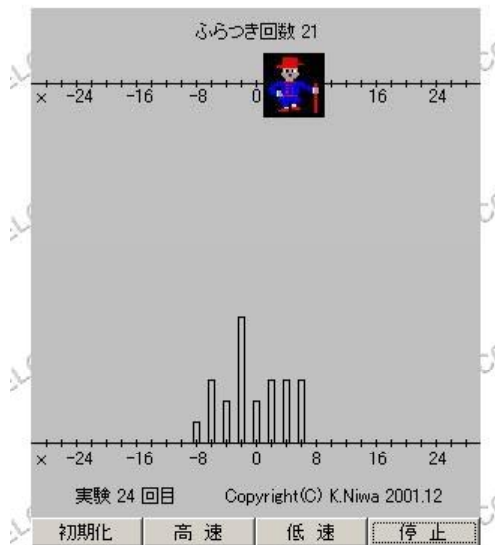
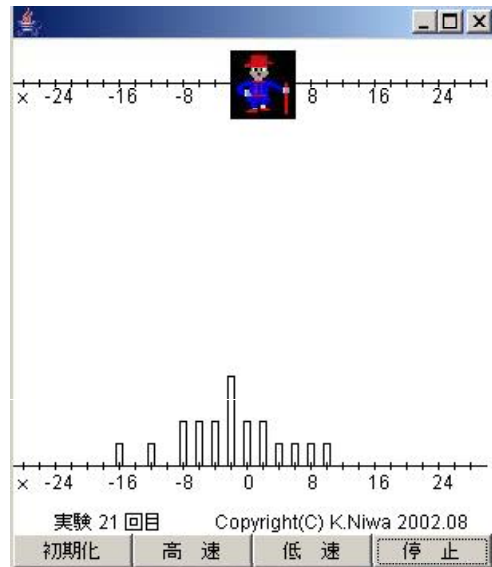


【ランダムウォーク】



[J a v a アプレット]



[J a v a アプレケーション]

1. はじめに

酔っばりは前後左右見境なくふらつきます。

酔っばりは目的地にたどり着こうと歩き回っているうちに何度も同じところに戻って来てしまったりするものです。

今、酔っばりが数直線上の原点にいます。

原点を出発して30回ふらつくとき、30回目に酔っばりがいる位置は、出発点である原点からどれくらい離れてしまっているのでしょうか。

酔っばりは左右の2方向にふらつき、1回のふらつきで、左にふらつくのか右にふらつくのかは半々で、移動距離は1とします。

1回の実験ごとに、原点を出発して、ふらつき30回目の位置をカウントし、棒グラフで表示します。

シミュレーションソフト「ランダムウォーク」で酔っばりの足取りを観察してみてください。

(ただし、実験61回目に止まります。)

2. J a v a アプレット

(1) J a v a プログラムリスト

```
////////////////////////////////////  
//  
//                               「ランダムウォーク」  
//                               Copyright (C) K.Niwa 2001.12.20  
//                               ( J a v a アプレット )  
//  
//  
////////////////////////////////////
```

```
//クラスの読み込み  
import java.applet.Applet;  
import java.awt.*;  
import java.awt.event.*;  
import java.lang.Math; //Mathクラスを読み込む
```

```

/***** public class Randomwalk2 extends Applet implements Runnable*****/
public class Randomwalk2 extends Applet implements Runnable{ //スレッドを使えるようにする

//変数とオブジェクトの型宣言
    Thread myTh; //スレッド型で宣言する
    Button[] myBtn; //ボタン型で宣言する
    Panel myPanel; //パネル型で宣言する
    int x=150,y=30,oldx,oldy; //酔っぱらいの x 座標と y 座標
    int px; //酔っぱらいのふらつき 3 0 回目の x 座標
    int xx; //座標軸の描写に使用
    int k=0; //ふらつき回数に使用
    int flag=0;
    int ct=1; //実験回数
    int[] cy=new int[61]; //ふらつき 3 0 回目の位置のカウント
    long t; //時間稼ぎに使用
    double r; //乱数
    Image myImg; //イメージ型で宣言する
    MediaTracker myMt; //メディアトラッカー型で宣言する
    int count; //ループカウンター
    int N; //実験回数
    int Speed=100; //速度

/***** public void init()メソッド *****/
public void init () {
    myTh=null; //スレッドの初期化
    setBackground (Color.lightGray); //背景色をライトグレーにする
    myImg=getImage (getDocumentBase (), "hito1.jpg"); //酔っぱらいの画像の読み込み

    myMt=new MediaTracker (this); //メディアトラッカーの実体化
    myMt.addImage (myImg,0); //画像をメディアトラッカーに貼り付ける

    try {
        myMt.waitForID (0); //メディアトラッカーに画像を読み込む
    }
    catch (InterruptedException e) {
    }

    myBtn=new Button [4]; //ボタンの実体化
    myBtn [0]=new Button ("初期化");
    myBtn [1]=new Button ("高 速");
    myBtn [2]=new Button ("低 速");
    myBtn [3]=new Button ("停 止");

    myPanel=new Panel (); //パネルの実体化

    myPanel.setLayout (new GridLayout (1,4)); //パネルをグリッドレイアウトにする

    for (count=0;count<=3;count++) {
        myPanel.add (myBtn [count]); //パネルにボタンを貼り付ける
    }
    setLayout (new BorderLayout ()); //全体をボーダーレイアウトにする
    add ("South",myPanel); //パネルを南に貼り付ける

    myBtn [0].addActionListener (new ActionListener () { //初期化ボタン
        public void actionPerformed (ActionEvent e) {
            flag=0; //識別子
            x=150;y=30; //酔っぱらいの位置の初期化
            k=0; //ふらつき回数の初期化
            ct=1; //実験回数の初期化
            repaint ();
        }
    });
}
}

```

```

myBtn[1].addActionListener(new ActionListener() { //高速ボタン
    public void actionPerformed(ActionEvent e) {
        flag=1; //識別子
        Speed=50; //速度
        //x=150;y=30; //酔っぱらいの位置の初期化
        //k=0; //ふらつき回数の初期化
        //ct=1; //実験回数の初期化
        repaint(); //paint()メソッドを呼び出す
    }
});

myBtn[2].addActionListener(new ActionListener() { //低速ボタン
    public void actionPerformed(ActionEvent e) {
        flag=2; //識別子
        Speed=100; //速度
        //x=150;y=30; //酔っぱらいの位置の初期化
        //k=0; //ふらつき回数の初期化
        //ct=1; //実験回数の初期化
        repaint(); //paint()メソッドを呼び出す
    }
});

myBtn[3].addActionListener(new ActionListener() { //停止ボタン
    public void actionPerformed(ActionEvent e) {
        flag=3; //識別子
        repaint(); //paint()メソッドを呼び出す
    }
});
} //public void init()

/***** public void start()メソッド *****/
public void start() {
    if (myTh==null) {
        myTh=new Thread(this); //スレッドの実体化
        myTh.start(); //スレッドを開始する
    }
}

/***** public void run()メソッド <イベントが発生しなくても処理する部分を記述する> *****/
public void run() {
    while (true) {
        try {
            myTh.sleep(Speed); //休む
        }
        catch (InterruptedException e) {}
        if (flag==1 || flag==2) {
            repaint(); //paint()メソッドを呼び出す
        }
    }
}

/***** public void paint(Graphics g)メソッド *****/
public void paint(Graphics g) {
    if (myMt.checkID(0)) {
        //初期状態または初期化ボタンを押したときのイベント処理
        if (flag==0) {
            for (count=0;count<=60;count++) {
                //ふらつき30回目の位置のカウントの初期化
                cy[count]=0;
            }
        }
        //座標軸描写
        g.clearRect(0,0,300,360); //全体のクリア
    }
}

