



```

import java.lang.Math;

public class Montecal2 extends Applet {
    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 flag;                //変数宣言
    double x;
    int N;                    //ゴマの個数
    int count;               //ループカウンター

    /****public void init()   init メソッド (void は値を返さないメソッドの意) *****/
    public void init () {
        setBackground (Color.lightGray);           //背景色をグレーにする
        myBtn=new Button [4];                      //ボタンの実体化
        myBtn [0]=new Button ("初期化");
        myBtn [1]=new Button ("自動 500 個");
        myBtn [2]=new Button ("自動 5000 個");
        myBtn [3]=new Button ("自動 50000 個");

        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++) {          //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

        //初期化ボタンを押したときのイベント処理
        myBtn [0].addActionListener (new ActionListener () {
            public void actionPerformed (ActionEvent e) {
                flag=0;                            //ボタン識別子
                repaint ();
            }
        });

        //自動 500 個ボタンを押したときのイベント処理
        myBtn [1].addActionListener (new ActionListener () {
            public void actionPerformed (ActionEvent e) {
                flag=1;                            //ボタン識別子
                N=500;
                repaint ();
            }
        });

        //自動 5000 個ボタンを押したときのイベント処理
        myBtn [2].addActionListener (new ActionListener () {
            public void actionPerformed (ActionEvent e) {
                flag=2;                            //ボタン識別子
                N=5000;
                repaint ();
            }
        });

        //自動 50000 個ボタンを押したときのイベント処理
        myBtn [3].addActionListener (new ActionListener () {
            public void actionPerformed (ActionEvent e) {
                flag=3;                            //ボタン識別子
                N=50000;
                repaint ();
            }
        });

    } //public void init ()

```

```

//*****paint メソッド*****
public void paint(Graphics g) {
    if (flag==0) {
        g.clearRect(0,0,300,300);
        g.drawRect(0,0,300,300);
        g.drawString("Copyright (C) K.Niwa 2001.11",10,20); //作者名
    }
    else if (flag==1 || flag==2 || flag==3) {
        double x,y,pai,t; //変数宣言
        int i,sum=0,px,py; //変数宣言と変数初期化
        g.clearRect(0,0,300,300);
        g.drawRect(0,0,300,300);

        for (i=1;i<=N;i++) {
            for (t=0;t<1000;t++) {} //時間稼ぎ
            x=Math.random();
            y=Math.random();
            if (x*x+y*y<1) { //円内に入ったら
                px=(int) (300*x);
                py=(int) (300*y);
                g.setColor(Color.red);
                g.drawLine(px,py,px,py); //点を打つ
                sum++;
            }
            else if (x*x+y*y>1) {
                px=(int) (300*x);
                py=(int) (300*y);
                g.setColor(Color.white);
                g.drawLine(px,py,px,py); //点を打つ
            }
        }
        pai=(double) 4*sum/N;
        Font f=new Font((g.getFont()).getName(),Font.BOLD,20);
        g.setFont(f);
        g.clearRect(0,305,300,55);
        g.setColor(Color.black);
        g.drawString(sum+" / "+i+" 個",80,325);
    }
    g.setColor(Color.red);
    pai=(double) 4*sum/N;
    g.drawString("π="+pai,170,280);
}
}
}

```

## (2) HTML リスト

```

<HTML>
  <HEAD>
< !-----
                                ゴマをまいて円周率πを求める
                                (モンテカルロ法によるπの近似)
                                Copyright (C) K.Niwa 2001.11. 6
----->
  </HEAD>
<BODY>
  <CENTER>
    <B>ゴマをまいて円周率πを求める!!</B>
    <BR>
    (モンテカルロ法によるπの近似)
    <BR><BR>
    <APPLET CODE="Montecal2.class" WIDTH="350" HEIGHT="360"></APPLET>
    <BR><BR>
    ※ ゴマを 50,000 個まいて、πの近似値を求めています...

```

```
</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モンテカルロ法によるπの近似」
//                      (ゴマをまいて、円周率πが求まる!!)
//                      Copyright (C) K.Niwa 2002.08.24
//                      (J a v a アプリケーション)
//
////////////////////////////////////

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

public class FMontecal2 extends Frame {           // Frame クラスを継承する

    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 flag;                                     // 変数宣言
    double x;
    int N;                                        // ゴマの個数
    int count;                                   // ループカウンタ

    /*** フレームとイベントの定義 ****
    public FMontecal2() {

        setBackground(Color.lightGray);         // 背景色を黒にする
        setSize(300+20,360);                    // フレームの大きさ

        addWindowListener(new WindowAdapter() { // 閉じるボタンのイベント
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });

        myBtn=new Button[4];                     // ボタンの実体化
        myBtn[0]=new Button("初期化");
        myBtn[1]=new Button("自動 500 個");
        myBtn[2]=new Button("自動 5000 個");
        myBtn[3]=new Button("自動 50000 個");

        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

        // 初期化ボタンを押したときのイベント処理
        myBtn[0].addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                flag=0;                           // ボタンの識別子
                repaint();
            }
        });
    }
};
```

```

//自動 500 個ボタンを押したときのイベント処理
myBtn[1].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        flag=1;           //ボタンの識別子
        N=500;
        repaint();
    }
});

//自動 5000 個ボタンを押したときのイベント処理
myBtn[2].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        flag=2;           //ボタンの識別子
        N=5000;
        repaint();
    }
});

//自動 50000 個ボタンを押したときのイベント処理
myBtn[3].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        flag=3;           //ボタンの識別子
        N=50000;
        repaint();
    }
});

} //public FMontecal2()

*****paint メソッド*****
public void paint(Graphics g) {
    if (flag==0) {
        g.clearRect(0,0,300,300);
        g.drawRect(0,0,300,300);
        g.drawString("Copyright (C) K.Niwa 2001.11",10,20); //作者名
    }
    else if (flag==1 || flag==2 || flag==3) {
        double x,y,pai,t;           //変数宣言
        int i,sum=0,px,py;         //変数宣言と変数初期化
        g.clearRect(0,0,300,300);
        g.drawRect(0,0,300,300);

        for (i=1;i<=N;i++) {
            for (t=0;t<1000;t++) {} //時間稼ぎ
            x=Math.random();
            y=Math.random();

            if (x*x+y*y<1) { //円内に入ったら
                px=(int) (300*x);
                py=(int) (300*y);
                g.setColor(Color.red);
                g.drawLine(px,py,px,py); //点を打つ
                sum++;
            }
            else if (x*x+y*y>1) {
                px=(int) (300*x);
                py=(int) (300*y);
                g.setColor(Color.white);
                g.drawLine(px,py,px,py); //点を打つ
            }
        }

        pai=(double) 4*sum/N;
        Font f=new Font((g.getFont()).getName(),Font.BOLD,20);
        g.setFont(f);
    }
}

```

```

        g.clearRect(0,305,300,55);
        g.setColor(Color.black);
        g.drawString(sum+" / "+i+" 個",80,325);
    }

    g.setColor(Color.red);
    pai=(double)4*sum/N;
    g.drawString("π="+pai,170,280);
}
} //public void paint(Graphics g)

/***** public static void main ｽﾞｯｯｯ *****/
public static void main(String[] args) {
    Frame w=new FMontecal2();
    w.show();
} //public static void main(String[] args)

} //public class FMontecal2 extends Frame

```