

【2個のコイントス】

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]

2個のコインを同時に投げて、その表裏の出方を考えてみましょう。
 この2個のコインをそれぞれコイン1・コイン2とすると、その出方は順に、表・表、表・裏、裏・表、裏・裏の4通りあります。
 つまり、この4つの場合の起こる確率は、数学的には、それぞれ $1/4$ になります。
 しかし、実際には、2個のコインを同時に4回投げたら2個とも表になることが1回だけ起こるとは限りませんね。
 また、他の3つの場合も同様ですね。
 では、数学的に求めた理論上の確率の $1/4$ 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「2個のコイントス」を使って、2個のコインを投げたときの表裏の出方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

```

////////////////////////////////////
//                                     //
//                                     //
//                                     //
//                                     //
//                                     //
//                                     //
//                                     //
//                                     //
//                                     //
//                                     //
////////////////////////////////////

/***** 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 Coins2 extends Applet implements Runnable *****/
public class Coins2 extends Applet implements Runnable {

    Thread myTh;           //スレッド型として宣言する
    Button[] myBtn;       //ボタン型として宣言する
    Panel myPanel;        //パネル型として宣言する
    int imgNo;
    Image[] img;          //イメージ型として宣言する
    MediaTracker myMt;    //メディアトラッカー型として宣言する
    int flag=0;
    double r1=0.1,r2=0.9; //乱数
    int ct1;              //実験回数
    long t;               //時間かせぎ
    int d1=0,d2=0,d3=0,d4=0; //度数
    int ct;               //ループカウンター
    int N=1000;           //速度 (1 秒)
    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;                        //スレッドの実体化
    imgNo=0;

    img=new Image [2];                //イメージの実体化
    img [0]=getImage (getDocumentBase (), "coin1.jpg"); //コイン表画像読み込み
    img [1]=getImage (getDocumentBase (), "coin2.jpg"); //コイン裏画像読み込み

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

    for (count=0;count<=1;count++) {
        myMt.addImage (img [count],0); //メディアトラッカーへのイメージの貼り付け
    }

    try {
        myMt.waitForID (0);           //メディアトラッカーに全ての画像が
    }                                  //読み込まれたかをチェックする。
    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++) {  //パネルにボタンを貼り付ける
        myPanel.add (myBtn [count]);
    }

    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
            r1=0.1;r2=0.9;            //乱数
            repaint ();                //paint ()メソッドを呼び出す
        }
    });
}

```

```

myBtn[1].addActionListener(new ActionListener() { //高速ボタン
    public void actionPerformed(ActionEvent e) {
        flag=1; //識別子
        N=100; //スピード
        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
        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
    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() { //イベントが発生しなくても処理を行う
    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) { //ほとんどのプログラムはここに書く
    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(0,0,300,360); //全体のクリア
            d1=0;d2=0;d3=0;d4=0; //度数の初期化
            ct1=0; //実験回数の初期化
            g.drawImage(img[0],90,30,this); //コイン1の表示
            g.drawString("コイン1",90,75);
            g.drawImage(img[1],180,30,this); //コイン2の表示
            g.drawString("コイン2",180,75);

            g.drawString("コイン1 表 表 裏
            g.drawString("コイン2 表 裏 裏
            g.drawString("裏",40,110);
            g.drawString("裏",40,110);

```

```

                                                                 裏",40,130);
//枠
g.drawString("度    数",40,150);
g.drawLine(25,95,269,95);
g.drawLine(25,115,269,115);
g.drawLine(25,135,269,135);
g.drawLine(25,155,269,155);
g.drawLine(25,95,25,155);
g.drawLine(94,95,94,155);
g.drawLine(137,95,137,155);
g.drawLine(181,95,181,155);
g.drawLine(225,95,225,155);
g.drawLine(269,95,269,155);

g.drawString("実験回数="+ct1,110,20); //実験回数の表示
g.drawString("Copyright(C) K.Niwa 2001.12",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++;
    //g.clearRect(0,0,300,360); //全体のクリア

    //度数の表示
    g.drawString(""+d1,125-20+5,150);
    g.drawString(""+d2,169-20+5,150);
    g.drawString(""+d3,213-20+5,150);
    g.drawString(""+d4,257-20+5,150);

    //割合の表示
    if (d1+d2+d3+d4!=0) {
        g.drawString(" [表・表] の割合="+((float) (d1)/
            (float) (d1+d2+d3+d4)),80,200-10);
        g.drawString(" [表・裏] の割合="+((float) (d2)/
            (float) (d1+d2+d3+d4)),80,230-10);
        g.drawString(" [裏・表] の割合="+((float) (d3)/
            (float) (d1+d2+d3+d4)),80,260-10);
        g.drawString(" [裏・裏] の割合="+((float) (d4)/
            (float) (d1+d2+d3+d4)),80,290-10);
    }

    g.drawString("実験回数="+ct1,110,20); //実験回数の表示

    //作者表示
    g.drawString("Copyright(C) K.Niwa 2001.12",80,325);
    g.drawString("コイン 1",90,75);
    g.drawString("コイン 2",180,75);

    g.drawString("コイン 1          表          表          裏
                                                                 裏",40,110);
    g.drawString("コイン 2          表          裏          表
                                                                 裏",40,130);
    g.drawString("度    数",40,150);

    //枠
    g.drawLine(25,95,269,95);
    g.drawLine(25,115,269,115);
    g.drawLine(25,135,269,135);
    g.drawLine(25,155,269,155);
    g.drawLine(25,95,25,155);
    g.drawLine(94,95,94,155);
    g.drawLine(137,95,137,155);
    g.drawLine(181,95,181,155);
    g.drawLine(225,95,225,155);