```

```

g.drawString("ふらつき回数 "+k,110,20);
g.drawLine(150,50-2,150,50+2); //y 軸 (上側)
g.drawLine(0,50,300,50); //x 軸 (上側)

for (xx=10;xx<=300;xx=xx+10) {
    g.drawLine(xx,48,xx,52); //x 軸メモリ線 (上側)
}

g.drawString("0",148,63); //x 軸メモリ値 (上側)
g.drawString("8",188,63);
g.drawString("16",226,63);
g.drawString("24",266,63);
g.drawString("-8",106,63);
g.drawString("-16",64,63);
g.drawString("-24",24,63);
g.drawString(" x ",4,63);

g.drawLine(0,290,300,290); //x 軸 (下側)

for (xx=10;xx<300;xx=xx+10) {
    g.drawLine(xx,288,xx,292); //x 軸メモリ線 (下側)
}

g.drawString("0",148,305); //x 軸メモリ値 (下側)
g.drawString("8",188,305);
g.drawString("16",226,305);
g.drawString("24",266,305);
g.drawString("-8",106,305);
g.drawString("-16",64,305);
g.drawString("-24",24,305);
g.drawString(" x ",4,305);

g.drawImage(myImg,x-20,y,this); //酔っぱらいの表示
g.drawString("実験 "+ct+" 回目",30,330); //実験回数の表示
g.drawString("Copyright(C) K.Niwa 2001.12",130,330); //作者表示

} //if(flag==0)

//高速ボタンまたは低速ボタンを押したときのイベント処理
else if (flag==1 || flag==2) {
    //---- 棒グラフの描写の開始
    g.drawRect(148,290-cy[0]*14,4,cy[0]*14);
    g.drawRect(153,290-cy[1]*14,4,cy[1]*14);
    g.drawRect(158,290-cy[2]*14,4,cy[2]*14);
    g.drawRect(163,290-cy[3]*14,4,cy[3]*14);
    g.drawRect(168,290-cy[4]*14,4,cy[4]*14);
    g.drawRect(173,290-cy[5]*14,4,cy[5]*14);
    g.drawRect(178,290-cy[6]*14,4,cy[6]*14);
    g.drawRect(183,290-cy[7]*14,4,cy[7]*14);
    g.drawRect(188,290-cy[8]*14,4,cy[8]*14);
    g.drawRect(193,290-cy[9]*14,4,cy[9]*14);
    g.drawRect(198,290-cy[10]*14,4,cy[10]*14);
    g.drawRect(203,290-cy[11]*14,4,cy[11]*14);
    g.drawRect(208,290-cy[12]*14,4,cy[12]*14);
    g.drawRect(213,290-cy[13]*14,4,cy[13]*14);
    g.drawRect(218,290-cy[14]*14,4,cy[14]*14);
    g.drawRect(223,290-cy[15]*14,4,cy[15]*14);
    g.drawRect(228,290-cy[16]*14,4,cy[16]*14);
    g.drawRect(233,290-cy[17]*14,4,cy[17]*14);
    g.drawRect(238,290-cy[18]*14,4,cy[18]*14);
    g.drawRect(243,290-cy[19]*14,4,cy[19]*14);
    g.drawRect(248,290-cy[20]*14,4,cy[20]*14);
    g.drawRect(253,290-cy[21]*14,4,cy[21]*14);
    g.drawRect(258,290-cy[22]*14,4,cy[22]*14);
}

```

```

g.drawRect (263,290-cy [23] *14,4,cy [23] *14);
g.drawRect (268,290-cy [24] *14,4,cy [24] *14);
g.drawRect (273,290-cy [25] *14,4,cy [25] *14);
g.drawRect (278,290-cy [26] *14,4,cy [26] *14);
g.drawRect (283,290-cy [27] *14,4,cy [27] *14);
g.drawRect (288,290-cy [28] *14,4,cy [28] *14);
g.drawRect (293,290-cy [29] *14,4,cy [29] *14);
g.drawRect (143,290-cy [31] *14,4,cy [31] *14);
g.drawRect (138,290-cy [32] *14,4,cy [32] *14);
g.drawRect (133,290-cy [33] *14,4,cy [33] *14);
g.drawRect (128,290-cy [34] *14,4,cy [34] *14);
g.drawRect (123,290-cy [35] *14,4,cy [35] *14);
g.drawRect (118,290-cy [36] *14,4,cy [36] *14);
g.drawRect (113,290-cy [37] *14,4,cy [37] *14);
g.drawRect (108,290-cy [38] *14,4,cy [38] *14);
g.drawRect (103,290-cy [39] *14,4,cy [39] *14);
g.drawRect (98,290-cy [40] *14,4,cy [40] *14);
g.drawRect (93,290-cy [41] *14,4,cy [41] *14);
g.drawRect (88,290-cy [42] *14,4,cy [42] *14);
g.drawRect (83,290-cy [43] *14,4,cy [43] *14);
g.drawRect (78,290-cy [44] *14,4,cy [44] *14);
g.drawRect (73,290-cy [45] *14,4,cy [45] *14);
g.drawRect (68,290-cy [46] *14,4,cy [46] *14);
g.drawRect (63,290-cy [47] *14,4,cy [47] *14);
g.drawRect (58,290-cy [48] *14,4,cy [48] *14);
g.drawRect (53,290-cy [49] *14,4,cy [49] *14);
g.drawRect (48,290-cy [50] *14,4,cy [50] *14);
g.drawRect (43,290-cy [51] *14,4,cy [51] *14);
g.drawRect (38,290-cy [52] *14,4,cy [52] *14);
g.drawRect (33,290-cy [53] *14,4,cy [53] *14);
g.drawRect (28,290-cy [54] *14,4,cy [54] *14);
g.drawRect (23,290-cy [55] *14,4,cy [55] *14);
g.drawRect (18,290-cy [56] *14,4,cy [56] *14);
g.drawRect (13,290-cy [57] *14,4,cy [57] *14);
g.drawRect (8,290-cy [58] *14,4,cy [58] *14);
g.drawRect (3,290-cy [59] *14,4,cy [59] *14);

```

// 1 回分の実験 (ふらつき回数 30 回)

```

if (k<30) {
    k++; //ふらつき回数

//実験回数の表示
g.drawString ("実験 "+ct+" 回目",30,330);
g.clearRect (110,10,100,20);
g.drawString ("ふらつき回数 "+k,110,20);

oldx=x; //酔っぱらいの x 座標
oldy=y; //酔っぱらいの y 座標

r=Math.random ();
if (r<0.5) { //左へふらつく
    x=x-5;
}
else if (r>=0.5) { //右へふらつく
    x=x+5;
}

g.clearRect (oldx-20,oldy,41,43); //酔っぱらいの消去
g.drawImage (myImg,x-20,y,this); //酔っぱらいの描写

//座標軸描写
g.drawLine (150,50-2,150,50+2); // y 軸 (上側)
g.drawLine (0,50,300,50); // x 軸 (上側)

