

【3の倍数のトランプカード】

1. はじめに



[J a v a アプレット]



[J a v a アプリケーション]

ここにトランプが1組あります。ジョーカー2枚を除いて52枚を使います。
3の倍数は、スペード、クローバ、ダイヤ、ハートに、それぞれ3と6と9と12の4枚ずつあるので、 $4 \times 4 = 16$ 枚あります。
この52枚のトランプから1枚引いたとき、そのカードが3の倍数である確率を考えます。
52枚のうち16枚が3の倍数のカードなので、数学的には、 $16 / 52$ になります。
しかし、実際には、1枚ずつトランプを引いては戻すことを52回行ったら、3の倍数のカードがちょうど16回出るなんてことはありませんね。
では、数学的に求めた理論上の確率の $16 / 52$ との関係はどうなっているのでしょうか。
シミュレーションソフト「3の倍数のトランプカード」を使って、3の倍数のカードが出る様子を観察してみてください。

2. J a v a アプレット

(1) J a v a プログラムリスト

```
////////////////////////////////////  
//  
//          「3の倍数のトランプカード」  
//          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;
```

```
/** public class Toranpu2 extends Applet implements Runnable */  
public class Toranpu2 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 imgNo; //整数型で宣言する
Image[] img; //イメージ型で宣言する
MediaTracker myMt;
int flag=0;
int r=40; //乱数
int ct1=0; //実験回数
long t; //時間かせぎ
int d=0; //度数
int ct,ctt;
int N=1000; //速度
int count; //ループカウンター
```

```
/****** public void init()メソッド *****/
```

```
public void init() {
    setBackground(Color.lightGray); //背景色をライトグレーにする
    myTh=null; //スレッドの初期化
    imgNo=0;
    img=new Image[53]; //イメージの実体化

    img[1]=getImage(getDocumentBase(),"d01.jpg"); //ダイヤのエース画像の読み込み
    img[2]=getImage(getDocumentBase(),"d02.jpg");
    img[3]=getImage(getDocumentBase(),"d03.jpg");
    img[4]=getImage(getDocumentBase(),"d04.jpg");
    img[5]=getImage(getDocumentBase(),"d05.jpg");
    img[6]=getImage(getDocumentBase(),"d06.jpg");
    img[7]=getImage(getDocumentBase(),"d07.jpg");
    img[8]=getImage(getDocumentBase(),"d08.jpg");
    img[9]=getImage(getDocumentBase(),"d09.jpg");
    img[10]=getImage(getDocumentBase(),"d10.jpg");
    img[11]=getImage(getDocumentBase(),"d11.jpg");
    img[12]=getImage(getDocumentBase(),"d12.jpg");
    img[13]=getImage(getDocumentBase(),"d13.jpg");
    img[14]=getImage(getDocumentBase(),"h01.jpg"); //ハートのエース画像の読み込み
    img[15]=getImage(getDocumentBase(),"h02.jpg");
    img[16]=getImage(getDocumentBase(),"h03.jpg");
    img[17]=getImage(getDocumentBase(),"h04.jpg");
    img[18]=getImage(getDocumentBase(),"h05.jpg");
    img[19]=getImage(getDocumentBase(),"h06.jpg");
    img[20]=getImage(getDocumentBase(),"h07.jpg");
    img[21]=getImage(getDocumentBase(),"h08.jpg");
    img[22]=getImage(getDocumentBase(),"h09.jpg");
    img[23]=getImage(getDocumentBase(),"h10.jpg");
    img[24]=getImage(getDocumentBase(),"h11.jpg");
    img[25]=getImage(getDocumentBase(),"h12.jpg");
    img[26]=getImage(getDocumentBase(),"h13.jpg");
    img[27]=getImage(getDocumentBase(),"m01.jpg"); //クローバのエース画像の読み込み
    img[28]=getImage(getDocumentBase(),"m02.jpg");
    img[29]=getImage(getDocumentBase(),"m03.jpg");
    img[30]=getImage(getDocumentBase(),"m04.jpg");
    img[31]=getImage(getDocumentBase(),"m05.jpg");
    img[32]=getImage(getDocumentBase(),"m06.jpg");
    img[33]=getImage(getDocumentBase(),"m07.jpg");
    img[34]=getImage(getDocumentBase(),"m08.jpg");
    img[35]=getImage(getDocumentBase(),"m09.jpg");
    img[36]=getImage(getDocumentBase(),"m10.jpg");
    img[37]=getImage(getDocumentBase(),"m11.jpg");
    img[38]=getImage(getDocumentBase(),"m12.jpg");
    img[39]=getImage(getDocumentBase(),"m13.jpg");
    img[40]=getImage(getDocumentBase(),"s01.jpg"); //スペードのエース画像の読み込み
}
```

```

img[41]=getImage(getDocumentBase(),"s02.jpg");
img[42]=getImage(getDocumentBase(),"s03.jpg");
img[43]=getImage(getDocumentBase(),"s04.jpg");
img[44]=getImage(getDocumentBase(),"s05.jpg");
img[45]=getImage(getDocumentBase(),"s06.jpg");
img[46]=getImage(getDocumentBase(),"s07.jpg");
img[47]=getImage(getDocumentBase(),"s08.jpg");
img[48]=getImage(getDocumentBase(),"s09.jpg");
img[49]=getImage(getDocumentBase(),"s10.jpg");
img[50]=getImage(getDocumentBase(),"s11.jpg");
img[51]=getImage(getDocumentBase(),"s12.jpg");
img[52]=getImage(getDocumentBase(),"s13.jpg");

myMt=new MediaTracker(this);           //メディアトラッカーの実体化

for (count=1;count<=52;count++) {
    myMt.addImage(img[count],0);      //ランプ画像のメディアトラッカーへの貼り付け
}

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

myBtn=new Button[4];
myBtn[0]=new Button("初期化");       //ボタンの実体化
myBtn[1]=new Button("高速");         //ボタンの実体化
myBtn[2]=new Button("低速");         //ボタンの実体化
myBtn[3]=new Button("停止");         //ボタンの実体化
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() { //初期化ボタンを押したとき
    public void actionPerformed(ActionEvent e) {
        flag=0;           //識別子
        repaint();        //paint()メソッドを呼び出す
    }
});

myBtn[1].addActionListener(new ActionListener() { //高速ボタンを押したとき
    public void actionPerformed(ActionEvent e) {
        flag=1;           //識別子
        N=100;            //スピード0.1秒
        repaint();        //paint()メソッドを呼び出す
    }
});

myBtn[2].addActionListener(new ActionListener() { //低速ボタンを押したとき
    public void actionPerformed(ActionEvent e) {
        flag=2;           //識別子
        N=1000;           //スピード1秒
        repaint();        //paint()メソッドを呼び出す
    }
});