```

```

g.drawLine(269,95,269,155);
//判断前のコイン1とコイン2の裏表の点滅
for (ct=1;ct<=3;ct++){
    g.drawImage(img[0],90,30,this);
    g.drawImage(img[1],180,30,this);
    for (t=1;t<=1000000;t++){ //時間稼ぎ
        g.drawImage(img[1],90,30,this);
        g.drawImage(img[0],180,30,this);
    }
}

//コイン1の表裏の判断
r1=Math.random(); //乱数を発生する
if (r1<0.5){
    g.drawImage(img[0],90,30,this);
}
else{
    g.drawImage(img[1],90,30,this);
}

//コイン2の表裏の判断
r2=Math.random();
if (r2<0.5){ //乱数を発生する
    g.drawImage(img[0],180,30,this);
}
else{
    g.drawImage(img[1],180,30,this);
}

//度数のカウント
if (r1<0.5 && r2<0.5){
    d1=d1+1;
}
else if (r1<0.5 && r2>=0.5){
    d2=d2+1;
}
else if (r1>=0.5 && r2<0.5){
    d3=d3+1;
}
else if (r1>=0.5 && r2>=0.5){
    d4=d4+1;
}

g.clearRect(0,95,300,200); //枠の消去

//枠の項目
g.drawString("コイン1    表    表    裏",40,110);
g.drawString("コイン2    表    裏    表",40,130);
g.drawString("度    数",40,150);

//枠
g.drawLine(25,95,269,95);
g.drawLine(25,115,269,115);
g.drawLine(25,135,269,135);
g.drawLine(25,155,269,155);
g.drawLine(25,95,25,155);
g.drawLine(94,95,94,155);
g.drawLine(137,95,137,155);
g.drawLine(181,95,181,155);
g.drawLine(225,95,225,155);
g.drawLine(269,95,269,155);

```

```

//度数の表示
g.drawString(""+d1,125-20+5,150);
g.drawString(""+d2,169-20+5,150);
g.drawString(""+d3,213-20+5,150);
g.drawString(""+d4,257-20+5,150);

//割合の表示
g.drawString(" [表・表] の割合="+((float) (d1)/(float)
(d1+d2+d3+d4)),80,200-10);
g.drawString(" [表・裏] の割合="+((float) (d2)/(float)
(d1+d2+d3+d4)),80,230-10);
g.drawString(" [裏・表] の割合="+((float) (d3)/(float)
(d1+d2+d3+d4)),80,260-10);
g.drawString(" [裏・裏] の割合="+((float) (d4)/(float)
(d1+d2+d3+d4)),80,290-10);

g.drawString("実験回数="+ct1,110,20); //実験回数の表示

} //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) {
//コイン 1 の画像表示
if (r1<0.5) {
g.drawImage(img[0],90,30,this);
}
else {
g.drawImage(img[1],90,30,this);
}
g.drawString("コイン 1 ",90,75);

//コイン 2 の画像表示-
if (r2<0.5) {
g.drawImage(img[0],180,30,this);
}
else {
g.drawImage(img[1],180,30,this);
}
g.drawString("コイン 2 ",180,75);

g.clearRect(0,95,300,200); //枠の消去

//枠の項目
g.drawString("コイン 1      表      表      裏
裏",40,110);
g.drawString("コイン 2      表      裏      表
裏",40,130);
g.drawString("度      数",40,150);

//枠
g.drawLine(25,95,269,95);
g.drawLine(25,115,269,115);
g.drawLine(25,135,269,135);
g.drawLine(25,155,269,155);
g.drawLine(25,95,25,155);
g.drawLine(94,95,94,155);
g.drawLine(137,95,137,155);
g.drawLine(181,95,181,155);
g.drawLine(225,95,225,155);
g.drawLine(269,95,269,155);

