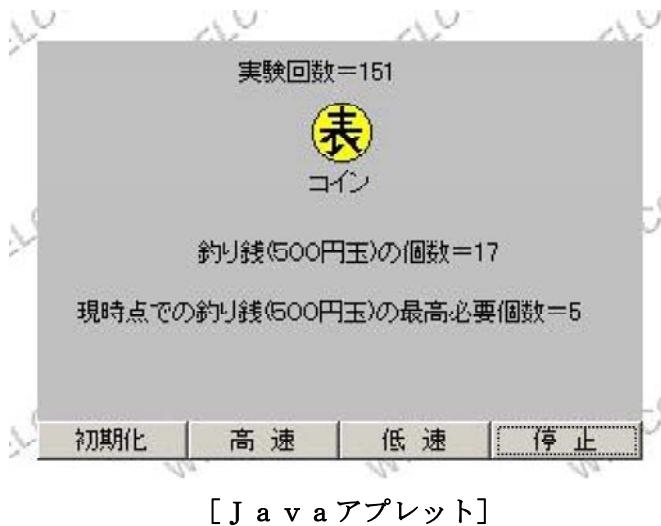


【釣り銭の用意の実験】



1. はじめに

クラス会などの幹事を務めることはありますか。

幹事になったつもりで考えてみてください。

仮に、クラス会への参加者人数は35人で、会費は3500円あるとします。

また、参加者は、1000円札4枚でお釣りを必要とする人と、1000円札3枚と500円玉1個でお釣りの要らない人の2通りだけあるとします。

会費は、会場に到着した者から順に集めて、お釣りの要る人にはその時その場でお釣りの500円玉を渡します。

ただし、経験上、お釣りの要る人と要らない人の割合は半々であるとします。

さあ、幹事のあなたはお釣りの500円玉を何個用意しておけばよいのでしょうか。

【シミュレーションの考え方】

お釣りの必要な人と必要でない人の割合は半々なので、1枚のコインを投げて、表が出たらお釣りの必要でない人が来たと考え、裏が出たらお釣りの必要な人が来たと考えます。

つまり、表が出たら3000円と500円が支払われる所以、幹事の手元に500円玉が1個増えることになります。

また、裏が出たら4000円が支払われる所以、お釣りが必要となり、幹事の手元から500円玉が1個減ることになります。

シミュレーションソフト「釣り銭の用意の実験」を使って、お釣りの500円玉を何個用意しておけばよいのかを観察してみてください。

2. Java アプレット

(1) Java プログラムリスト

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//                                     「釣り銭の用意の実験」
//                                     Copyright (C) K.Niwa 2001.12.20
//                                     (Java 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 Turisen2 extends Applet implements Runnable{           //スレッドを使えるようにする

    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;                                    //メディアトラッカー型で宣言する
    int flag=0;                                          //整数型で宣言し、0を代入する
    double r;                                            //乱数
    int ct1;                                             //実験回数
    long t;                                              //時間かせぎ
    int d=0;                                             //釣り銭の個数
    int ct;                                              //コインの裏表の点滅ループカウンター
    int min=0;                                           //現時点での釣り銭の最高必要個数
    int N=1000;                                          //速度

/********* public void init() メソッド *****/
public void init() {
    setBackground(Color.lightGray);                      //背景色の設定
    myTh=null;                                         //スレッドの初期化
    int count;                                          //ループカウンター
    imgNo=0;                                            //イメージの実体化
    img=new Image[2];                                   

    img[0]=getImage(getDocumentBase(),"coin1.jpg");    //コイン表画像の読み込み
    img[1]=getImage(getDocumentBase(),"coin2.jpg");    //コイン裏画像の読み込み

    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に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();                            //パネルの実態化
        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
                r=0.1;
                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;
        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;
        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); //スレッドを実体化する
        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>*****
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
            ct1=0; //実験回数
            d=0; //釣り銭の個数の初期化
            min=0; //現時点での釣り銭の最高必要個数の初期化
            g.drawString("実験回数="+ct1,100,20); //実験回数の表示
            g.drawImage(img[0],135,30,this); //コインの表示
            g.drawString("コイン",135,75);
            g.drawString("釣り銭(500円玉)の個数="+d,80,120-10);
            g.drawString("現時点での釣り銭(500円玉)の最高必要個数="+
                (-1*min),20,150-10);