```

```

        myBtn[3].addActionListener(new ActionListener() { //停止ボタンを押したとき
            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 *****/
    public void run() {
        while (true) {
            try {
                myTh.sleep(N); //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)) { //メディアトラッカーに全て読み込まれているかチェック

            //初期状態と初期化ボタンを押したときのイベント処理
            if (flag==0) {
                g.clearRect(0,0,300,350); //全体のクリア
                d=0; //引いた3の倍数のカード枚数の初期化
                ct1=0; //引いたすべてのカードの枚数の初期化
                r=40; //スペードのエース
                g.drawImage(img[40],115,30,this); //トランプの表示

                g.drawString("引いた3の倍数のカード枚数=",20+40,110+50);
                g.drawString("引いたすべてのカードの枚数=",20+40,130+50);
                g.drawString("3の倍数を引いた割合=",20+40,150+70);

                g.drawString("3の倍数を引く理論上の割合 = 16/52 =",
                    20+40,170+260);
                g.drawString("0.30769231",23,260);
            } //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) {
                ct1++; //実験回数

                g.drawImage(img[40],115,30,this); //トランプの表示
                g.drawString("引いた3の倍数のカード枚数=",20+40,160);
                g.drawString(""+d,180+40,160);
                g.drawString("引いたすべてのカードの枚数=",20+40,180);
                g.drawString(""+ct1,180+40,180);
                g.drawString("3の倍数を引いた割合 =",20+40,220);
                g.drawString(""+((float) d)/(float) (ct1)),150+40,220);
                g.drawString("3の倍数を引く理論上の割合 = 16/52 =",
                    20+40,240+260);
                g.drawString("0.30769231",23,260);
            }
        }
    }
}