```



```

import java.awt.event.*;
import java.lang.Math;

/***** public class FCoins2 extends Frame implements Runnable *****/
public class FCoins2 extends Frame implements Runnable {
    Thread myTh; //スレッド型として宣言する
    Button[] myBtn; //ボタン型として宣言する
    Panel myPanel; //パネル型として宣言する
    int imgNo;
    Image[] img; //イメージ型として宣言する
    MediaTracker myMt; //メディアトラッカー型として宣言する
    int flag=0;
    double r1=0.1,r2=0.9; //乱数
    int ct1; //実験回数
    long t; //時間かせぎ
    int d1=0,d2=0,d3=0,d4=0; //度数
    int ct; //ループカウンター
    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 FCoins2 () {

        setSize (300,360); //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) {
            myTh=new Thread (this); //スレッドの実体化
            myTh.start (); //スレッドをスタートする
        }

        imgNo=0;
        img=new Image [2]; //イメージの実体化
        img [0]=Toolkit.getDefaultToolkit ().getImage ("coin1.jpg"); //コイン表画像読み込み
        img [1]=Toolkit.getDefaultToolkit ().getImage ("coin2.jpg"); //コイン裏画像読み込み

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

        for (count=0;count<=1;count++) {
            myMt.addImage (img [count],0); //メディアトラッカーへのイメージの貼り付け
        }

        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 ();
        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;
        r1=0.1;r2=0.9;                 //乱数
        repaint();
    }
});

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

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

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

} //public FCoins2()

/***** 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(0,0,300,360); //全体のクリア
            d1=0;d2=0;d3=0;d4=0;      //度数の初期化
            ct1=0;                     //実験回数の初期化

            g.drawImage(img[0],90,30,this); //コイン1の表示
            g.drawString("コイン1",90-5,75);

```

```

g.drawImage(img[1],180,30,this); //コイン2の表示
g.drawString("コイン2",180-5,75);

g.drawString("コイン1 表 表 裏 裏",40,110);
g.drawString("コイン2 表 裏 表 裏",40,130);
g.drawString("度数",40,150);

//枠
g.drawLine(25,95,269,95);
g.drawLine(25,115,269,115);
g.drawLine(25,135,269,135);
g.drawLine(25,155,269,155);
g.drawLine(25,95,25,155);
g.drawLine(94,95,94,155);
g.drawLine(137,95,137,155);
g.drawLine(181,95,181,155);
g.drawLine(225,95,225,155);
g.drawLine(269,95,269,155);

g.drawString("実験回数="+ct1,110,20); //実験回数の表示

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++;
    //g.clearRect(0,0,300,360); //全体のクリア

    //度数の表示
    g.drawString(""+d1,125-20-5,150);
    g.drawString(""+d2,169-20-5,150);
    g.drawString(""+d3,213-20-5,150);
    g.drawString(""+d4,257-20-5,150);

    //割合の表示
    if (d1+d2+d3+d4!=0) {
        g.drawString(" [表・表] の割合="+((float)(d1)/
            (float)(d1+d2+d3+d4)),80,200-10);
        g.drawString(" [表・裏] の割合="+((float)(d2)/
            (float)(d1+d2+d3+d4)),80,230-10);
        g.drawString(" [裏・表] の割合="+((float)(d3)/
            (float)(d1+d2+d3+d4)),80,260-10);
        g.drawString(" [裏・裏] の割合="+((float)(d4)/
            (float)(d1+d2+d3+d4)),80,290-10);
    }
    //実験回数の表示
    g.drawString("実験回数="+ct1,110,20);
    //作者表示
    g.drawString("Copyright(C) K.Niwa 2002.08",80,325);
    g.drawString("コイン1",90-5,75);
    g.drawString("コイン2",180-5,75);

    g.drawString("コイン1 表 表 裏 裏",40,110);
    g.drawString("コイン2 表 裏 表 裏",40,130);
    //g.drawString("コイン1 表 裏",40,110);
    //g.drawString("コイン2 裏 表",40,130);
    g.drawString("度数",40,150);

