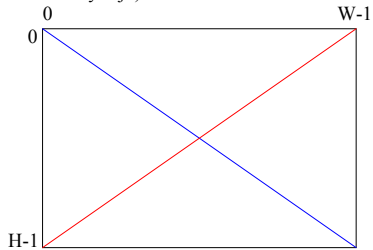
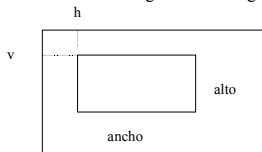


Problema 1. Dibujar las diagonales de la ventana (en colores azul y rojo)



```
import java.awt.*;
class Ventana extends Frame{
    static public void main(String[] args){
        Ventana v=new Ventana();//invoca ctor
        v.show();//show invoca a método paint
    }
    private static final int W=300, H=300;
    public Ventana(){
        setSize(W,H);
    }
    public void paint(Graphics x){
        x.setColor( Color.blue );
        x.drawLine(0,0,W-1,H-1);
        x.setColor( Color.red );
        x.drawLine(0,H-1,W-1,0);
    }
}
```

Problema 2. Dibujar el siguiente rectángulo:



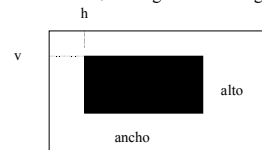
Solución 1. Con 4 líneas

```
x.drawLine(h, v, h+ancho, v);
x.drawLine(h+ancho, v, h+ancho, v+alto);
x.drawLine(h+ancho, v+alto, h, v+alto);
x.drawLine(h, v+alto, h, v);
```

Solución 2.

```
x.drawRect(h,v, ancho, alto);
```

Problema 3. Dibujar el siguiente rectángulo:



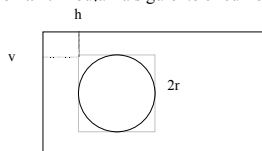
Solución 1.

- a) con alto líneas horizontales
- b) con ancho líneas verticales
- c) con alto/2 rectángulos

Solución 2.

```
x.fillRect(h,v, ancho, alto);
```

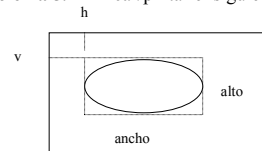
Problema 4. Dibujar la siguiente circunferencia:



Solución 1. Algoritmo que aplique ecuación $x^2 + y^2 = r^2$

Solución 2. `x.drawOval(h,v, 2*r, 2*r);`

Problema 5. Graficar/pintar el siguiente óvalo



Solución 1. Algoritmo ad-hoc

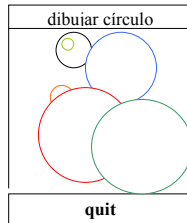
Solución 2. `x.drawOval(h,v, ancho, alto);`

¿Pintar?

```
x.fillOval(h,v, ancho, alto);
```

Clase 13: Dibujos y Gráficos

Problema. dibujar círculos de tamaño, ubicación y color aleatorios



```
import java.awt.*; import java.awt.event.*;
class Circulos
    extends Frame
    implements ActionListener
{
    static public void main(String[] args){
        new Circulos().show();
    }
    //componentes
    private Button
        circulo=new Button("dibujar círculo"),
        quit=new Button("quit");

    //tela de dibujo
    private Canvas cv=new Canvas();
    private final int W=300,H=300;//ancho y alto
```

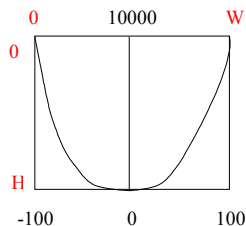
```
//constructor
public Circulos()
{
    //diagramar ventana
    setSize(W,H+2*25);//25 pixeles por borde
    setLayout(new BorderLayout());

    //agregar componentes
    add("North",circulo);
    add("South",quit);
    cv.setSize(W,H); //tamaño de la tela
    add("Center",cv);

    //activar escuchador
    circulo.addActionListener(this);
    quit.addActionListener(this);
}
```

```
//dibujar Circulo
public void actionPerformed(ActionEvent x){
    if(x.getSource()==quit) System.exit(0);
    //coordenadas del centro del círculo
    int h=U.azar(0,W-1), v=U.azar(0,H-1);
    //diámetro y radio
    int d=U.azar(0,Math.min(W,H)-1), r=d/2;
    //obtener objeto de clase Graphics
    Graphics g=cv.getGraphics();
    //establecer color (convención RGB)
    g.setColor(new Color(U.azar(0,255), //rojo
                        U.azar(0,255), //verde
                        U.azar(0,255))); //azul
    //dibujar círculo inscrito en rectángulo
    g.drawOval(h-r, v-r, d, d);
}
```

Problema. Graficar la función $y=x^2$ para x en $[-100,100]$.