```

```

//作者表示
g.drawString("Copyright (C) K.Niwa 2001.12",80,325);

//判断前のトランプを切る点減
for (ctt=1;ctt<=52;ctt++){
    g.drawImage(img[ctt],115,30,this);
    for (t=1;t<=100000;t++){ //時間稼ぎ
    }
}

//トランプが3の倍数のカードかどうかの判断の開始
r=(int)(1+52*Math.random()); 1～52の乱数
switch (r){
    case 1: //ダイヤのエース
        g.drawImage(img[1],115,30,this);
        break;
    case 2:
        g.drawImage(img[2],115,30,this);
        break;
    case 3:
        g.drawImage(img[3],115,30,this);
        d=d+1;
        break;
    case 4:
        g.drawImage(img[4],115,30,this);
        break;
    case 5:
        g.drawImage(img[5],115,30,this);
        break;
    case 6:
        g.drawImage(img[6],115,30,this);
        d=d+1;
        break;
    case 7:
        g.drawImage(img[7],115,30,this);
        break;
    case 8:
        g.drawImage(img[8],115,30,this);
        break;
    case 9:
        g.drawImage(img[9],115,30,this);
        d=d+1;
        break;
    case 10:
        g.drawImage(img[10],115,30,this);
        break;
    case 11:
        g.drawImage(img[11],115,30,this);
        break;
    case 12:
        g.drawImage(img[12],115,30,this);
        d=d+1;
        break;
    case 13:
        g.drawImage(img[13],115,30,this);
        break;
    case 14: //ハートのエース
        g.drawImage(img[14],115,30,this);
        break;
    case 15:
        g.drawImage(img[15],115,30,this);
        break;
}

```

```

case 16:
    g.drawImage (img [16],115,30,this) ;
    d=d+1;
    break;
case 17:
    g.drawImage (img [17],115,30,this) ;
    break;
case 18:
    g.drawImage (img [18],115,30,this) ;
    break;
case 19:
    g.drawImage (img [19],115,30,this) ;
    d=d+1;
    break;
case 20:
    g.drawImage (img [20],115,30,this) ;
    break;
case 21:
    g.drawImage (img [21],115,30,this) ;
    break;
case 22:
    g.drawImage (img [22],115,30,this) ;
    d=d+1;
    break;
case 23:
    g.drawImage (img [23],115,30,this) ;
    break;
case 24:
    g.drawImage (img [24],115,30,this) ;
    break;
case 25:
    g.drawImage (img [25],115,30,this) ;
    d=d+1;
    break;
case 26:
    g.drawImage (img [26],115,30,this) ;
    break;
case 27: //クローバのエース
    g.drawImage (img [27],115,30,this) ;
    break;
case 28:
    g.drawImage (img [28],115,30,this) ;
    break;
case 29:
    g.drawImage (img [29],115,30,this) ;
    d=d+1;
    break;
case 30:
    g.drawImage (img [30],115,30,this) ;
    break;
case 31:
    g.drawImage (img [31],115,30,this) ;
    break;
case 32:
    g.drawImage (img [32],115,30,this) ;
    d=d+1;
    break;
case 33:
    g.drawImage (img [33],115,30,this) ;
    break;
case 34:
    g.drawImage (img [34],115,30,this) ;
    break;

```

```

case 35:
    g.drawImage (img [35],115,30,this) ;
    d=d+1;
    break;
case 36:
    g.drawImage (img [36],115,30,this) ;
    break;
case 37:
    g.drawImage (img [37],115,30,this) ;
    break;
case 38:
    g.drawImage (img [38],115,30,this) ;
    d=d+1;
    break;
case 39:
    g.drawImage (img [39],115,30,this) ;
    break;
case 40: //スペースのエース
    g.drawImage (img [40],115,30,this) ;
    break;
case 41:
    g.drawImage (img [41],115,30,this) ;
    break;
case 42:
    g.drawImage (img [42],115,30,this) ;
    d=d+1;
    break;
case 43:
    g.drawImage (img [43],115,30,this) ;
    break;
case 44:
    g.drawImage (img [44],115,30,this) ;
    break;
case 45:
    g.drawImage (img [45],115,30,this) ;
    d=d+1;
    break;
case 46:
    g.drawImage (img [46],115,30,this) ;
    break;
case 47:
    g.drawImage (img [47],115,30,this) ;
    break;
case 48:
    g.drawImage (img [48],115,30,this) ;
    d=d+1;
    break;
case 49:
    g.drawImage (img [49],115,30,this) ;
    break;
case 50:
    g.drawImage (img [50],115,30,this) ;
    break;
case 51:
    g.drawImage (img [51],115,30,this) ;
    d=d+1;
    break;
case 52:
    g.drawImage (img [52],115,30,this) ;
    break;
} //switch (r)

