

CLASE FIGURA

```
package Practica7.Dominio;
import java.awt.Color;

public abstract class Figura
{
    //ATRIBUTOS DE INSTANCIA

    protected int PosX;
    protected int PosY;
    protected Color Col;
    protected boolean Relleno = false;
    protected boolean visible = true;    //Las pintaré si son visibles

    //ATRIBUTOS DE CLASE

    //MÉTODOS DE INSTANCIA

    void SetPosX(int X)
    {
        if(X>0 && X<800)
            this.PosX = X;
    }

    public int getX()
    {
        return PosX;
    }

    void SetPosY(int Y)
    {
        if(Y>0 && Y<600)
            this.PosY = Y;
    }

    public int getY()
    {
        return PosY;
    }

    void setRelleno(boolean relleno)
    {
        this.Relleno = relleno;
    }
}
```

```

    public boolean isRelleno()
    {
        return Relleno;
    }

    void setColor(Color C)
    {
        this.Col = C;
    }

    public Color getColor()
    {
        return Col;
    }

    public String getInfo()
    {
        return "Posicion X: " + this.getX() + "\nPosicion Y: " +
this.getY() + "\nColor: " + this.getColor() + "\nRelleno? =>" +
this.isRelleno();
    }

    public void pintar(java.awt.Graphics g)
    {
        g.setColor(this.Col);
    }

    public void setVisible(boolean visible)
    {
        this.visible = visible;
    }

    public boolean isVisible()
    {
        return visible;
    }

    Figura(int X, int Y, boolean R, Color C)
    {
        this.SetPosX(X);
        this.SetPosY(Y);
        this.setRelleno(R);
        this.setColor(C);
    }
}

```

CLASE CUADRADO

```
package Practica7.Dominio;
import java.awt.Color;
import Practica7.Dominio.Figura;

public class Cuadrado extends Figura
{
    //ATRIBUTOS DE INSTANCIA

    private int Lado;

    //MÉTODOS DE INSTANCIA SE HEREDAN DE FIGURA EXCEPTO LOS DE LADO

    void setLado(int L)
    {
        if(L<600 && L>1)
            this.Lado = L;
        else
            this.Lado = 200;
    }

    int getLado()
    {
        return Lado;
    }

    @Override
    public String getInfo() //IMPORTANTE HACERLO PUBLIC
    {
        return "Clase: " + this.getClass() + "\n" + super.getInfo()
+ "\nLado: " + this.getLado();
    }

    @Override
    public void pintar(java.awt.Graphics g)
    {
        //g.setColor(this.getColor()); LO HAGO EN LA PADRE
        super.pintar(g);
        if(this.isRelleno())
            g.fillRect(this.getX(), this.getY(), this.getLado(),
this.getLado());
    }
}
```

```

        else
            g.drawRect(this.getX(), this.getY(), this.getLado(),
this.getLado());
    }

    //CONSTRUCTOR

    public Cuadrado(int X, int Y, boolean R, Color C, int Lado)
    {
        super(X,Y,R,C);
        this.setLado(Lado);
    }
}

```

CLASE CIRCULO

```

package Practica7.Dominio;
import java.awt.Color;
import Practica7.Dominio.Figura;
import java.awt.*;
import javax.swing.JFrame;

public class Circulo extends Figura
{
    private int Radio;

    void setRadio(int Rad)
    {
        if(Rad>1 && Rad<600)
            this.Radio = Rad;
        else
            this.Radio = 100;
    }

    int getRadio()
    {
        return Radio;
    }

    @Override
    public String getInfo() //IMPORTANTE HACERLO PUBLIC

```

```

    {
        return "Clase: " + this.getClass() + "\n" + super.getInfo()
+ "\nRadio: " + this.getRadio();
    }

    @Override
    public void pintar(java.awt.Graphics g)
    {
        //g.setColor(this.getColor()); LO HAGO EN LA PADRE
        super.pintar(g);
        if(this.isRelleno())
            g.fillOval(this.getX(), this.getY(), this.getRadio(),
this.getRadio());
        else
            g.drawOval(this.getX(), this.getY(), this.getRadio(),
this.getRadio());
    }

    public Circulo(int X, int Y, boolean R, Color C, int Rad)
    {
        super(X,Y,R,C);
        this.setRadio(Rad);
    }
}

```

CLASE DIBUJO

```

package Practica7.Dominio;
import Practica7.Dominio.Figura;
import Practica7.Dominio.Lienzo;
import java.awt.*;
import javax.swing.JFrame;

public class Dibujo extends JFrame
{
    Lienzo lienzo;

    public Dibujo()
    {
        super("Dibujo");
        lienzo = new Lienzo();
        lienzo.setSize(800,600);
    }
}