```

```

for (xx=10;xx<=300;xx=xx+10) {
    g.drawLine(xx,48,xx,52); // x 軸メモリ線 (上側)
}

g.drawString("0",148,63); // x 軸メモリ値 (上側)
g.drawString("8",188,63);
g.drawString("16",226,63);
g.drawString("24",266,63);
g.drawString("-8",106,63);
g.drawString("-16",64,63);
g.drawString("-24",24,63);
g.drawString(" x ",4,63);

g.drawLine(0,290,300,290); // x 軸 (下側)

for (xx=10;xx<300;xx=xx+10) {
    // x 軸メモリ線 (下側)
    g.drawLine(xx,288,xx,292);
}

// x 軸メモリ値 (下側)
g.drawString("0",148,305);
g.drawString("8",188,305);
g.drawString("16",226,305);
g.drawString("24",266,305);
g.drawString("-8",106,305);
g.drawString("-16",64,305);
g.drawString("-24",24,305);
g.drawString(" x ",4,305);

// x 軸メモリ値 (下側)
g.drawString("Copyright (C) K.Niwa 2001.12",130,330);
} //if (k<=30)

//----- ふらつき 30 回目の位置の判定
if (k==30) {
    switch (x) { //酔っぱらいの x 座標 x
        case 150:
            cy[0]=cy[0]+1;
            break;
        case 155:
            cy[1]=cy[1]+1;
            break;
        case 160:
            cy[2]=cy[2]+1;
            break;
        case 165:
            cy[3]=cy[3]+1;
            break;
        case 170:
            cy[4]=cy[4]+1;
            break;
        case 175:
            cy[5]=cy[5]+1;
            break;
        case 180:
            cy[6]=cy[6]+1;
            break;
        case 185:
            cy[7]=cy[7]+1;
            break;
        case 190:
            cy[8]=cy[8]+1;
            break;
    }
}

```

```
case 195:
    cy[9]=cy[9]+1;
    break;
case 200:
    cy[10]=cy[10]+1;
    break;
case 205:
    cy[11]=cy[11]+1;
    break;
case 210:
    cy[12]=cy[12]+1;
    break;
case 215:
    cy[13]=cy[13]+1;
    break;
case 220:
    cy[14]=cy[14]+1;
    break;
case 225:
    cy[15]=cy[15]+1;
    break;
case 230:
    cy[16]=cy[16]+1;
    break;
case 235:
    cy[17]=cy[17]+1;
    break;
case 240:
    cy[18]=cy[18]+1;
    break;
case 245:
    cy[19]=cy[19]+1;
    break;
case 250:
    cy[20]=cy[20]+1;
    break;
case 255:
    cy[21]=cy[21]+1;
    break;
case 260:
    cy[22]=cy[22]+1;
    break;
case 265:
    cy[23]=cy[23]+1;
    break;
case 270:
    cy[24]=cy[24]+1;
    break;
case 275:
    cy[25]=cy[25]+1;
    break;
case 280:
    cy[26]=cy[26]+1;
    break;
case 285:
    cy[27]=cy[27]+1;
    break;
case 290:
    cy[28]=cy[28]+1;
    break;
case 295:
    cy[29]=cy[29]+1;
    break;
```

```
case 145:
    cy[31]=cy[31]+1;
    break;
case 140:
    cy[32]=cy[32]+1;
    break;
case 135:
    cy[33]=cy[33]+1;
    break;
case 130:
    cy[34]=cy[34]+1;
    break;
case 125:
    cy[35]=cy[35]+1;
    break;
case 120:
    cy[36]=cy[36]+1;
    break;
case 115:
    cy[37]=cy[37]+1;
    break;
case 110:
    cy[38]=cy[38]+1;
    break;
case 105:
    cy[39]=cy[39]+1;
    break;
case 100:
    cy[40]=cy[40]+1;
    break;
case 95:
    cy[41]=cy[41]+1;
    break;
case 90:
    cy[42]=cy[42]+1;
    break;
case 85:
    cy[43]=cy[43]+1;
    break;
case 80:
    cy[44]=cy[44]+1;
    break;
case 75:
    cy[45]=cy[45]+1;
    break;
case 70:
    cy[46]=cy[46]+1;
    break;
case 65:
    cy[47]=cy[47]+1;
    break;
case 60:
    cy[48]=cy[48]+1;
    break;
case 55:
    cy[49]=cy[49]+1;
    break;
case 50:
    cy[50]=cy[50]+1;
    break;
case 45:
    cy[51]=cy[51]+1;
    break;
```



```

        case 40:
            cy[52]=cy[52]+1;
            break;
        case 35:
            cy[53]=cy[53]+1;
            break;
        case 30:
            cy[54]=cy[54]+1;
            break;
        case 25:
            cy[55]=cy[55]+1;
            break;
        case 20:
            cy[56]=cy[56]+1;
            break;
        case 15:
            cy[57]=cy[57]+1;
            break;
        case 10:
            cy[58]=cy[58]+1;
            break;
        case 5:
            cy[59]=cy[59]+1;
            break;
    }//switch (x)

//---- 棒グラフの描写の開始
g.drawRect(148,290-cy[0]*14,4,cy[0]*14);
g.drawRect(153,290-cy[1]*14,4,cy[1]*14);
g.drawRect(158,290-cy[2]*14,4,cy[2]*14);
g.drawRect(163,290-cy[3]*14,4,cy[3]*14);
g.drawRect(168,290-cy[4]*14,4,cy[4]*14);
g.drawRect(173,290-cy[5]*14,4,cy[5]*14);
g.drawRect(178,290-cy[6]*14,4,cy[6]*14);
g.drawRect(183,290-cy[7]*14,4,cy[7]*14);
g.drawRect(188,290-cy[8]*14,4,cy[8]*14);
g.drawRect(193,290-cy[9]*14,4,cy[9]*14);
g.drawRect(198,290-cy[10]*14,4,cy[10]*14);
g.drawRect(203,290-cy[11]*14,4,cy[11]*14);
g.drawRect(208,290-cy[12]*14,4,cy[12]*14);
g.drawRect(213,290-cy[13]*14,4,cy[13]*14);
g.drawRect(218,290-cy[14]*14,4,cy[14]*14);
g.drawRect(223,290-cy[15]*14,4,cy[15]*14);
g.drawRect(228,290-cy[16]*14,4,cy[16]*14);
g.drawRect(233,290-cy[17]*14,4,cy[17]*14);
g.drawRect(238,290-cy[18]*14,4,cy[18]*14);
g.drawRect(243,290-cy[19]*14,4,cy[19]*14);
g.drawRect(248,290-cy[20]*14,4,cy[20]*14);
g.drawRect(253,290-cy[21]*14,4,cy[21]*14);
g.drawRect(258,290-cy[22]*14,4,cy[22]*14);
g.drawRect(263,290-cy[23]*14,4,cy[23]*14);
g.drawRect(268,290-cy[24]*14,4,cy[24]*14);
g.drawRect(273,290-cy[25]*14,4,cy[25]*14);
g.drawRect(278,290-cy[26]*14,4,cy[26]*14);
g.drawRect(283,290-cy[27]*14,4,cy[27]*14);
g.drawRect(288,290-cy[28]*14,4,cy[28]*14);
g.drawRect(293,290-cy[29]*14,4,cy[29]*14);
g.drawRect(143,290-cy[31]*14,4,cy[31]*14);
g.drawRect(138,290-cy[32]*14,4,cy[32]*14);
g.drawRect(133,290-cy[33]*14,4,cy[33]*14);
g.drawRect(128,290-cy[34]*14,4,cy[34]*14);
g.drawRect(123,290-cy[35]*14,4,cy[35]*14);
g.drawRect(118,290-cy[36]*14,4,cy[36]*14);
g.drawRect(113,290-cy[37]*14,4,cy[37]*14);