```

```

//トランプが3の倍数のカードかどうかの判断の終了
g.clearRect(0,140,300,100); //枠の消去

g.drawString("引いた3の倍数のカード枚数=",20+40,160);
g.drawString(""+d,180+40,160);

g.drawString("引いたすべてのカードの枚数=",20+40,180);
g.drawString(""+ct1,180+40,180);

g.drawString("3の倍数を引いた割合=",20+40,220);
g.drawString(""+((float)(d)/(float)(ct1)),150+40,220);

} //else if (flag==1 || flag==2)

//停止ボタンを押したときのイベント処理
else if (flag==3) {
    //トランプカードの表示
    switch (r) {
        case 1: //ダイヤのエース
            g.drawImage(img[1],115,30,this);
            break;
        case 2:
            g.drawImage(img[2],115,30,this);
            break;
        case 3:
            g.drawImage(img[3],115,30,this);
            break;
        case 4:
            g.drawImage(img[4],115,30,this);
            break;
        case 5:
            g.drawImage(img[5],115,30,this);
            break;
        case 6:
            g.drawImage(img[6],115,30,this);
            break;
        case 7:
            g.drawImage(img[7],115,30,this);
            break;
        case 8:
            g.drawImage(img[8],115,30,this);
            break;
        case 9:
            g.drawImage(img[9],115,30,this);
            break;
        case 10:
            g.drawImage(img[10],115,30,this);
            break;
        case 11:
            g.drawImage(img[11],115,30,this);
            break;
        case 12:
            g.drawImage(img[12],115,30,this);
            break;
        case 13:
            g.drawImage(img[13],115,30,this);
            break;
        case 14: //ハートのエース
            g.drawImage(img[14],115,30,this);
            break;
        case 15:
            g.drawImage(img[15],115,30,this);
            break;
    }
}

```



```
case 16:
    g.drawImage (img [16],115,30,this) ;
    break;
case 17:
    g.drawImage (img [17],115,30,this) ;
    break;
case 18:
    g.drawImage (img [18],115,30,this) ;
    break;
case 19:
    g.drawImage (img [19],115,30,this) ;
    break;
case 20:
    g.drawImage (img [20],115,30,this) ;
    break;
case 21:
    g.drawImage (img [21],115,30,this) ;
    break;
case 22:
    g.drawImage (img [22],115,30,this) ;
    break;
case 23:
    g.drawImage (img [23],115,30,this) ;
    break;
case 24:
    g.drawImage (img [24],115,30,this) ;
    break;
case 25:
    g.drawImage (img [25],115,30,this) ;
    break;
case 26:
    g.drawImage (img [26],115,30,this) ;
    break;
case 27: //クローバのエース
    g.drawmage (img [27],115,30,this) ;
    break;
case 28:
    g.drawImage (img [28],115,30,this) ;
    break;
case 29:
    g.drawImage (img [29],115,30,this) ;
    break;
case 30:
    g.drawImage (img [30],115,30,this) ;
    break;
case 31:
    g.drawImage (img [31],115,30,this) ;
    break;
case 32:
    g.drawImage (img [32],115,30,this) ;
    break;
case 33:
    g.drawImage (img [33],115,30,this) ;
    break;
case 34:
    g.drawImage (img [34],115,30,this) ;
    break;
case 35:
    g.drawImage (img [35],115,30,this) ;
    break;
case 36:
    g.drawImage (img [36],115,30,this) ;
    break;
```

```

case 37:
    g.drawImage (img [37],115,30,this) ;
    break;
case 38:
    g.drawImage (img [38],115,30,this) ;
    break;
case 39:
    g.drawImage (img [39],115,30,this) ;
    break;
case 40: //スペースのエース
    g.drawImage (img [40],115,30,this) ;
    break;
case 41:
    g.drawImage (img [41],115,30,this) ;
    break;
case 42:
    g.drawImage (img [42],115,30,this) ;
    break;
case 43:
    g.drawImage (img [43],115,30,this) ;
    break;
case 44:
    g.drawImage (img [44],115,30,this) ;
    break;
case 45:
    g.drawImage (img [45],115,30,this) ;
    break;
case 46:
    g.drawImage (img [46],115,30,this) ;
    break;
case 47:
    g.drawImage (img [47],115,30,this) ;
    break;
case 48:
    g.drawImage (img [48],115,30,this) ;
    break;
case 49:
    g.drawImage (img [49],115,30,this) ;
    break;
case 50:
    g.drawImage (img [50],115,30,this) ;
    break;
case 51:
    g.drawImage (img [51],115,30,this) ;
    break;
case 52:
    g.drawImage (img [52],115,30,this) ;
    break;
} //switch (r)
//トランプカードの表示の終了

g.clearRect (0,140,300,100) ; //枠の消去

g.drawString ("引いた 3 の倍数のカード枚数=",20+40,160) ;
g.drawString (""+d,180+40,160) ;

g.drawString ("引いたすべてのカードの枚数=",20+40,180) ;
g.drawString (""+ct1,180+40,180) ;

g.drawString (" 3 の倍数を引いた割合=",20+40,220) ;