```

```

//枠
g.drawLine (25,95,269,95);
g.drawLine (25,115,269,115);
g.drawLine (25,135,269,135);
g.drawLine (25,155,269,155);
g.drawLine (25,95,25,155);
g.drawLine (94,95,94,155);
g.drawLine (137,95,137,155);
g.drawLine (181,95,181,155);
g.drawLine (225,95,225,155);
g.drawLine (269,95,269,155);

//判断前のコイン1 とコイン2 の裏表の点滅
for (ct=1;ct<=3;ct++) {
    g.drawImage (img[0],90,30,this);
    g.drawImage (img[1],180,30,this);
    for (t=1;t<=1000000;t++) {}
    g.drawImage (img[1],90,30,this);
    g.drawImage (img[0],180,30,this);
    for (t=1;t<=1000000;t++) {}
}

//コイン1 の表裏の判断
r1=Math.random();
if (r1<0.5) {
    g.drawImage (img[0],90,30,this);
}
else {
    g.drawImage (img[1],90,30,this);
}

//コイン2 の表裏の判断
r2=Math.random();
if (r2<0.5) {
    g.drawImage (img[0],180,30,this);
}
else {
    g.drawImage (img[1],180,30,this);
}

//度数のカウント
if (r1<0.5 && r2<0.5) {
    d1=d1+1;
}
else if (r1<0.5 && r2>=0.5) {
    d2=d2+1;
}
else if (r1>=0.5 && r2<0.5) {
    d3=d3+1;
}
else if (r1>=0.5 && r2>=0.5) {
    d4=d4+1;
}

g.clearRect(0,95,300,200); //枠の消去

//枠の項目
g.drawString("コイン1   表   表   裏   裏
",40,110);
g.drawString("コイン2   表   裏   表   裏
",40,130);
//g.drawString("コイン1           表   表   裏
",40,110);

```

```

//g.drawString("コイン 2      表      裏
g.drawString("度数",40,150);

//枠
g.drawLine(25,95,269,95);
g.drawLine(25,115,269,115);
g.drawLine(25,135,269,135);
g.drawLine(25,155,269,155);
g.drawLine(25,95,25,155);
g.drawLine(94,95,94,155);
g.drawLine(137,95,137,155);
g.drawLine(181,95,181,155);
g.drawLine(225,95,225,155);
g.drawLine(269,95,269,155);

//度数の表示
g.drawString(""+d1,125-20-5,150);
g.drawString(""+d2,169-20-5,150);
g.drawString(""+d3,213-20-5,150);
g.drawString(""+d4,257-20-5,150);

//割合の表示
g.drawString(" [表・表] の割合="+((float)(d1)/(float)
(d1+d2+d3+d4)),80,200-10);
g.drawString(" [表・裏] の割合="+((float)(d2)/(float)
(d1+d2+d3+d4)),80,230-10);
g.drawString(" [裏・表] の割合="+((float)(d3)/(float)
(d1+d2+d3+d4)),80,260-10);
g.drawString(" [裏・裏] の割合="+((float)(d4)/(float)
(d1+d2+d3+d4)),80,290-10);

g.drawString("実験回数="+ct1,110,20); //実験回数の表示

} //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) {
//コイン 1 の画像表示
if (r1<0.5) {
g.drawImage(img[0],90,30,this);
}
else {
g.drawImage(img[1],90,30,this);
}
g.drawString("コイン 1 ",90-5,75);

//コイン 2 の画像表示
if (r2<0.5) {
g.drawImage(img[0],180,30,this);
}
else {
g.drawImage(img[1],180,30,this);
}
g.drawString("コイン 2 ",180-5,75);

g.clearRect(0,95,300,200); //枠の消去

//枠の項目
g.drawString("コイン 1      表      表      裏      裏
",40,110);
g.drawString("コイン 2      表      裏      表      裏
",40,130);
//g.drawString("コイン 1      表      表

```

```

//g.drawString("コイン 2
g.drawString("度数",40,150);

//枠
g.drawLine(25,95,269,95);
g.drawLine(25,115,269,115);
g.drawLine(25,135,269,135);
g.drawLine(25,155,269,155);
g.drawLine(25,95,25,155);
g.drawLine(94,95,94,155);
g.drawLine(137,95,137,155);
g.drawLine(181,95,181,155);
g.drawLine(225,95,225,155);
g.drawLine(269,95,269,155);

//度数の表示
g.drawString(""+d1,125-20-5,150);
g.drawString(""+d2,169-20-5,150);
g.drawString(""+d3,213-20-5,150);
g.drawString(""+d4,257-20-5,150);

//割合の表示
if (d1+d2+d3+d4!=0) {
    g.drawString("[表・表] の割合="+((float) (d1)/
(float) (d1+d2+d3+d4)),80,200-10);
    g.drawString("[表・裏] の割合="+((float) (d2)/
(float) (d1+d2+d3+d4)),80,230-10);
    g.drawString("[裏・表] の割合="+((float) (d3)/
(float) (d1+d2+d3+d4)),80,260-10);
    g.drawString("[裏・裏] の割合="+((float) (d4)/
(float) (d1+d2+d3+d4)),80,290-10);
}

//実験回数の表示
g.drawString("実験回数="+ct1,110,20);

//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) {
    Frame w=new FCoins2();
    w.show();

} //public static void main(String[] args)

} //public class FCoins2 extends Frame implements Runnable

```

裏
表
裏
表

,40,110);
,40,130);