```

```

g.drawRect(108,290-cy[38]*14,4,cy[38]*14);
g.drawRect(103,290-cy[39]*14,4,cy[39]*14);
g.drawRect(98,290-cy[40]*14,4,cy[40]*14);
g.drawRect(93,290-cy[41]*14,4,cy[41]*14);
g.drawRect(88,290-cy[42]*14,4,cy[42]*14);
g.drawRect(83,290-cy[43]*14,4,cy[43]*14);
g.drawRect(78,290-cy[44]*14,4,cy[44]*14);
g.drawRect(73,290-cy[45]*14,4,cy[45]*14);
g.drawRect(68,290-cy[46]*14,4,cy[46]*14);
g.drawRect(63,290-cy[47]*14,4,cy[47]*14);
g.drawRect(58,290-cy[48]*14,4,cy[48]*14);
g.drawRect(53,290-cy[49]*14,4,cy[49]*14);
g.drawRect(48,290-cy[50]*14,4,cy[50]*14);
g.drawRect(43,290-cy[51]*14,4,cy[51]*14);
g.drawRect(38,290-cy[52]*14,4,cy[52]*14);
g.drawRect(33,290-cy[53]*14,4,cy[53]*14);
g.drawRect(28,290-cy[54]*14,4,cy[54]*14);
g.drawRect(23,290-cy[55]*14,4,cy[55]*14);
g.drawRect(18,290-cy[56]*14,4,cy[56]*14);
g.drawRect(13,290-cy[57]*14,4,cy[57]*14);
g.drawRect(8,290-cy[58]*14,4,cy[58]*14);
g.drawRect(3,290-cy[59]*14,4,cy[59]*14);

x=150;y=30;      //酔っぱらいの位置の初期化
k=0;             //ふらつき回数の初期化
ct++;           //実験回数

if (ct>60){ //実験回数が 60 回を越えたら実験を停止する
    flag=3;
}
} //if (k==30)
} //else if (flag==1 || flag==2)

//停止ボタンを押したときのイベント処理
if(flag==3) {
    g.clearRect(0,0,300,360);      //全体のクリア

//----- 棒グラフの描写の開始
g.drawRect(148,290-cy[0]*14,4,cy[0]*14);
g.drawRect(153,290-cy[1]*14,4,cy[1]*14);
g.drawRect(158,290-cy[2]*14,4,cy[2]*14);
g.drawRect(163,290-cy[3]*14,4,cy[3]*14);
g.drawRect(168,290-cy[4]*14,4,cy[4]*14);
g.drawRect(173,290-cy[5]*14,4,cy[5]*14);
g.drawRect(178,290-cy[6]*14,4,cy[6]*14);
g.drawRect(183,290-cy[7]*14,4,cy[7]*14);
g.drawRect(188,290-cy[8]*14,4,cy[8]*14);
g.drawRect(193,290-cy[9]*14,4,cy[9]*14);
g.drawRect(198,290-cy[10]*14,4,cy[10]*14);
g.drawRect(203,290-cy[11]*14,4,cy[11]*14);
g.drawRect(208,290-cy[12]*14,4,cy[12]*14);
g.drawRect(213,290-cy[13]*14,4,cy[13]*14);
g.drawRect(218,290-cy[14]*14,4,cy[14]*14);
g.drawRect(223,290-cy[15]*14,4,cy[15]*14);
g.drawRect(228,290-cy[16]*14,4,cy[16]*14);
g.drawRect(233,290-cy[17]*14,4,cy[17]*14);
g.drawRect(238,290-cy[18]*14,4,cy[18]*14);
g.drawRect(243,290-cy[19]*14,4,cy[19]*14);
g.drawRect(248,290-cy[20]*14,4,cy[20]*14);
g.drawRect(253,290-cy[21]*14,4,cy[21]*14);
g.drawRect(258,290-cy[22]*14,4,cy[22]*14);
g.drawRect(263,290-cy[23]*14,4,cy[23]*14);
g.drawRect(268,290-cy[24]*14,4,cy[24]*14);
g.drawRect(273,290-cy[25]*14,4,cy[25]*14);

```

```

g.drawRect (278,290-cy [26] *14,4,cy [26] *14);
g.drawRect (283,290-cy [27] *14,4,cy [27] *14);
g.drawRect (288,290-cy [28] *14,4,cy [28] *14);
g.drawRect (293,290-cy [29] *14,4,cy [29] *14);
g.drawRect (143,290-cy [31] *14,4,cy [31] *14);
g.drawRect (138,290-cy [32] *14,4,cy [32] *14);
g.drawRect (133,290-cy [33] *14,4,cy [33] *14);
g.drawRect (128,290-cy [34] *14,4,cy [34] *14);
g.drawRect (123,290-cy [35] *14,4,cy [35] *14);
g.drawRect (118,290-cy [36] *14,4,cy [36] *14);
g.drawRect (113,290-cy [37] *14,4,cy [37] *14);
g.drawRect (108,290-cy [38] *14,4,cy [38] *14);
g.drawRect (103,290-cy [39] *14,4,cy [39] *14);
g.drawRect (98,290-cy [40] *14,4,cy [40] *14);
g.drawRect (93,290-cy [41] *14,4,cy [41] *14);
g.drawRect (88,290-cy [42] *14,4,cy [42] *14);
g.drawRect (83,290-cy [43] *14,4,cy [43] *14);
g.drawRect (78,290-cy [44] *14,4,cy [44] *14);
g.drawRect (73,290-cy [45] *14,4,cy [45] *14);
g.drawRect (68,290-cy [46] *14,4,cy [46] *14);
g.drawRect (63,290-cy [47] *14,4,cy [47] *14);
g.drawRect (58,290-cy [48] *14,4,cy [48] *14);
g.drawRect (53,290-cy [49] *14,4,cy [49] *14);
g.drawRect (48,290-cy [50] *14,4,cy [50] *14);
g.drawRect (43,290-cy [51] *14,4,cy [51] *14);
g.drawRect (38,290-cy [52] *14,4,cy [52] *14);
g.drawRect (33,290-cy [53] *14,4,cy [53] *14);
g.drawRect (28,290-cy [54] *14,4,cy [54] *14);
g.drawRect (23,290-cy [55] *14,4,cy [55] *14);
g.drawRect (18,290-cy [56] *14,4,cy [56] *14);
g.drawRect (13,290-cy [57] *14,4,cy [57] *14);
g.drawRect (8,290-cy [58] *14,4,cy [58] *14);
g.drawRect (3,290-cy [59] *14,4,cy [59] *14);

g.drawString ("ふらつき回数 "+k,110,20);

g.drawLine (150,50-2,150,50+2); // y 軸 (上側)
g.drawLine (0,50,300,50); // x 軸 (上側)

for (xx=10;xx<=300;xx=xx+10) {
    g.drawLine (xx,48,xx,52); // x 軸メモリ線 (上側)
}

g.drawString ("0",148,63); // x 軸メモリ値 (上側)
g.drawString ("8",188,63);
g.drawString ("16",226,63);
g.drawString ("24",266,63);
g.drawString ("-8",106,63);
g.drawString ("-16",64,63);
g.drawString ("-24",24,63);
g.drawString (" x ",4,63);

g.drawLine (0,290,300,290); // x 軸 (下側)

for (xx=10;xx<300;xx=xx+10) {
    g.drawLine (xx,288,xx,292); // x 軸メモリ線 (下側)
}

g.drawString ("0",148,305); // x 軸メモリ値 (下側)
g.drawString ("8",188,305);
g.drawString ("16",226,305);
g.drawString ("24",266,305);
g.drawString ("-8",106,305);
g.drawString ("-16",64,305);