        //作者表示
        g.drawString("Copyright(C) K.Niwa 2001.12",80,180+5+40);
    }
}

```

```

} //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.drawImage(img[0],135,30,this);           //コインの表示
    g.drawString("コイン",135,75);

    g.drawString("釣り銭(500円玉)の個数=" + d, 80, 120-10);

    g.drawString("現時点での釣り銭(500円玉)の最高必要個数=" + (-1*min), 20, 150-10);

    g.drawString("実験回数=" + ct1, 100, 20); //実験回数の表示

    //作者表示
    g.drawString("Copyright(C) K.Niwa 2001.12", 80, 180+5+40);

    //判断前のコインの裏表の点滅
    for (ct=1;ct<=3;ct++) {
        g.drawImage(img[0],135,30,this);
        for (t=1;t<=3000000;t++) {}
        g.drawImage(img[1],135,30,this);
        for (t=1;t<=3000000;t++) {}
    }

    //コインの表裏の判断
    r=Math.random();
    if (r<0.5) {
        g.drawImage(img[0],135,30,this); //表が出た
        d=d+1;
    }
    else {
        g.drawImage(img[1],135,30,this); //裏が出た
        d=d-1;
    }

    g.clearRect(0,100,300,60);           //部分消去

    g.drawString("釣り銭(500円玉)の個数=" + d, 80, 120-10);

    //現時点での釣り銭(500円玉)の最高必要個数の判断
    if (d<=min) {
        min=d;
    }

    g.drawString("現時点での釣り銭(500円玉)の最高必要個数=" + (-1*min), 20, 150-10);

    //実験回数の表示
    g.drawString("実験回数=" + ct1, 100, 20);

} //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) {
    g.drawImage(img[0],135,30,this);           //コインの表示
    g.drawString("コイン",135,75);

    g.drawString("釣り銭(500円玉)の個数=" + d, 80, 120-10);

    g.drawString("現時点での釣り銭(500円玉)の最高必要個数=" + (-1*min), 20, 150-10);

    //実験回数の表示
    g.drawString("実験回数=" + ct1, 100, 20);
}

```

```

//作者表示
g.drawString("Copyright(C) K.Niwa 2001.12",80,180+5+40);

//コインの表示
if (r<0.5) {
    g.drawImage(img[0],135,30,this); //表が出た
} else {
    g.drawImage(img[1],135,30,this); //裏が出た
}

g.clearRect(0,100,300,60); //部分消去

g.drawString("釣り銭(500円玉)の個数="+d,80,120-10);

//現時点での釣り銭(500円玉)の最高必要個数の判断
if (d<=min) {
    min=d;
}

g.drawString("現時点での釣り銭(500円玉)の最高必要  
個数=" + (-1*min),20,150-10);

//実験回数の表示
g.drawString("実験回数="+ct1,100,20);

} //else if (flag==3)
} //if (myMt.checkID(0))
} //public void paint(Graphics g)
} //public class Turisen2 extends Applet implements Runnable

```

(2) HTMLリスト

```

<HTML>
<HEAD>
<!--
----->
</HEAD>
<BODY>
<CENTER>
<B>「釣り銭の用意の実験」</B>
<BR><BR>
<APPLET CODE="Turisen2.class" WIDTH="300" HEIGHT="250"></APPLET>
<BR><BR>
</CENTER>
</BODY>
</HTML>

```

3. Javaアプリケーション・プログラムリスト

```

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

//  

//      「F釣り銭の用意の実験」  

//      Copyright (C) K.Niwa 2002.08.18  

//      (Java 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 FTurisen2 extends Frame implements Runnable{ //スレッドを使えるようにする

    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; //イメージ型で宣言する
    Image[] img; //メディアトラッカー型で宣言する
    MediaTracker myMt;
    int flag=0; //乱数
    double r; //実験回数
    int ct1; //時間かせぎ
    long t; //釣り銭の個数
    int d=0; //コインの裏表の点滅ループカウンター
    int ct; //現時点での釣り銭の最高必要個数
    int min=0; //速度
    int N=1000;