```



```

int imgNo;
Image[] img; //イメージ型で宣言する
MediaTracker myMt;
int flag=0;
int r=40; //乱数
int ct1=0; //実験回数
long t; //時間かせぎ
int d=0; //度数
int ct,ctt;
int N=1000; //速度
int count; //ループカウンター

/***** フレームとイベントの定義 *****/
public FToranpu2 () {

    setSize (300+20,360); //フレームの大きさ

    addWindowListener(new WindowAdapter () { //フレームの定義
        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);
        myTh.start ();
    }

    setBackground (Color.lightGray); //背景色の設定

    imgNo=0;
    img=new Image [53]; //イメージの実体化

//ダイヤのエース画像の読み込み
img [1]=Toolkit.getDefaultToolkit ().getImage ("d01.jpg");
img [2]=Toolkit.getDefaultToolkit ().getImage ("d02.jpg");
img [3]=Toolkit.getDefaultToolkit ().getImage ("d03.jpg");
img [4]=Toolkit.getDefaultToolkit ().getImage ("d04.jpg");
img [5]=Toolkit.getDefaultToolkit ().getImage ("d05.jpg");
img [6]=Toolkit.getDefaultToolkit ().getImage ("d06.jpg");
img [7]=Toolkit.getDefaultToolkit ().getImage ("d07.jpg");
img [8]=Toolkit.getDefaultToolkit ().getImage ("d08.jpg");
img [9]=Toolkit.getDefaultToolkit ().getImage ("d09.jpg");
img [10]=Toolkit.getDefaultToolkit ().getImage ("d10.jpg");
img [11]=Toolkit.getDefaultToolkit ().getImage ("d11.jpg");
img [12]=Toolkit.getDefaultToolkit ().getImage ("d12.jpg");
img [13]=Toolkit.getDefaultToolkit ().getImage ("d13.jpg");

//ハートのエース画像の読み込み
img [14]=Toolkit.getDefaultToolkit ().getImage ("h01.jpg");
img [15]=Toolkit.getDefaultToolkit ().getImage ("h02.jpg");
img [16]=Toolkit.getDefaultToolkit ().getImage ("h03.jpg");
img [17]=Toolkit.getDefaultToolkit ().getImage ("h04.jpg");
img [18]=Toolkit.getDefaultToolkit ().getImage ("h05.jpg");
img [19]=Toolkit.getDefaultToolkit ().getImage ("h06.jpg");
img [20]=Toolkit.getDefaultToolkit ().getImage ("h07.jpg");
img [21]=Toolkit.getDefaultToolkit ().getImage ("h08.jpg");
img [22]=Toolkit.getDefaultToolkit ().getImage ("h09.jpg");
img [23]=Toolkit.getDefaultToolkit ().getImage ("h10.jpg");
img [24]=Toolkit.getDefaultToolkit ().getImage ("h11.jpg");
img [25]=Toolkit.getDefaultToolkit ().getImage ("h12.jpg");
img [26]=Toolkit.getDefaultToolkit ().getImage ("h13.jpg");

```

```

//クローバのエース画像の読み込み
img[27]=Toolkit.getDefaultToolkit().getImage("m01.jpg");
img[28]=Toolkit.getDefaultToolkit().getImage("m02.jpg");
img[29]=Toolkit.getDefaultToolkit().getImage("m03.jpg");
img[30]=Toolkit.getDefaultToolkit().getImage("m04.jpg");
img[31]=Toolkit.getDefaultToolkit().getImage("m05.jpg");
img[32]=Toolkit.getDefaultToolkit().getImage("m06.jpg");
img[33]=Toolkit.getDefaultToolkit().getImage("m07.jpg");
img[34]=Toolkit.getDefaultToolkit().getImage("m08.jpg");
img[35]=Toolkit.getDefaultToolkit().getImage("m09.jpg");
img[36]=Toolkit.getDefaultToolkit().getImage("m10.jpg");
img[37]=Toolkit.getDefaultToolkit().getImage("m11.jpg");
img[38]=Toolkit.getDefaultToolkit().getImage("m12.jpg");
img[39]=Toolkit.getDefaultToolkit().getImage("m13.jpg");
//スペースのエース画像の読み込み
img[40]=Toolkit.getDefaultToolkit().getImage("s01.jpg");
img[41]=Toolkit.getDefaultToolkit().getImage("s02.jpg");
img[42]=Toolkit.getDefaultToolkit().getImage("s03.jpg");
img[43]=Toolkit.getDefaultToolkit().getImage("s04.jpg");
img[44]=Toolkit.getDefaultToolkit().getImage("s05.jpg");
img[45]=Toolkit.getDefaultToolkit().getImage("s06.jpg");
img[46]=Toolkit.getDefaultToolkit().getImage("s07.jpg");
img[47]=Toolkit.getDefaultToolkit().getImage("s08.jpg");
img[48]=Toolkit.getDefaultToolkit().getImage("s09.jpg");
img[49]=Toolkit.getDefaultToolkit().getImage("s10.jpg");
img[50]=Toolkit.getDefaultToolkit().getImage("s11.jpg");
img[51]=Toolkit.getDefaultToolkit().getImage("s12.jpg");
img[52]=Toolkit.getDefaultToolkit().getImage("s13.jpg");

myMt=new MediaTracker(this); //メディアトラッカーの実体化

for (count=1;count<=52;count++) {
    myMt.addImage(img[count],0); //トランプ画像のメディアトラッカーへの貼り付け
}

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

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

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]);
}
setLayout(new BorderLayout());
add("South",myPanel);

myBtn[0].addActionListener(new ActionListener() { //初期化ボタンを押したとき
    public void actionPerformed(ActionEvent e) {
        flag=0; //識別子
        repaint(); //paint()メソッドを呼び出す
    }
});