```

```
g.drawString("-24",24,305);
g.drawString(" x ",4,305);

g.drawImage(myImg,x-20,y,this); //酔っばらの表示
g.drawString("実験 "+ct+" 回目",30,330); //実験回数の表示
g.drawString("Copyright (C) K.Niwa 2001.12",130,330); //作者表示

    } //if (flag==3)
  } //if (myMt.checkID(0))
} //public void paint(Graphics g)
} //public class Randomwalk2 extends Applet implements Runnable
```

(2) HTML リスト

```
<HTML>
<HEAD>
< !-----
                                「ランダムウォーク」
                                Copyright (C) K.Niwa 2001.12.20
                                ----->
</HEAD>
<BODY>
  <CENTER>
    <B>「ランダムウォーク」</B>
    <BR><BR>
    <APPLET CODE="Randomwalk2.class" WIDTH="300" HEIGHT="360"></APPLET>
  </CENTER>
</BODY>
</HTML>
```

3. J a v a アプリケーション・プログラムリスト

```
//////////////////////////////////////
//
//                               「Fランダムウォーク2」
//                               Copyright (C) K.Niwa 2002.08.17
//                               ( J a v a アプリケーション)
//
//////////////////////////////////////

//クラスの読み込み
import java.awt.*;
import java.awt.event.*;
import java.lang.Math;

/***** public class FRandomwalk2 extends Frame implements Runnable *****/
public class FRandomwalk2 extends Frame implements Runnable { //スレッドをさせるようにする

//変数とオブジェクトの型宣言
  Thread myTh; //スレッド型で宣言する
  Button[] myBtn; //ボタン型で宣言する
  Panel myPanel; //パネル型で宣言する
  int x=150,y=30,oldx,oldy; //酔っばらの x 座標と y 座標
  int px; //酔っばらのふらつき 30 回目の x 座標
  int xx; //座標軸の描写に使用
  int k=0; //ふらつき回数に使用
  int flag=0;
  int ct=1; //実験回数
  int[] cy=new int[61]; //ふらつき 30 回目の位置のカウンタ
  long t; //時間稼ぎに使用
  double r; //乱数
```

```

Image myImg;           //イメージ型で宣言する
MediaTracker myMt;    //メディアトラッカー型で宣言する
int count;            //ループカウンター
int N;                //実験回数
int Speed=100;        //速度

/***** フレームとイベントの定義 *****/
public FRandomwalk2 () {

    setSize (300+10,360);           //フレームの大きさ
    addWindowListener (new WindowAdapter () { //閉じるボタンのイベント処理
        public void windowClosing (WindowEvent e) {
            System.exit (0);
        }
    });

    myTh=null;                       //スレッドの初期化
    if (myTh==null) {                 //スレッドの実体化
        myTh=new Thread (this);       //スレッドの開始
        myTh.start ();
    }
    //酔っぱらいの画像の読み込み
    myImg=Toolkit.getDefaultToolkit ().getImage ("hito1.jpg");

    myMt=new MediaTracker (this);      //メディアトラッカーの実体化
    myMt.addImage (myImg,0);          //画像をメディアトラッカーに貼り付ける

    try {
        myMt.waitForID (0);
    }
    catch (InterruptedException e) {
    }

    myBtn=new Button [4];             //ボタンの実体化
    myBtn [0]=new Button ("初期化");
    myBtn [1]=new Button ("高 速");
    myBtn [2]=new Button ("低 速");
    myBtn [3]=new Button ("停 止");

    myPanel=new Panel ();             //パネルの実体化
    myPanel.setLayout (new GridLayout (1,4));
    for (count=0;count<=3;count++) {
        myPanel.add (myBtn [count]); //パネルにボタンを貼り付ける
    }
    setLayout (new BorderLayout ()); //全体をボーダーレイアウトにする
    add ("South",myPanel);           //パネルを南に貼り付ける

    myBtn [0].addActionListener (new ActionListener () { //初期化ボタン
        public void actionPerformed (ActionEvent e) {
            flag=0;                   //ボタンの識別子
            x=150;y=30;               //酔っぱらいの位置の初期化
            k=0;                       //ふらつき回数の初期化
            ct=1;                       //実験回数の初期化
            repaint ();
        }
    });

    myBtn [1].addActionListener (new ActionListener () { //高速ボタン
        public void actionPerformed (ActionEvent e) {
            flag=1;                   //ボタンの識別子
            Speed=50;                 //速度
            //x=150;y=30;             //酔っぱらいの位置の初期化
            //k=0;                     //ふらつき回数の初期化
            //ct=1;                     //実験回数の初期化
        }
    });
}

```

```

        repaint();
    }
});

myBtn[2].addActionListener(new ActionListener() { //低速ボタン
    public void actionPerformed(ActionEvent e) {
        flag=2; //ボタンの識別子
        Speed=100; //速度
        //x=150;y=30; //酔っぱらいの位置の初期化
        //k=0; //ふらつき回数の初期化
        //ct=1; //実験回数の初期化
        repaint();
    }
});

myBtn[3].addActionListener(new ActionListener() { //停止ボタン
    public void actionPerformed(ActionEvent e) {
        flag=3; //ボタンの識別子
        repaint();
    }
});

} //public FRandomwalk2()

/***** public void run() メソッド *****/
public void run() {
    while (true) {
        try {
            myTh.sleep(Speed);
        }
        catch (InterruptedException e) {}
        if (flag==1 || flag==2) {
            repaint();
        }
    }
}

/***** public void paint(Graphics g) メソッド *****/
public void paint(Graphics g) {
    if (myMt.checkID(0)) {

        //初期状態または初期化ボタンを押したときのイベント処理
        if (flag==0) {
            for (count=0;count<=60;count++) {
                //ふらつき30回目の位置のカウントの初期化
                cy[count]=0;
            }

            //座標軸描写
            g.clearRect(0,0,300+10,360); //全体のクリア

            g.drawString("ふらつき回数 "+k,110,20);

            g.drawLine(150,50-2,150,50+2); //y軸(上側)
            g.drawLine(0,50,300,50); //x軸(上側)

            for (xx=10;xx<=300;xx=xx+10) {
                g.drawLine(xx,48,xx,52); //x軸メモリ線(上側)
            }

            g.drawString("0",148,63); //x軸メモリ値(上側)
            g.drawString("8",188,63);
        }
    }
}

