import java.awt.event.*;
4
5  public class double_in3 extends Applet{
6      Graphics g;
7      static int DOUBLENUM = 2; //GUI の変数の数をいれる。
8      Gui_double []gui= new Gui_double[DOUBLENUM];
9
10     public void init(){
11         Button bt;
12         g=getGraphics();
13         //ここにラベルと初期値、欄の大きさをいれる。変数の数だけ必要
14         gui[0]=new Gui_double("矩形の大きさ(0-1)","0.1",5);
15         gui[1]=new Gui_double("青の濃さ(0-255)","255",5);
16         //
17         add(bt=new Button("Draw"));
18         bt.addActionListener(new ActionListener(){
19             public void actionPerformed(ActionEvent e){
20                 int i;
21                 for (i=0; i < DOUBLENUM ;i++){
22                     gui[i].dval=new Double(gui[i].tf.
23                                             getText()).doubleValue();
24                 }
25                 repaint();
26             }
27         });
28         //ここに計算と描画のプログラムを記述
29         public void paint(Graphics g){
30             g.setColor(new Color(0,0,(int)gui[1].dval));
31             g.fillRect(0,40,(int)(gui[0].dval*400.),
32                       40+(int)(gui[0].dval*400.f));
```

```

33     }
34     class Gui_double{
35         Label la;
36         TextField tf;
37         double dval;
38         public Gui_double(String st1,String st2,int d_field){
39             add(la=new Label(st1));
40             add(tf=new TextField(st2,d_field));
41         }
42     }
43 }

```

このプログラムは、BASIC で書かれたプログラムを書き換える時のテンプレートとして使われるように書かれている。コメントで示した部分だけを書き換えれば、大抵のプログラムは変更できるはずである。

変数のいくつかは GUI から与えられるためにプログラムが多少複雑になっている。7 行目のように GUI から与える変数の数を最初にセットする必要がある。これはあとで、描画ボタンを押したときに自動的に文字列を double 型に変換するためのループの回数を決めるために必要になる(21-23 行目)。このため、8 行目のように GUI から与える変数を配列で定義しておく。これはオブジェクトを参照するための変数を作成するもので、実体は 14、15 行目で作られる。

このプログラムにはオブジェクト指向プログラミングのいくつかの要素があらわれている。データ型には、基本型(boolean,int,float,double,char など)とオブジェクトがあり、基本型は 20 行目のように宣言するだけで値を保持する場所がとられる。これに対して、オブジェクト(ここでは Graphics、TextField、Label、Gui_double など)は以下のように new 演算子を使って値を保持する場所を確保する。

```
gui[0]=new Gui_double("矩形の大きさ(0-1)","0.1",5);
```

ここで Gui_double("x","y",n)として、オブジェクト Gui_double のコンストラクターという関数が呼ばれ、ラベルと初期値、欄の大きさをセットしている。ラベルに日本語が使われているが Solaris で EUC を使うと正しく動作することを確認した。

17,39,40 行目の add()はオブジェクトに対するメソッドと呼ばれる関数の実行命令となっている。ここでは オブジェクト Applet を拡張(extends)したオブジェクト double_in3 に作用する関数として呼ばれている。この呼出は this.add()を省略した形である。これによってオブジェクト double_in3 にボタン、ラベル、テキスト入力欄が付加される。実際に動かすためには以下の Uniform

Resource Locator(URL)を参照する。

<http://natsci.kyokyo-u.ac.jp/~takasima/myjava/din3/din3.html>

IV. BASIC プログラムの書き換えの例

次に、BASIC プログラムの書き換えの例を実際に示す。これは画面に入力欄に示した数の振動を二つ表示し、波の重ね合わせが起きたとき、うなりが生ずることを示したものである。(平田,1988)プログラムの動作は以下の URL で試することができる。使用されているコードも示されている。

<http://natsci.kyokyo-u.ac.jp/~takasima/myjava/vw4/vw4.html>

実際の動作の様子を図1に示す。書き換える時の注意すべき点をいくつかのべる。BASICでは関数が良く使われているが、Java流の関数に置き換える必要がある。型が違う変数への代入を行うときはキャストという操作を行う。ラベルつき実数入力欄の数値の取り扱いは `gui[0].dval` のような形で行うことが少し面倒な点である。BASICの `pset` に対応するものが無いので、同じ点を結ぶことによって `pset` のかわりにしている。

教育的には時間的に変化する現象を示すことも重要である。以下の URL に波動方程式を数値計算によって解いている BASIC プログラムの例を JAVA によって書き換えたものを示す。(平田,1988)

<http://natsci.kyokyo-u.ac.jp/~takasima/myjava/wave/wave.html>

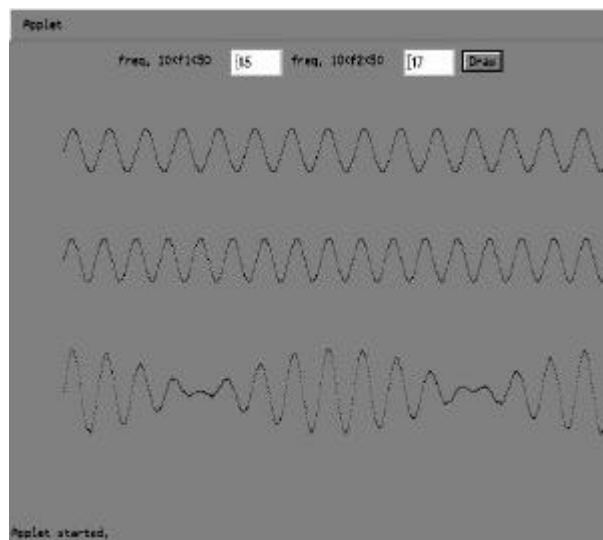


図1 : BASIC プログラムを書き換えて実行した例

このようなアニメーション技法と、GUI 変数入力を組み合わせて、教育ソフトを作ることが簡単にできるようになって来た。

V. おわりに

現在は、ソフトウェアにかけるお金を少なくできるようになって来ている。とくに教育のためにはフリーソフトウェアの導入が必要である。使いやすいソフトウェアがたくさん出て来ているのは喜ばしい。

すでに各所で教育への活用が試みられているが、一つだけ JAVA 物理教材メーリングリストと呼ばれる以下のサイトを示しておく。

<http://www.nep.chubu.ac.jp/~nepjava/>

今回のテンプレート(double_in3)に関しては、利用者が必要無い部分まで見えるようになっている。JAVA 言語の特徴としては、作成したプログラムをクラスライブラリとして利用することによって、このような不必要な部分にふれないようにすることもできる。今後も今まで他の言語で書かれていた、教育的なプログラムを Java に書き換えて行く作業を続ける予定である。

文献

河西朝雄、1998、Internet Language Java 入門、技術評論社、154-157

平田邦男、1988、新/BASIC による物理、共立出版社、28-32、44-49