```

```

myBtn[1].addActionListener(new ActionListener() { //高速ボタンを押したとき
    public void actionPerformed(ActionEvent e) {
        flag=1;
        N=100;
        repaint();
    }
});

myBtn[2].addActionListener(new ActionListener() { //低速ボタンを押したとき
    public void actionPerformed(ActionEvent e) {
        flag=2;
        N=1000;
        repaint();
    }
});

myBtn[3].addActionListener(new ActionListener() { //停止ボタンを押したとき
    public void actionPerformed(ActionEvent e) {
        flag=3;
        repaint();
    }
});

} //public FToranpu2()

/***** public void run() メソッド *****/
public void run() {
    while (true) {
        try {
            myTh.sleep(N); //休む
        }
        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)) {
        //初期状態と初期化ボタンを押したときのイベント処理
        if (flag==0) {
            g.clearRect(0,0,300,350); //全体のクリア
            d=0; //引いた3の倍数のカード枚数の初期化
            ct1=0; //引いたすべてのカードの枚数の初期化
            r=40; //スペードのエース
            g.drawImage(img[40],115,30,this); //トランプの表示

            g.drawString("引いた3の倍数のカード枚数=",20+40,110+50);
            g.drawString("引いたすべてのカードの枚数=",20+40,130+50);
            g.drawString("3の倍数を引いた割合=",20+40,150+70);
            g.drawString("3の倍数を引く理論上の割合 = 16/52 =",
                730,23,260);
            g.drawString("Copyright(C) K.Niwa 2002.08",80,325); //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) {
            ct1++; //実験回数

            g.drawImage(img[40],115,30,this); //トランプの表示

```

```

g.drawString("引いた3の倍数のカード枚数=",20+40,160)
;
g.drawString(""+d,180+40+10,160);
g.drawString("引いたすべてのカードの枚数=",20+40,180)
;
g.drawString(""+ct1,180+40+10,180);
g.drawString("3の倍数を引いた割合=",20+40,220);
g.drawString(""+((float)(d)/(float)(ct1)),150+40+5,220);
g.drawString("3の倍数を引く理論上の割合 = 16/52 =
0.30769231",23,260);
//作者表示
g.drawString("Copyright (C) K.Niwa 2002.08",80,325);
//判断前のトランプを切る点減-
for (ctt=1;ctt<=52;ctt++) {
    g.drawImage(img[ctt],115,30,this);
    for (t=1;t<=100000;t++) {} //時間稼ぎ
}
//トランプが3の倍数のカードかどうかの判断の開始
r=(int)(1+52*Math.random());
switch (r) {
    case 1:
        //ダイヤのエース
        g.drawImage(img[1],115,30,this);
        break;
    case 2:
        g.drawImage(img[2],115,30,this);
        break;
    case 3:
        g.drawImage(img[3],115,30,this);
        d=d+1;
        break;
    case 4:
        g.drawImage(img[4],115,30,this);
        break;
    case 5:
        g.drawImage(img[5],115,30,this);
        break;
    case 6:
        g.drawImage(img[6],115,30,this);
        d=d+1;
        break;
    case 7:
        g.drawImage(img[7],115,30,this);
        break;
    case 8:
        g.drawImage(img[8],115,30,this);
        break;
    case 9:
        g.drawImage(img[9],115,30,this);
        d=d+1;
        break;
    case 10:
        g.drawImage(img[10],115,30,this);
        break;
    case 11:
        g.drawImage(img[11],115,30,this);
        break;
}