```

```

g.drawString("16",226,63);
g.drawString("24",266,63);
g.drawString("-8",106,63);
g.drawString("-16",64,63);
g.drawString("-24",24,63);
g.drawString(" x ",4,63);

g.drawLine(0,290,300,290); // x 軸 (下側)

for (xx=10;xx<300;xx=xx+10) {
    g.drawLine(xx,288,xx,292); // x 軸メモリ線 (下側)
}

g.drawString("0",148,305); // x 軸メモリ値 (下側)
g.drawString("8",188,305);
g.drawString("16",226,305);
g.drawString("24",266,305);
g.drawString("-8",106,305);
g.drawString("-16",64,305);
g.drawString("-24",24,305);
g.drawString(" x ",4,305);

g.drawImage(myImg,x-20,y,this); //酔っぱらいの表示
g.drawString("実験 "+ct+" 回目",30,330); //実験回数の表示
g.drawString("Copyright(C) K.Niwa 2002.08",130,330); //作者表示

} //if(flag==0)

//高速ボタンまたは低速ボタンを押したときのイベント処理
else if (flag==1 || flag==2) {
//---- 棒グラフの描写の開始
g.drawRect(148,290-cy[0]*14,4,cy[0]*14);
g.drawRect(153,290-cy[1]*14,4,cy[1]*14);
g.drawRect(158,290-cy[2]*14,4,cy[2]*14);
g.drawRect(163,290-cy[3]*14,4,cy[3]*14);
g.drawRect(168,290-cy[4]*14,4,cy[4]*14);
g.drawRect(173,290-cy[5]*14,4,cy[5]*14);
g.drawRect(178,290-cy[6]*14,4,cy[6]*14);
g.drawRect(183,290-cy[7]*14,4,cy[7]*14);
g.drawRect(188,290-cy[8]*14,4,cy[8]*14);
g.drawRect(193,290-cy[9]*14,4,cy[9]*14);
g.drawRect(198,290-cy[10]*14,4,cy[10]*14);
g.drawRect(203,290-cy[11]*14,4,cy[11]*14);
g.drawRect(208,290-cy[12]*14,4,cy[12]*14);
g.drawRect(213,290-cy[13]*14,4,cy[13]*14);
g.drawRect(218,290-cy[14]*14,4,cy[14]*14);
g.drawRect(223,290-cy[15]*14,4,cy[15]*14);
g.drawRect(228,290-cy[16]*14,4,cy[16]*14);
g.drawRect(233,290-cy[17]*14,4,cy[17]*14);
g.drawRect(238,290-cy[18]*14,4,cy[18]*14);
g.drawRect(243,290-cy[19]*14,4,cy[19]*14);
g.drawRect(248,290-cy[20]*14,4,cy[20]*14);
g.drawRect(253,290-cy[21]*14,4,cy[21]*14);
g.drawRect(258,290-cy[22]*14,4,cy[22]*14);
g.drawRect(263,290-cy[23]*14,4,cy[23]*14);
g.drawRect(268,290-cy[24]*14,4,cy[24]*14);
g.drawRect(273,290-cy[25]*14,4,cy[25]*14);
g.drawRect(278,290-cy[26]*14,4,cy[26]*14);
g.drawRect(283,290-cy[27]*14,4,cy[27]*14);
g.drawRect(288,290-cy[28]*14,4,cy[28]*14);
g.drawRect(293,290-cy[29]*14,4,cy[29]*14);
g.drawRect(143,290-cy[31]*14,4,cy[31]*14);
g.drawRect(138,290-cy[32]*14,4,cy[32]*14);
g.drawRect(133,290-cy[33]*14,4,cy[33]*14);

```

```

g.drawRect(128,290-cy[34]*14,4,cy[34]*14);
g.drawRect(123,290-cy[35]*14,4,cy[35]*14);
g.drawRect(118,290-cy[36]*14,4,cy[36]*14);
g.drawRect(113,290-cy[37]*14,4,cy[37]*14);
g.drawRect(108,290-cy[38]*14,4,cy[38]*14);
g.drawRect(103,290-cy[39]*14,4,cy[39]*14);
g.drawRect(98,290-cy[40]*14,4,cy[40]*14);
g.drawRect(93,290-cy[41]*14,4,cy[41]*14);
g.drawRect(88,290-cy[42]*14,4,cy[42]*14);
g.drawRect(83,290-cy[43]*14,4,cy[43]*14);
g.drawRect(78,290-cy[44]*14,4,cy[44]*14);
g.drawRect(73,290-cy[45]*14,4,cy[45]*14);
g.drawRect(68,290-cy[46]*14,4,cy[46]*14);
g.drawRect(63,290-cy[47]*14,4,cy[47]*14);
g.drawRect(58,290-cy[48]*14,4,cy[48]*14);
g.drawRect(53,290-cy[49]*14,4,cy[49]*14);
g.drawRect(48,290-cy[50]*14,4,cy[50]*14);
g.drawRect(43,290-cy[51]*14,4,cy[51]*14);
g.drawRect(38,290-cy[52]*14,4,cy[52]*14);
g.drawRect(33,290-cy[53]*14,4,cy[53]*14);
g.drawRect(28,290-cy[54]*14,4,cy[54]*14);
g.drawRect(23,290-cy[55]*14,4,cy[55]*14);
g.drawRect(18,290-cy[56]*14,4,cy[56]*14);
g.drawRect(13,290-cy[57]*14,4,cy[57]*14);
g.drawRect(8,290-cy[58]*14,4,cy[58]*14);
g.drawRect(3,290-cy[59]*14,4,cy[59]*14);

```

```

// 1 回分の実験(ふらつき回数 30 回)
if (k<30) {
    k++; //ふらつき回数

    g.drawString("実験 "+ct+" 回目",30,330);//実験回数の表示
    g.clearRect(110,10,100,20);
    //ふらつき回数の表示
    g.drawString("ふらつき回数 "+k,110,20);

    oldx=x; //酔っぱらいの x 座標
    oldy=y; //酔っぱらいの y 座標

    r=Math.random();
    if (r<0.5) { //左へふらつく
        x=x-5;
    }
    else if (r>=0.5) { //右へふらつく
        x=x+5;
    }

    g.clearRect(oldx-20,oldy,41,43); //酔っぱらいの消去
    g.drawImage(myImg,x-20,y,this); //酔っぱらいの描写

    //座標軸描写
    g.drawLine(150,50-2,150,50+2); // y 軸 (上側)
    g.drawLine(0,50,300,50); // x 軸 (上側)

    for (xx=10;xx<=300;xx=xx+10) {
        g.drawLine(xx,48,xx,52);// x 軸メモリ線 (上側)
    }

    g.drawString("0",148,63); // x 軸メモリ値 (上側)
    g.drawString("8",188,63);
    g.drawString("16",226,63);
    g.drawString("24",266,63);
    g.drawString("-8",106,63);

```



```

g.drawString("-16",64,63);
g.drawString("-24",24,63);
g.drawString(" x ",4,63);

g.drawLine(0,290,300,290); // x 軸 (下側)

for (xx=10;xx<300;xx=xx+10) {
// x 軸メモリ線 (下側)
    g.drawLine(xx,288,xx,292);
}

g.drawString("0",148,305); // x 軸メモリ値 (下側)
g.drawString("8",188,305);
g.drawString("16",226,305);
g.drawString("24",266,305);
g.drawString("-8",106,305);
g.drawString("-16",64,305);
g.drawString("-24",24,305);
g.drawString(" x ",4,305);
//作者表示
g.drawString("Copyright(C) K.Niwa 2002.08",130,330);

} //if (k<=30)

//ふらつき 30 回目の位置の判定
if (k==30) {
    switch (x) { //酔っぱらいの x 座標 x
        case 150:
            cy[0]=cy[0]+1;
            break;
        case 155:
            cy[1]=cy[1]+1;
            break;
        case 160:
            cy[2]=cy[2]+1;
            break;
        case 165:
            cy[3]=cy[3]+1;
            break;
        case 170:
            cy[4]=cy[4]+1;
            break;
        case 175:
            cy[5]=cy[5]+1;
            break;
        case 180:
            cy[6]=cy[6]+1;
            break;
        case 185:
            cy[7]=cy[7]+1;
            break;
        case 190:
            cy[8]=cy[8]+1;
            break;
        case 195:
            cy[9]=cy[9]+1;
            break;
        case 200:
            cy[10]=cy[10]+1;
            break;
        case 205:
            cy[11]=cy[11]+1;
            break;
    }
}

