Clock.java

```
import java.applet.*;
import java.awt.*;
import java.util.*;

public class Clock extends Applet implements Runnable {
    Thread t;
    Image dbi, cim;
    Graphics dbg;
    Color bgcolor, arrowscolor, textcolor;   //culoarea fundalului, a acelor si a textului
    Font f = new Font("Helvetica", 0, 9);
    Date d;            //data
    int delay=100;     //intarzierea de refresh
    int w, h;          //dimensiunile apletului
    int hh, mm, ss;    //ora, minutul, secunda
    int rs=10, rm=40, rh=30; //lungimile secundarului si acelor cronometrului, minutarului, orarului
    //(x0,y0) coord centrului ceasului, (x1,y1) coord varfului unui ac
    //(x2,y2) centrul secundarului, (x3,y3) centrul cronometrului
    //(x4,y4) unde este afisata ora digitala, (x5,y5) unde este afisat AM/PM
    int x0=62, y0=91, x1, y1, x2=62, y2=111, x3=62, y3=70, x4=72, y4=98, x5=80, y5=86;
    double fim, fih, fis;     //unghiul minutarului, orarului, secundarului
    //double PI2 = 6.2831853071795862D; //2*Math.PI
    double PI2 = 2*Math.PI;
    double fist, fimt; //fist foloseste la afisarea secundarului cronometrului
                       //fimt foloseste la afisarea minutarului cronometrului
    long t0;           //stocheaza timpul la care este incarcat apletul si foloseste la cronometru
    int d1, d2, d3, d4, d5, d6;       //folosesc la afisarea digitala a orei

    public void init(){
        w = size().width;
        h = size().height;
        t0 = (new Date()).getTime();  //nr de milisecunde de la 1 ianuarie 1970 00:00:00 GMT
        dbi = createImage(w, h);
        dbg = dbi.getGraphics();
        cim = getImage(getCodeBase(), "clock.gif");
        bgcolor = parsecolor("BGCOLOR", 0);
        textcolor = parsecolor("TEXTCOLOR", 0);
        arrowscolor = parsecolor("ARROWSCOLOR", 0);
    }

    Color parsecolor(String s, int i){
        String s1 = getParameter(s);
        Color color = new Color(i);
        Color color1;
        try{color1 = new Color(Integer.parseInt(s1, 16));}
        catch(Exception e){return color;}
        return color1;
    }

    public void start(){if(t == null) {t = new Thread(this); t.start();}}
    public void stop(){if(t != null) {t.stop(); t = null;}}

    public void run(){
        while(true) {
            repaint();
            try{Thread.sleep(delay);}
            catch(Exception e){return;}
        }
    }

    public void update(Graphics g){

       //obtinerea orei, minutului, secundei
        Date date = new Date();
        hh = date.getHours();
        mm = date.getMinutes();
        ss = date.getSeconds();

        fis = (PI2 * (double)ss) / 60D;
        fim = (PI2 * (double)mm + fis) / 60D;
        fih = (PI2 * (double)(hh % 12) + fim) / 12D;

        //desenarea ceasului
        dbg.setColor(bgcolor);
        dbg.fillRect(0, 0, w, h);
        dbg.drawImage(cim, 0, 0, this);

        //AM sau PM
```

```
        String s = "AM";
        if(hh >= 12)s = "PM";
        hh = hh % 12;
        if(hh == 0)hh = 12;

        //afsajul digital
        dbg.setColor(textcolor);
        d1 = hh / 10;
        d2 = hh % 10;
        d3 = mm / 10;
        d4 = mm % 10;
        dbg.setFont(f);
        //dbg.drawString(hh + ":" + mm, x4, y4); //ora
        dbg.drawString(d1 + "" + d2 + ":" + d3 + "" + d4, x4, y4); //ora in formatul hh:mm
        dbg.drawString(s, x5, y5);  //AM sau PM

        //desenarea acelor

        //secundarul
        x1 = (int)((double)x2 + (double)rs * Math.sin(fis));
        y1 = (int)(((double)y2 - (double)rs * Math.cos(fis)));
        dbg.setColor(arrowscolor);
        dbg.drawLine(x2, y2, x1, y1);

       //cronometrul
        long l = date.getTime() - t0;
        fist = (PI2 * (double)l) / 60000D;
        fimt = (PI2 * (double)l) / 3600000D;
        x1 = (int)((double)x3 + (double)rs * Math.sin(fist));
        y1 = (int)(((double)y3 - (double)rs * Math.cos(fist)));
        dbg.drawLine(x3, y3, x1, y1); //secundarul
        x1 = (int)((double)x3 + (double)rs * 0.8D * Math.sin(fimt));
        y1 = (int)(((double)y3 - (double)rs * 0.8D * Math.cos(fimt)));
        dbg.drawLine(x3, y3, x1, y1); //minutarul

       //minutarul
        x1 = (int)((double)x0 + (double)rm * Math.sin(fim));
        y1 = (int)(((double)y0 - (double)rm * Math.cos(fim)));
        dbg.drawLine(x0, y0, x1, y1);

        //orarul
        x1 = (int)((double)x0 + (double)rh * Math.sin(fih));
        y1 = (int)(((double)y0 - (double)rh * Math.cos(fih)));
        dbg.drawLine(x0, y0, x1, y1);
        x1++;
        dbg.drawLine(x0, y0, x1, y1);
        y1++;
        dbg.drawLine(x0, y0, x1, y1);
        x1--;
        dbg.drawLine(x0, y0, x1, y1);

        //centrul ceasului
        dbg.fillOval(x0 - 2, y0 - 2, 4, 4);
        dbg.setColor(arrowscolor.darker());
        dbg.drawOval(x0 - 2, y0 - 2, 4, 4);

        //desenarea imaginii
        g.drawImage(dbi, 0, 0, this);
    }
}
```

Clock.htm

==========

```
<HTML>
<BODY BGCOLOR=#000000>
<BR><BR>
<CENTER>
<APPLET CODE="Clock.class" WIDTH=130 HEIGHT=193>
<PARAM NAME="BGCOLOR"     VALUE="000000">
<PARAM NAME="TEXTCOLOR"     VALUE="000000">
<PARAM NAME="ARROWSCOLOR"     VALUE="00000">
</APPLET>
</BODY>
</HTML>
```