```

```

case 12:
    g.drawImage (img [12],115,30,this) ;
    d=d+1;
    break;
case 13:
    g.drawImage (img [13],115,30,this) ;
    break;
case 14:
    //ハートのエース
    g.drawImage (img [14],115,30,this) ;
    break;
case 15:
    g.drawImage (img [15],115,30,this) ;
    break;
case 16:
    g.drawImage (img [16],115,30,this) ;
    d=d+1;
    break;
case 17:
    g.drawImage (img [17],115,30,this) ;
    break;
case 18:
    g.drawImage (img [18],115,30,this) ;
    break;
case 19:
    g.drawImage (img [19],115,30,this) ;
    d=d+1;
    break;
case 20:
    g.drawImage (img [20],115,30,this) ;
    break;
case 21:
    g.drawImage (img [21],115,30,this) ;
    break;
case 22:
    g.drawImage (img [22],115,30,this) ;
    d=d+1;
    break;
case 23:
    g.drawImage (img [23],115,30,this) ;
    break;
case 24:
    g.drawImage (img [24],115,30,this) ;
    break;
case 25:
    g.drawImage (img [25],115,30,this) ;
    d=d+1;
    break;
case 26:
    g.drawImage (img [26],115,30,this) ;
    break;
case 27:
    //クローバのエース
    g.drawImage (img [27],115,30,this) ;
    break;
case 28:
    g.drawImage (img [28],115,30,this) ;
    break;
case 29:
    g.drawImage (img [29],115,30,this) ;
    d=d+1;
    break;

```



```

case 30:
    g.drawImage (img [30],115,30,this) ;
    break;
case 31:
    g.drawImage (img [31],115,30,this) ;
    break;
case 32:
    g.drawImage (img [32],115,30,this) ;
    d=d+1;
    break;
case 33:
    g.drawImage (img [33],115,30,this) ;
    break;
case 34:
    g.drawImage (img [34],115,30,this) ;
    break;
case 35:
    g.drawImage (img [35],115,30,this) ;
    d=d+1;
    break;
case 36:
    g.drawImage (img [36],115,30,this) ;
    break;
case 37:
    g.drawImage (img [37],115,30,this) ;
    break;
case 38:
    g.drawImage (img [38],115,30,this) ;
    d=d+1;
    break;
case 39:
    g.drawImage (img [39],115,30,this) ;
    break;
case 40:
    //スペースのエース
    g.drawImage (img [40],115,30,this) ;
    break;
case 41:
    g.drawImage (img [41],115,30,this) ;
    break;
case 42:
    g.drawImage (img [42],115,30,this) ;
    d=d+1;
    break;
case 43:
    g.drawImage (img [43],115,30,this) ;
    break;
case 44:
    g.drawImage (img [44],115,30,this) ;
    break;
case 45:
    g.drawImage (img [45],115,30,this) ;
    d=d+1;
    break;
case 46:
    g.drawImage (img [46],115,30,this) ;
    break;
case 47:
    g.drawImage (img [47],115,30,this) ;
    break;
case 48:
    g.drawImage (img [48],115,30,this) ;
    d=d+1;
    break;

```

```

        case 49:
            g.drawImage (img [49],115,30,this);
            break;
        case 50:
            g.drawImage (img [50],115,30,this);
            break;
        case 51:
            g.drawImage (img [51],115,30,this);
            d=d+1;
            break;
        case 52:
            g.drawImage (img [52],115,30,this);
            break;

    }//switch (r)
    //トランプが 3 の倍数のカードかどうかの判断の終了

    g.clearRect (0,140,300,100); //枠の消去

    g.drawString ("引いた 3 の倍数のカード枚数=",20+40,160);
    g.drawString (""+d,180+40+10,160);

    g.drawString ("引いたすべてのカードの枚数=",20+40,180);
    g.drawString (""+ct1,180+40+10,180);

    g.drawString (" 3 の倍数を引いた割合=",20+40,220);
    g.drawString (""+((float) (d)/(float) (ct1)),150+40+5,220);

    }//else if (flag==1 || flag==2)
//停止ボタン を押したときのイベント処理
    else if (flag==3) {
        //トランプカードの表示 -
        switch (r) {
            case 1:
                //ダイヤのエース
                g.drawImage (img [1],115,30,this);
                break;
            case 2:
                g.drawImage (img [2],115,30,this);
                break;
            case 3:
                g.drawImage (img [3],115,30,this);
                break;
            case 4:
                g.drawImage (img [4],115,30,this);
                break;
            case 5:
                g.drawImage (img [5],115,30,this);
                break;
            case 6:
                g.drawImage (img [6],115,30,this);
                break;
            case 7:
                g.drawImage (img [7],115,30,this);
                break;
            case 8:
                g.drawImage (img [8],115,30,this);
                break;
            case 9:
                g.drawImage (img [9],115,30,this);
                break;
        }
    }
}