```

```
case 210:
    cy[12]=cy[12]+1;
    break;
case 215:
    cy[13]=cy[13]+1;
    break;
case 220:
    cy[14]=cy[14]+1;
    break;
case 225:
    cy[15]=cy[15]+1;
    break;
case 230:
    cy[16]=cy[16]+1;
    break;
case 235:
    cy[17]=cy[17]+1;
    break;
case 240:
    cy[18]=cy[18]+1;
    break;
case 245:
    cy[19]=cy[19]+1;
    break;
case 250:
    cy[20]=cy[20]+1;
    break;
case 255:
    cy[21]=cy[21]+1;
    break;
case 260:
    cy[22]=cy[22]+1;
    break;
case 265:
    cy[23]=cy[23]+1;
    break;
case 270:
    cy[24]=cy[24]+1;
    break;
case 275:
    cy[25]=cy[25]+1;
    break;
case 280:
    cy[26]=cy[26]+1;
    break;
case 285:
    cy[27]=cy[27]+1;
    break;
case 290:
    cy[28]=cy[28]+1;
    break;
case 295:
    cy[29]=cy[29]+1;
    break;
case 145:
    cy[31]=cy[31]+1;
    break;
case 140:
    cy[32]=cy[32]+1;
    break;
case 135:
    cy[33]=cy[33]+1;
    break;
```

```
case 130:
    cy[34]=cy[34]+1;
    break;
case 125:
    cy[35]=cy[35]+1;
    break;
case 120:
    cy[36]=cy[36]+1;
    break;
case 115:
    cy[37]=cy[37]+1;
    break;
case 110:
    cy[38]=cy[38]+1;
    break;
case 105:
    cy[39]=cy[39]+1;
    break;
case 100:
    cy[40]=cy[40]+1;
    break;
case 95:
    cy[41]=cy[41]+1;
    break;
case 90:
    cy[42]=cy[42]+1;
    break;
case 85:
    cy[43]=cy[43]+1;
    break;
case 80:
    cy[44]=cy[44]+1;
    break;
case 75:
    cy[45]=cy[45]+1;
    break;
case 70:
    cy[46]=cy[46]+1;
    break;
case 65:
    cy[47]=cy[47]+1;
    break;
case 60:
    cy[48]=cy[48]+1;
    break;
case 55:
    cy[49]=cy[49]+1;
    break;
case 50:
    cy[50]=cy[50]+1;
    break;
case 45:
    cy[51]=cy[51]+1;
    break;
case 40:
    cy[52]=cy[52]+1;
    break;
case 35:
    cy[53]=cy[53]+1;
    break;
case 30:
    cy[54]=cy[54]+1;
    break;
```

```

case 25:
    cy[55]=cy[55]+1;
    break;
case 20:
    cy[56]=cy[56]+1;
    break;
case 15:
    cy[57]=cy[57]+1;
    break;
case 10:
    cy[58]=cy[58]+1;
    break;
case 5:
    cy[59]=cy[59]+1;
    break;

```

```

} //switch (x)

```

//---- 棒グラフの描写の開始

```

g.drawRect(148,290-cy[0]*14,4,cy[0]*14);
g.drawRect(153,290-cy[1]*14,4,cy[1]*14);
g.drawRect(158,290-cy[2]*14,4,cy[2]*14);
g.drawRect(163,290-cy[3]*14,4,cy[3]*14);
g.drawRect(168,290-cy[4]*14,4,cy[4]*14);
g.drawRect(173,290-cy[5]*14,4,cy[5]*14);
g.drawRect(178,290-cy[6]*14,4,cy[6]*14);
g.drawRect(183,290-cy[7]*14,4,cy[7]*14);
g.drawRect(188,290-cy[8]*14,4,cy[8]*14);
g.drawRect(193,290-cy[9]*14,4,cy[9]*14);
g.drawRect(198,290-cy[10]*14,4,cy[10]*14);
g.drawRect(203,290-cy[11]*14,4,cy[11]*14);
g.drawRect(208,290-cy[12]*14,4,cy[12]*14);
g.drawRect(213,290-cy[13]*14,4,cy[13]*14);
g.drawRect(218,290-cy[14]*14,4,cy[14]*14);
g.drawRect(223,290-cy[15]*14,4,cy[15]*14);
g.drawRect(228,290-cy[16]*14,4,cy[16]*14);
g.drawRect(233,290-cy[17]*14,4,cy[17]*14);
g.drawRect(238,290-cy[18]*14,4,cy[18]*14);
g.drawRect(243,290-cy[19]*14,4,cy[19]*14);
g.drawRect(248,290-cy[20]*14,4,cy[20]*14);
g.drawRect(253,290-cy[21]*14,4,cy[21]*14);
g.drawRect(258,290-cy[22]*14,4,cy[22]*14);
g.drawRect(263,290-cy[23]*14,4,cy[23]*14);
g.drawRect(268,290-cy[24]*14,4,cy[24]*14);
g.drawRect(273,290-cy[25]*14,4,cy[25]*14);
g.drawRect(278,290-cy[26]*14,4,cy[26]*14);
g.drawRect(283,290-cy[27]*14,4,cy[27]*14);
g.drawRect(288,290-cy[28]*14,4,cy[28]*14);
g.drawRect(293,290-cy[29]*14,4,cy[29]*14);
g.drawRect(143,290-cy[31]*14,4,cy[31]*14);
g.drawRect(138,290-cy[32]*14,4,cy[32]*14);
g.drawRect(133,290-cy[33]*14,4,cy[33]*14);
g.drawRect(128,290-cy[34]*14,4,cy[34]*14);
g.drawRect(123,290-cy[35]*14,4,cy[35]*14);
g.drawRect(118,290-cy[36]*14,4,cy[36]*14);
g.drawRect(113,290-cy[37]*14,4,cy[37]*14);
g.drawRect(108,290-cy[38]*14,4,cy[38]*14);
g.drawRect(103,290-cy[39]*14,4,cy[39]*14);
g.drawRect(98,290-cy[40]*14,4,cy[40]*14);
g.drawRect(93,290-cy[41]*14,4,cy[41]*14);
g.drawRect(88,290-cy[42]*14,4,cy[42]*14);
g.drawRect(83,290-cy[43]*14,4,cy[43]*14);
g.drawRect(78,290-cy[44]*14,4,cy[44]*14);
g.drawRect(73,290-cy[45]*14,4,cy[45]*14);
g.drawRect(68,290-cy[46]*14,4,cy[46]*14);

