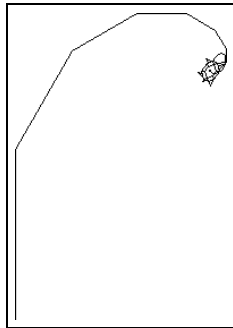


計算機プログラミング I (2)  
対話的なプログラム (補足)

2004 年 10 月 18 日 増原 英彦

## 1 プログラム実行中の変数の値を調べる

例えば下のような渦巻き模様を描くプログラムを作りたかったとしよう。



そしてあなたは、

最初一定の長さ前進して、30 度回転したら、  
前回の  $\frac{2}{3}$  倍の長さ前進して、また 30 度回転して、  
また前回の  $\frac{2}{3}$  倍の長さ前進して、また 30 度回転して、  
.....を何回か繰り返す

ことで描けることに気づき、これは、

長さ `length` だけ前進して、  
30 度回転して、  
`length` を  $\frac{2}{3}$  倍する

命令を何回かコピーすることでできるだろうと思い次のようなプログラムを書いたとする。

```
1 public class DebugVariable {
2     public static void main(String[] args){
3         int length = 150, angle = 30, factor = 2/3;
4
5         TurtleFrame f = new TurtleFrame();
6         Turtle m = new Turtle(100,350,0);
7         f.add(m);
8
9         m.fd(length);           //length だけ前進
10        m.rt(angle);           //angle だけ回転
11        length = length*factor; //length を factor 倍する
12
13        m.fd(length);           //length だけ前進
14        m.rt(angle);           //angle だけ回転
15        length = length*factor; //length を factor 倍する
16
17        (上の 4 行のコピーが続く)
18    }
19 }
```



```

1 public class DebugVariable {
2     public static void main(String[] args){
3         int length = 150, angle = 30, factor = 2/3;
4
5         TurtleFrame f = new TurtleFrame();
6         Turtle m = new Turtle(100,350,0);
7         f.add(m);
8
9         System.out.println("value of length = " + length); //length を表示
10        System.out.println("value of factor = " + factor); //factor を表示
11        m.fd(length); //length だけ前進
12        m.rt(angle); //angle だけ回転
13        length = length*factor; //length を factor 倍する
14
15        System.out.println("value of length = " + length); //length を表示
16        System.out.println("value of factor = " + factor); //factor を表示
17        m.fd(length); //length だけ前進
18        m.rt(angle); //angle だけ回転
19        length = length*factor; //length を factor 倍する
20
21        (上の 4 行のコピーが続く)
22    }
23 }

```

図 1: length と factor の値を表示するプログラム

変数 length と変数 factor の値が交互に表示されているので見づらいが、 $2/3$  という値を持っているはずの変数 factor が最初から 0 であることが分かった。これでは 13 行目の命令を実行すると変数 length が 0 になってしまう訳である。

変数 factor が 0 になる理由とその対策は、この週の範囲を越える<sup>1</sup>。しかしこのようにして変数の値を調べること、プログラムが期待通りに動かない原因に近付けることが分かる。

さらに次週以降で、繰り返し・条件分岐・メソッド定義などを使うようになると、「プログラムがどの命令をどの順番で何回実行するか」も表示して調べたいくなる。そのような場合にも、この方法は有効である。

## 2 変数の値をキーボードから入力する

プログラムを実行したときに、キーボードから「辺の長さ」を入力するとその大きさの図形を描くようなことができると、プログラムを変更したりコンパイルすることなく色々な値について試みることができるので便利である。

配布プログラムには、キーボードから数値（や文字列）を入力する機能が備っているので、その使い方を見よう。図 2 は、そのようなプログラムの例である。このプログラムでは、8 行目に

```
int length = TurtleFrame.askInt("length of edges");
```

という命令がある。この右辺の TurtleFrame.askInt("length of edges"); は int 型の整数をキーボードから入力するウインドウを表示せよ、という意味である。同様に TurtleFrame.askDouble("message") や TurtleFrame.askString("message") という式は、それぞれ倍精度実数型、文字列型の値をキーボードから入力してもらう式である。

練習 2-1: (キーボードからの入力 1) 上を参考にして、キーボードから整数 x を入力して、1 辺の長さがそれぞれ 10, 10x, 20x の 3 つの三角形を描くプログラムを作ってみよ。

<sup>1</sup>int 型は整数型なので、割り算の結果は全て切り捨てになる。そのため、ある数を  $2/3$  倍するときは、その数を 2 倍してから 3 で割らなければならない。

```

1  /** キーボードから入力をするプログラムの例 */
2  public class AskExample {
3      public static void main(String[] args){
4          TurtleFrame f = new TurtleFrame();
5          Turtle m = new Turtle();
6          f.add(m);
7
8          int length = TurtleFrame.askInt("length of edges"); //辺の長さをキーボードから入力してもらう
9
10         m.fd(length); //入力ぶんだけ前進
11         m.rt(120); //120 度回転
12
13         m.fd(length); //入力ぶんだけ前進
14         m.rt(120); //120 度回転
15
16         m.fd(length); //入力ぶんだけ前進
17         m.rt(120); //120 度回転
18     }
19 }

```

図 2: キーボードから入力してもらうプログラム

練習 2-2: (キーボードからの入力 2) 上を参考にして、キーボードから整数  $x$  を入力して、半径  $x$  の円に内接する正三角形を描くプログラムを作れるか? (ヒント: 3 ページの注)

練習 2-3: (キーボードからの入力 3) 上を参考にして、キーボードから整数  $x, y, z$  を入力して、1 ページ目のような渦巻き模様を描くプログラムを作ってみよ。ただし、辺の長さは毎回  $x/y$  倍に減り、毎回  $z$  度だけ回転するものとする。

練習 2-4: (キーボードからの入力 4) 上を参考にして、キーボードから整数  $n$  を受け取って 1 辺の長さが 100 の  $n$  角形を描くプログラムは (今週まで習ったことだけで) 作れるか? 1 辺の長さが 100 の正三角形を  $n$  個、中心のまわりに描くプログラムはどうか? 考えてみよ。