```

```

case 10:
    g.drawImage (img [10],115,30,this) ;
    break;
case 11:
    g.drawImage (img [11],115,30,this) ;
    break;
case 12:
    g.drawImage (img [12],115,30,this) ;
    break;
case 13:
    g.drawImage (img [13],115,30,this) ;
    break;
case 14:
    //ハートのエース
    g.drawImage (img [14],115,30,this) ;
    break;
case 15:
    g.drawImage (img [15],115,30,this) ;
    break;
case 16:
    g.drawImage (img [16],115,30,this) ;
    break;
case 17:
    g.drawImage (img [17],115,30,this) ;
    break;
case 18:
    g.drawImage (img [18],115,30,this) ;
    break;
case 19:
    g.drawImage (img [19],115,30,this) ;
    break;
case 20:
    g.drawImage (img [20],115,30,this) ;
    break;
case 21:
    g.drawImage (img [21],115,30,this) ;
    break;
case 22:
    g.drawImage (img [22],115,30,this) ;
    break;
case 23:
    g.drawImage (img [23],115,30,this) ;
    break;
case 24:
    g.drawImage (img [24],115,30,this) ;
    break;
case 25:
    g.drawImage (img [25],115,30,this) ;
    break;
case 26:
    g.drawImage (img [26],115,30,this) ;
    break;
    //クローバのエース
case 27:
    g.drawImage (img [27],115,30,this) ;
    break;
case 28:
    g.drawImage (img [28],115,30,this) ;
    break;
case 29:
    g.drawImage (img [29],115,30,this) ;
    break;

```

```
case 30:
    g.drawImage (img [30],115,30,this) ;
    break;
case 31:
    g.drawImage (img [31],115,30,this) ;
    break;
case 32:
    g.drawImage (img [32],115,30,this) ;
    break;
case 33:
    g.drawImage (img [33],115,30,this) ;
    break;
case 34:
    g.drawImage (img [34],115,30,this) ;
    break;
case 35:
    g.drawImage (img [35],115,30,this) ;
    break;
case 36:
    g.drawImage (img [36],115,30,this) ;
    break;
case 37:
    g.drawImage (img [37],115,30,this) ;
    break;
case 38:
    g.drawImage (img [38],115,30,this) ;
    break;
case 39:
    g.drawImage (img [39],115,30,this) ;
    break;
case 40:
    //スペースのエース
    g.drawImage (img [40],115,30,this) ;
    break;
case 41:
    g.drawImage (img [41],115,30,this) ;
    break;
case 42:
    g.drawImage (img [42],115,30,this) ;
    break;
case 43:
    g.drawImage (img [43],115,30,this) ;
    break;
case 44:
    g.drawImage (img [44],115,30,this) ;
    break;
case 45:
    g.drawImage (img [45],115,30,this) ;
    break;
case 46:
    g.drawImage (img [46],115,30,this) ;
    break;
case 47:
    g.drawImage (img [47],115,30,this) ;
    break;
case 48:
    g.drawImage (img [48],115,30,this) ;
    break;
case 49:
    g.drawImage (img [49],115,30,this) ;
    break;
case 50:
    g.drawImage (img [50],115,30,this) ;
    break;
```

```

        case 51:
            g.drawImage(img[51],115,30,this);
            break;
        case 52:
            g.drawImage(img[52],115,30,this);
            break;
    }//switch (r)
    //トランプカードの表示の終了

    g.clearRect(0,140,300,100);          //枠の消去

    g.drawString("引いた3の倍数のカード枚数=",20+40,160);
    g.drawString(""+d,180+40+10,160);

    g.drawString(""+ct1,180+40+10,180);

    g.drawString("3の倍数を引いた割合=",20+40,220);
    if (ct1>0){
        g.drawString(""+((float)(d)/(float)(ct1))
                    ,150+40+5,220);
    }

    g.drawString("3の倍数を引く理論上の割合 = 16/52 =
                0.30769231",23,260);
    //作者表示
    g.drawString("Copyright (C) K.Niwa 2002.08",80,325);

    }//else 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) {          //フレームを表示する
    Frame w=new FToranpu2();
    w.show();
} //public static void main(String[] args)

} //public class FToranpu2 extends Frame implements Runnable

```