

補足iとjで使ったJava Appletを掲載します。

下記は確認した動作環境です。

OS : Windows7, 32bitまたは64bit

Java : JDK8

Eclipse : Luna 4.4.2

1) 補足 i : 素数による画素分散

```
import java.applet.Applet;
import java.awt.*;
import java.awt.image.*;
import java.awt.Graphics;

public class randomtest1 extends Applet{
    private static final long serialVersionUID=1L;

    int ix,iy,in,k,i,j;
    int pix[];
    int new_pix[];
    int red,green,blue;
    int iw,ih;
    int w=0,h=0,wh,n=0,nmax;
    double tau=(1.0+Math.sqrt(5.0))/2.0;
    double fai,theta,x,pi2=2.0*Math.PI;

    int gn1,gn2;

    PixelGrabber pixelG;
    Image img;
    Image new_img;

    public void init(){

        img=getImage(getCodeBase(), "test11.png");
        MediaTracker mt=new MediaTracker(this);
        mt.addImage(img,0);
        try{
            mt.waitForID(0);
        } catch (InterruptedException e){}

        w=img.getWidth(this);
        h=img.getHeight(this);
        wh=w*h;
        nmax=wh;
        //
        gn2=nmax;
        gn1=95483;
        iw=2;
        ih=iw;

        pix=new int[w*h];
        new_pix=new int[w*h];

        pixelG=new PixelGrabber(img,0,0,w,h,pix,0,w);
        try{
            pixelG.grabPixels();
        } catch(InterruptedException e) {}

        for (i=0; i<=wh-1; i++){
            new_pix[i]=pix[i];
        }
    }
}
```

```

    }
    new_img=createImage(new MemoryImageSource(w,h,new_pix,0,w));

}

public void paint(Graphics g){
    System.out.println ("w="+w+" h="+h+"nmax="+nmax);
    System.out.println ("gn1="+gn1+" gn2="+gn2);

    for (i=0;i<=(nmax-1)*0.2;i++){

        n=(n+gn1)% gn2;
        ix=(int)(n*w);
        iy=(int)(n/w);
        //
        //System.out.println ("n="+n+" ix="+ix+" iy="+iy);
        //if (i>wh){
        //    break;
        //}
        int red=(pix[n]>>16)&0xff;
        int green=(pix[n]>>8)&0xff;
        int blue=(pix[n]&0xff);
        g.setColor(new Color(red,green,blue));
        g.fillArc(ix,iy,iw,ih,0,360);

    }
}
}

```

2) 補足 j : フィボナッチ格子を使った数値積分
(変更した二次元の Java Applet プログラム)

```

package integral;

import java.applet.Applet;
import java.awt.*;
import java.awt.image.PixelGrabber;
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.PrintWriter;

//integral
public class sglpf extends Applet{
    private static final long serialVersionUID=1L;

    int i,ii,j,m,k,lm;
    int md=2,nv=0;
    int nt=0,n[]={13,21,34,55,89,144,233,377,610,987,1597,2584};
    int lg[]={8,13,21,34,55,89,144,233,377,610,987,1597};
    int lgt[]=new int[4];
    int il[]={0,12,12,6};

    double f2;
    double eps=1/1000000;
    double esterr[]=new double[20];
    double err[]=new double[20];
    double ev=4.540419758842611;
    double x,y;
    double func;
    double v,dv, h,v1,v2,xv,xv5,t;
    double c[]={-252,1386,-3080,3465,-1980,462};

```

```

double z[]=new double[4];
double p[]=new double[4];
double w[]=new double[4];
double one=1,half=one/2;

```

```

PixelGrabber pixelG;
Image img;

```

```

public void init(){
    PrintWriter fout;
    System.out.println ("n[2]="+n[2]+" il[0]="+il[0]);
    v1=1;
    ii=0;

    //initialization
    for (i=0;i<il[1];i++){
        nv=n[i];
        nt=nt+nv-1;
        h=one/nv;
        v=0.0;
        for (m=1;m<md;m++){
            lgt[m]=0;
        }

        //compute absissae
        for (k=1;k<=nv-1;k++){
            lm=1;
            z[0]=h*k;
            for(m=1;m<md;m++){
                lgt[m]=(lgt[m]+lg[i])%nv;
                if(lgt[m]==0){
                    lm=0;
                }
                z[m]=h*lgt[m];
            }
            if (lm!=0){

                //transformation
                for (m=0;m<md;m++){
                    xv=z[m];
                    t=c[0];
                    for (j=1;j<6;j++){
                        t=t*xv+c[j];
                    }
                    xv5=xv*xv*xv*xv*xv;
                    p[m]=xv5*xv*t;
                    w[m]=2772*xv5*(1-xv)*(1-xv)*(1-xv)*(1-xv)*(1-xv);
                }

                //evaluation
                if(md==2){
                    dv=(Math.exp(p[0]*p[1])/Math.sqrt(p[0]*p[1]))*w[0]*w[1];
                }
                v=v+dv;
            }
        }
        v2=h*v;

        //convergence test

```

```

    esterr[ii]=Math.abs(v1-v2);
    err[ii]=Math.abs(ev-v2);
    System.out.println (" esterr="+esterr[ii]);
    v1=v2;
    //if (esterr[ii]-eps<0.000001){
    //    System.out.println (" esterr="+esterr[ii]);
    //    //converged
    //    v=v1;
    //    break;
    //}
    ii++;
}
if (esterr[ii]-eps>=0.000001){
    nt=-nt;
    v=v1;
}

//err=v-ev;
System.out.println ("md="+md+" nt="+nt+" v="+v+" eps="+eps+" err="+err+" esterr="+esterr[ii]);

try{
    fout=new PrintWriter(new BufferedWriter(
        new FileWriter("test.txt")));

    for (i=0;i<=ii;i++){
        fout.println(esterr[i]);
    }
    for (i=0;i<=ii;i++){
        fout.println(err[i]);
    }
    fout.close();
}
catch (Exception e) {
    System.out.println("err: "+e);System.exit(1);
}

}

}

```