```

```

g.drawRect (63,290-cy[47]*14,4,cy[47]*14);
g.drawRect (58,290-cy[48]*14,4,cy[48]*14);
g.drawRect (53,290-cy[49]*14,4,cy[49]*14);
g.drawRect (48,290-cy[50]*14,4,cy[50]*14);
g.drawRect (43,290-cy[51]*14,4,cy[51]*14);
g.drawRect (38,290-cy[52]*14,4,cy[52]*14);
g.drawRect (33,290-cy[53]*14,4,cy[53]*14);
g.drawRect (28,290-cy[54]*14,4,cy[54]*14);
g.drawRect (23,290-cy[55]*14,4,cy[55]*14);
g.drawRect (18,290-cy[56]*14,4,cy[56]*14);
g.drawRect (13,290-cy[57]*14,4,cy[57]*14);
g.drawRect (8,290-cy[58]*14,4,cy[58]*14);
g.drawRect (3,290-cy[59]*14,4,cy[59]*14);

x=150;y=30;      //酔っばらの位置の初期化
k=0;             //ふらつき回数
ct++;           //実験回数

if (ct>60) { //実験回数が60回を越えたら実験を停止
    flag=3;
}
} //if (k==30)
} //else if (flag==1 || flag==2)

//停止ボタンを押したときのイベント処理
if (flag==3) {
    g.clearRect (0,0,300+10,360); //全体のクリア

//棒グラフの描写の開始
g.drawRect (148,290-cy[0]*14,4,cy[0]*14);
g.drawRect (153,290-cy[1]*14,4,cy[1]*14);
g.drawRect (158,290-cy[2]*14,4,cy[2]*14);
g.drawRect (163,290-cy[3]*14,4,cy[3]*14);
g.drawRect (168,290-cy[4]*14,4,cy[4]*14);
g.drawRect (173,290-cy[5]*14,4,cy[5]*14);
g.drawRect (178,290-cy[6]*14,4,cy[6]*14);
g.drawRect (183,290-cy[7]*14,4,cy[7]*14);
g.drawRect (188,290-cy[8]*14,4,cy[8]*14);
g.drawRect (193,290-cy[9]*14,4,cy[9]*14);
g.drawRect (198,290-cy[10]*14,4,cy[10]*14);
g.drawRect (203,290-cy[11]*14,4,cy[11]*14);
g.drawRect (208,290-cy[12]*14,4,cy[12]*14);
g.drawRect (213,290-cy[13]*14,4,cy[13]*14);
g.drawRect (218,290-cy[14]*14,4,cy[14]*14);
g.drawRect (223,290-cy[15]*14,4,cy[15]*14);
g.drawRect (228,290-cy[16]*14,4,cy[16]*14);
g.drawRect (233,290-cy[17]*14,4,cy[17]*14);
g.drawRect (238,290-cy[18]*14,4,cy[18]*14);
g.drawRect (243,290-cy[19]*14,4,cy[19]*14);
g.drawRect (248,290-cy[20]*14,4,cy[20]*14);
g.drawRect (253,290-cy[21]*14,4,cy[21]*14);
g.drawRect (258,290-cy[22]*14,4,cy[22]*14);
g.drawRect (263,290-cy[23]*14,4,cy[23]*14);
g.drawRect (268,290-cy[24]*14,4,cy[24]*14);
g.drawRect (273,290-cy[25]*14,4,cy[25]*14);
g.drawRect (278,290-cy[26]*14,4,cy[26]*14);
g.drawRect (283,290-cy[27]*14,4,cy[27]*14);
g.drawRect (288,290-cy[28]*14,4,cy[28]*14);
g.drawRect (293,290-cy[29]*14,4,cy[29]*14);
g.drawRect (143,290-cy[31]*14,4,cy[31]*14);
g.drawRect (138,290-cy[32]*14,4,cy[32]*14);
g.drawRect (133,290-cy[33]*14,4,cy[33]*14);
g.drawRect (128,290-cy[34]*14,4,cy[34]*14);
g.drawRect (123,290-cy[35]*14,4,cy[35]*14);

```

```

g.drawRect(118,290-cy[36]*14,4,cy[36]*14);
g.drawRect(113,290-cy[37]*14,4,cy[37]*14);
g.drawRect(108,290-cy[38]*14,4,cy[38]*14);
g.drawRect(103,290-cy[39]*14,4,cy[39]*14);
g.drawRect(98,290-cy[40]*14,4,cy[40]*14);
g.drawRect(93,290-cy[41]*14,4,cy[41]*14);
g.drawRect(88,290-cy[42]*14,4,cy[42]*14);
g.drawRect(83,290-cy[43]*14,4,cy[43]*14);
g.drawRect(78,290-cy[44]*14,4,cy[44]*14);
g.drawRect(73,290-cy[45]*14,4,cy[45]*14);
g.drawRect(68,290-cy[46]*14,4,cy[46]*14);
g.drawRect(63,290-cy[47]*14,4,cy[47]*14);
g.drawRect(58,290-cy[48]*14,4,cy[48]*14);
g.drawRect(53,290-cy[49]*14,4,cy[49]*14);
g.drawRect(48,290-cy[50]*14,4,cy[50]*14);
g.drawRect(43,290-cy[51]*14,4,cy[51]*14);
g.drawRect(38,290-cy[52]*14,4,cy[52]*14);
g.drawRect(33,290-cy[53]*14,4,cy[53]*14);
g.drawRect(28,290-cy[54]*14,4,cy[54]*14);
g.drawRect(23,290-cy[55]*14,4,cy[55]*14);
g.drawRect(18,290-cy[56]*14,4,cy[56]*14);
g.drawRect(13,290-cy[57]*14,4,cy[57]*14);
g.drawRect(8,290-cy[58]*14,4,cy[58]*14);
g.drawRect(3,290-cy[59]*14,4,cy[59]*14);

g.drawString("ふらつき回数 "+k,110,20);
g.drawLine(150,50-2,150,50+2); // y 軸 (上側)
g.drawLine(0,50,300,50); // x 軸 (上側)

for (xx=10;xx<=300;xx=xx+10) {
    g.drawLine(xx,48,xx,52); // x 軸メモリ線 (上側)
}

g.drawString("0",148,63); // x 軸メモリ値 (上側)
g.drawString("8",188,63);
g.drawString("16",226,63);
g.drawString("24",266,63);
g.drawString("-8",106,63);
g.drawString("-16",64,63);
g.drawString("-24",24,63);
g.drawString(" x ",4,63);

g.drawLine(0,290,300,290); // x 軸 (下側)

for (xx=10;xx<300;xx=xx+10) {
    g.drawLine(xx,288,xx,292); // x 軸メモリ線 (下側)
}

g.drawString("0",148,305); // x 軸メモリ値 (下側)
g.drawString("8",188,305);
g.drawString("16",226,305);
g.drawString("24",266,305);
g.drawString("-8",106,305);
g.drawString("-16",64,305);
g.drawString("-24",24,305);
g.drawString(" x ",4,305);

g.drawImage(myImg,x-20,y,this); //酔っぱらいの表示
g.drawString("実験 "+ct+" 回目",30,330); //実験回数の表示
g.drawString("Copyright(C) K.Niwa 2002.08",130,330); //作者表示

} //if(flag==3)
} //if (myMt.checkID(0))
} //public void paint(Graphics g)

```

```
/****** public static void main メソッド *****  
public static void main(String[] args) {  
    Frame w=new FRandomwalk2();  
    w.show();  
} //public static void main(String[] args)  
  
} //public class FRandomwalk2 extends Frame implements Runnable
```