    *****
    フレームとイベントの定義 *****
    public FTurisen2() {
        setSize(300+10,360); //フレームの大きさ
        addWindowListener(new WindowAdapter() { //閉じるボタンイベント処理
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
        myTh=null;
        if (myTh==null) { //スレッドの初期化
            myTh=new Thread(this); //スレッドの実体化
            myTh.start(); //スレッドの開始
        }
        //setBackground(Color.magenta); //背景色の設定
        int count; //ループカウンター
        imgNo=0; //イメージの実体化
        img=new Image[2];
        img[0]=Toolkit.getDefaultToolkit().getImage("coin1.jpg"); //コイン表画像の読み込み
        img[1]=Toolkit.getDefaultToolkit().getImage("coin2.jpg"); //コイン裏画像の読み込み
        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にイメージの貼り付け
        }
        try {
            myMt.waitForID(0); //例外処理
        }
        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();           //パネルの作成
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); //初期化ボタン

myBtn[0].addActionListener(new ActionListener() { //初期化ボタン
    public void actionPerformed(ActionEvent e) {
        flag=0;
        r=0.1;
        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 void init()

***** public void run() メソッド *****<イベントなしでも処理が行われる>*****
public void run() {
    while (true) {
        try{ //例外処理
            myTh.sleep(N);
        }
        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
            ct1=0; //実験回数
            d=0; //釣り銭の個数の初期化
    }
}

```

```

min=0; //現時点での釣り銭の最高必要個数の初期化
g.drawString("実験回数="+ct1,100,20); //実験回数の表示
g.drawImage(img[0],135,30,this); //コインの表示
g.drawString("コイン",135,75);
g.drawString("釣り銭(500円玉)の個数="+d,80,120-10);
g.drawString("現時点での釣り銭(500円玉)の最高必要個数="+
(-1*min),20,150-10);
g.drawString("Copyright(C) K.Niwa 2002.08",80,320); //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.drawImage(img[0],135,30,this); //コインの表示
    g.drawString("コイン",135,75);
    g.drawString("釣り銭(500円玉)の個数="+d,80,120-10);

    g.drawString("現時点での釣り銭(500円玉)の最高必要個数="+
(-1*min),20,150-10);

    g.drawString("実験回数="+ct1,100,20); //実験回数の表示
    //作者表示
    g.drawString("Copyright(C) K.Niwa 2002.08",80,320);

    //判断前のコインの裏表の点滅
    for (ct=1;ct<=3;ct++) {
        g.drawImage(img[0],135,30,this);
        for (t=1;t<=3000000;t++) {}
        g.drawImage(img[1],135,30,this);
        for (t=1;t<=3000000;t++) {}
    }

    //コインの表裏の判断
    r=Math.random();
    if (r<0.5) {
        g.drawImage(img[0],135,30,this); //表が出た
        d=d+1;
    }
    else {
        g.drawImage(img[1],135,30,this); //裏が出た
        d=d-1;
    }

    g.clearRect(0,100,300,60); //部分消去
    g.drawString("釣り銭(500円玉)の個数="+d,80,120-10);

    //現時点での釣り銭(500円玉)の最高必要個数の判断
    if (d<=min) {
        min=d;
    }

    g.drawString("現時点での釣り銭(500円玉)の最高必要個数="+
(-1*min),20,150-10);

    g.drawString("実験回数="+ct1,100,20); //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) {
    g.drawImage(img[0],135,30,this); //コインの表示
    g.drawString("コイン",135,75);
}

```

```

        g.drawString("釣り銭(500円玉)の個数="+d,80,120-10);
        g.drawString("現時点での釣り銭(500円玉)の最高必要個数="+(-1*min),20,150-10);
        g.drawString("実験回数="+ct1,100,20); //実験回数の表示
//作者表示
        g.drawString("Copyright(C) K.Niwa 2002.08",80,320);

//コインの表示
        if (r<0.5) {
            g.drawImage(img[0],135,30,this); //表が出た
        } else {
            g.drawImage(img[1],135,30,this); //裏が出た
        }

        g.clearRect(0,100,300,60); //部分消去

        g.drawString("釣り銭(500円玉)の個数="+d,80,120-10);
//現時点での釣り銭(500円玉)の最高必要個数の判断
        if (d<=min) {
            min=d;
        }

        g.drawString("現時点での釣り銭(500円玉)の最高必要個数="+(-1*min),20,150-10);
        g.drawString("実験回数="+ct1,100,20); //実験回数の表示

    } //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 FTurisen2();
    w.show();
} //public static void main(String[] args)

} //public class FTurisen2 extends Frame implements Runnable

```