10章，13章，14章で使ったJava Appletを掲載します。
下記は確認した動作環境です。
　OS：Windows7, Mac OS X 10.9.5, 32bitまたは64bit
　Java：JDK8
　Eclipse：Luna 4.4.2

1）10章：画像表示への応用

サンプリング（任意の元画像から10000点をサンプリングして表示する場合）

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

public class sample1 extends Applet{
	private static final long serialVersionUID=1L;

	int ix,iy,in,k,i,j;
	int pix[];
	int red,green,blue;
	int iw,ih;
	int w=0,h=0,wh,n,nmax=10000;
	double ps;
	double tau=(1.0+Math.sqrt(5))/2.0;
	double fai,theta,x,pi2=2.0*Math.PI;

	PixelGrabber pixelG;
	Image img;
	Image new_img;

	public void init(){

		img=getImage(getCodeBase(), "test1.jpg");

		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;
		iw=6;
		ih=iw;

		pix=new int[wh];

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

	}

	public void paint(Graphics g){
		System.out.println ("w="+w+"	h="+h);
		for (n=0;n<=nmax;n++){
			fai=pi2*(1.0-1.0/tau);
			theta=n*fai;
```

```
                        x=theta%pi2;
                        ix=(int)((w*x)/pi2);
                        iy=(int)(h*n*2/nmax);
                        in=iy*w+ix;

                        if (in>wh){
                                break;
                        }
                        int red=(pix[in]>>16)&0xff;
                        int green=(pix[in]>>8)&0xff;
                        int blue=(pix[in]&0xff);
                        g.setColor(new Color(red,green,blue));
                        g.fillArc(ix,iy,iw,ih, 0, 360);
                }
        }
}
```

2）13章：素数による画素分散（700×700画素数の元画像を素数187163で分散させる場合）

```
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(), "Fukaya700_700.jpg");
                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=187163;
                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);
                        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.fillRect(ix,iy,iw,ih);
                }
        }
}
```

3）14 章：フィボナッチ格子を使った数値積分

（二次元に変更した Java Applet プログラム）

```
import java.applet.Applet;
import java.awt.*;
import java.awt.Graphics;
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.PrintWriter;

//integral
public class sglpg3 extends Applet{
        private static final long serialVersionUID=1L;

        int i,ii,j,m,k,lm,iw=8,ih=8,ix,iy;
        int md=2,nv=0;
        int nt=0,n[]={20,30,50,70,100,200,300,500,700,1000,2000,3000};
        int lg[]={12,19,31,43,62,124,185,309,433,618,1236,1854};
        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;

public void paint(Graphics g){
        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);
                    g.setColor(Color.red);
                    ix=(int)(100*Math.log10(n[ii]));
                    iy=(int)(-50*Math.log10(err[ii]));
                    g.fillArc(ix,iy+100,iw,ih,0,360);
                    System.out.println (" esterr="+esterr[ii]);
                    v1=v2;
                    ii++;
                }
                if (esterr[ii]-eps>=0.000001){
                    nt=-nt;
                    v=v1;
                }

                //err=v-ev;
                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);
        }
        }
}
```