```

```

        this.add(lienzo);
        this.pack();
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }

    /**
     * Pinta el figura recibido por el App y actualiza el lienzo
     * (canvas)
     * @param figura figura a pintar
     */
    public void pintar(String grupo, Figura figura)
    {
        lienzo.addFigura(grupo, figura);
        lienzo.repaint();
    }

    public void ocultarGrupo(String grupo)
    {
        lienzo.visualizarGrupo(grupo, false);
        lienzo.repaint();
    }

    public void mostrarGrupo(String grupo)
    {
        lienzo.visualizarGrupo(grupo, true);
        lienzo.repaint();
    }

    public void ocultarTodasFiguras()
    {
        lienzo.visualizarTodo(false);
        lienzo.repaint();
    }

    public void mostrarTodasFiguras()
    {
        lienzo.visualizarTodo(true);
        lienzo.repaint();
    }
}

```

CLASE LIENZO

```
package Practica7.Dominio;
import Practica7.Dominio.Figura;
import java.awt.*;
import javax.swing.JFrame;
import java.util.*;

public class Lienzo extends Canvas
{
    HashMap<String,ArrayList<Figura>> grupos = new
    HashMap<String,ArrayList<Figura>>(); //<Color,Colection> => Todos Los
values se upcastean a object??

    public void addFigura(String grupo, Figura figura)
    {
        Set<String> claves = grupos.keySet(); //Los elementos de
keySet deben ser Strings no? (Un set de strings)

        if(claves.contains(grupo))
        {
            ArrayList<Figura> lista = grupos.get(grupo); //Por
que no me deja con ArrayList<Figura> lista = grupos.get(grupo); ??
            lista.add(figura);
            grupos.put(grupo, lista);
        }
        else
        {
            ArrayList<Figura> lista = new ArrayList<Figura>();
            lista.add(figura);
            grupos.put(grupo, lista);
        }
    }

    public void visualizarGrupo(String grupo, boolean visible)
    {
        ArrayList<Figura> figuras = grupos.get(grupo);

        for(Figura figura:figuras)
            figura.setVisible(visible);
    }

    /* AVERIGUAR POR QUÉ TIPIFICA MAL

```

```

    public void visualizarTodo(boolean visible)
    {
        Collection<ArrayList<Figura>> arreglos = grupos.values();

        for(ArrayList arreglo:arreglos)
            for(Figura figura:arreglo)
                figura.setVisible(visible);

    }*/

    public void visualizarTodo(boolean visible)
    {
        Iterator it = grupos.values().iterator();
        while(it.hasNext())
        {
            Iterator itFigurasGrupo = ((Collection)
it.next()).iterator();
            while(itFigurasGrupo.hasNext())
                ((Figura)
itFigurasGrupo.next()).setVisible(visible);
        }
    }

    /* AVERIGUAR POR QUÉ TIPIFICA MAL
    public void paint(Graphics g)
    {
        Collection<ArrayList<Figura>> arreglos = grupos.values();

        for(ArrayList arreglo:arreglos)
            for(Figura figura:arreglo)
                if(figura.isVisible())
                    figura.pintar(g);

    }*/

    public void paint(Graphics g)
    {
        Iterator it = grupos.values().iterator();
        while(it.hasNext())
        {
            Iterator itFigurasGrupo = ((Collection)
it.next()).iterator();
            while(itFigurasGrupo.hasNext())
            {
                Figura figura = (Figura) itFigurasGrupo.next();

```



```

        if(figura.isVisible())
            figura.pintar(g);
    }
}
}

```

CLASE UTIL

```

package Practica7.Dominio;

public class Util
{
    /**
     * Detiene el programa el tiempo especificado
     * @param segundos número de segundos a esperar
     */
    public static void wait(int segundos)
    {
        try
        {
            Thread.sleep(segundos*1000);
        }
        catch(Exception e)
        {
        }
    }
}

```

CLASE APPDIBUJOGRUPO

```

package Practica7.UI;

import Practica7.Dominio.Circulo;
import Practica7.Dominio.Cuadrado;

import Practica7.Dominio.Dibujo;

import Practica7.Dominio.Util;

```

```

import java.awt.Color;

public class AppDibujoGrupo
{
    public static void main(String[] args)
    {
        Dibujo dibujo = new Dibujo(); //Inicializo el dibujo

        //con dibujo.pintar(figura) meto directamente la figura en
el mapa

        //Circulo(int X, int Y, boolean R, Color C, int Rad)
        //Cuadrado(int X, int Y, boolean R, Color C, int Lado)

        dibujo.pintar("grupo verde", new
Circulo(10,10,true,Color.GREEN,80)); //NO OLVIDARSE EL Color. PARA
LOS COLORES!!!!
        dibujo.pintar("grupo verde", new
Circulo(100,10,true,Color.GREEN,50));
        dibujo.pintar("grupo rojo", new
Cuadrado(10,100,false,Color.RED,80));
        dibujo.pintar("grupo rojo", new
Circulo(400,300,true,Color.RED,120));
        dibujo.pintar("grupo azul", new
Cuadrado(600,500,true,Color.BLUE,70));

        Util.wait(1);
        dibujo.mostrarTodasFiguras();

        Util.wait(1);
        dibujo.ocultarGrupo("grupo verde");

        Util.wait(1);
        dibujo.ocultarGrupo("grupo azul");

        Util.wait(1);
        dibujo.ocultarGrupo("grupo rojo");

        Util.wait(1);
        dibujo.mostrarTodasFiguras();
    }
}

```

OUTPUT (final):