Conversión de coordenadas

x	horizontal	y	vertical
-100	0	10000	0
0	W/2	0	H
100	W	10000	0
x	$W(x+100)/200$	$y=x^2$	$H - H y/10000$

Clase 13: Dibujos y Gráficos

```
public void paint(Graphics g){
    //dibujar ejes
    g.drawLine(0,H-1,W,H-1); //eje horizontal
    g.drawLine(W/2,0,W/2,H); //eje vertical
    //coordenadas comienzo primera línea
    int h0=0, v0=0;
    //iterar con valores de x (de 10 en 10)
    for(double x=-100; x<=100; x+=10){
        //calcular coordenadas de ventana
        double y=x*x; //valor de la función
        int h = (int)(W*(x+100)/200 + 0.5);
        int v = H - (int)(H*y/10000 + 0.5);
        //dibujar línea
        g.drawLine(h0,v0,h,v);
        //actualizar coordenadas comienzo línea
        h0=h; v0=v;
    }
}
```

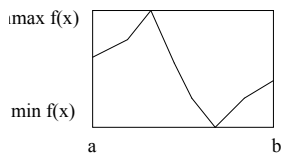
Generalización: método que permite de graficar cualquier función

```
static public void graficar(
    Funcion f,           //f: referencia a objeto
    int n,               //n° de puntos
    double a, double b, //extremos intervalo
    Graphics g,          //objeto gráfico
    int W, int H)        //ancho y alto de g

con

interface Funcion
{
    public double valor(double x);
}
```

Resultado. Grafico de función f en n puntos del intervalo $[a,b]$ en objeto gráfico g



```
static public void graficar(
    Funcion f,
    int n, double a, double b,
    Graphics g, int W, int H)
{
    //determinar menor y mayor valor de la funcion
    double
        min=f.valor(a), //valor de f(a)
        max=min,
        delta=(b-a)/(n-1); //distancia entre n puntos

    for(double x=a+delta; x<=b; x+=delta){
        double y = f.valor(x);
        min = Math.min(y,min);
        max = Math.max(y,max);
    }
}
```

```
//graficar funcion en objeto g de W*H pixeles
int h=0,
v=H-pixel(f.valor(a),min,max,H);
for(double x=a+delta; x<=b; x+=delta)
    g.drawLine(h, v,
        h=pixel(x,a,b,W),
        v=H-pixel(f.valor(x),min,max,H));
}
//convertir real x en [y,z] a entero en [0,w]
static public int pixel(
    double x, double y, double z, int w){
    return (int)Math.round(w*(x-y)/(z-y));
}
Nota. Math.round(x) es (double)(int)(x+0.5)
Ejemplos:
Math.round(5.5) entrega 6.0
Math.round(5.4) entrega 5.0
```

Ejemplo de uso. Graficar $y=x^2$ con 20 puntos del intervalo $[-100,100]$ en una ventana de 300 por 300

```
import java.awt.*;
class XX extends Frame{
    static public void main(String[]a){new XX().show();}
    private int W=300,H=300;
    public XX(){
        setSize(W,H);
    }
    public void paint(Graphics g){
        Cuadrado f=new Cuadrado();
        graficar(f,20,-100,100,g,W,H);
    }
    class Cuadrado implements Funcion{
        public double valor(double x){return x*x;}
    }
}
```

Clase 13: Dibujos y Gráficos

Ejercicio clase 13. Usar el método graficar para mostrar la función seno o coseno en $[-2\pi, 2\pi]$ en una ventana de 300x200 pixeles en la sgte forma:

				
Seno	Coseno	nº puntos?	100	quit

```
class Seno implements Funcion{
public double valor(double x){return Math.sin(x);}
}
class Coseno implements Funcion{
public double valor(double x){return Math.cos(x);}
}
```

```
import java.awt.*; import java.awt.event.*;
class GraficarSenoCoseno
extends Frame implements ActionListener
{
static public void main(String[]args)
new GraficarSenoCoseno().show();
}
private int W=300, H=200;
private Canvas cv=new Canvas();
private TextField puntos=new TextField("100");
private Label pregunta=new Label("nºpuntos?");
private Button
quit=new Button("quit"),
seno=new Button("seno"),
coseno=new Button("coseno");
```

```
public GraficarSenoCoseno(){
//diagramar panel para sur de ventana
Panel p=new Panel();
p.setLayout(new GridLayout(1,5));
p.add(seno); p.add(coseno);
p.add(pregunta); p.add(puntos); p.add(quit);

//diagramar ventana
setSize(W,H+25);
setLayout(new BorderLayout());
add("Center",cv); add("South",p);

//activar escuchadores
quit.addActionListener(this);
seno.addActionListener(this);
coseno.addActionListener(this);
puntos.addActionListener(this);//no en soluc.2
}
```

```
public void actionPerformed (ActionEvent x)
{
if(x.getSource()==quit) System.exit(0);

if(x.getSource()==puntos)
n=Integer.parseInt(puntos.getText());
return;
}
double b=2*Math.PI, a=-b;
Graphics g=cv.getGraphics();
if(x.getSource()==seno)
graficar(new Seno(),n,a,b,g,W,H);
else
graficar(new Coseno(),n,a,b,g,W,H);
}
private int n;
}
```

Solución 2. Sin escuchador para TextField puntos

```
public void actionPerformed (ActionEvent x){
if(x.getSource()==quit) System.exit(0);

int n=Integer.parseInt(puntos.getText());
double b=2*Math.PI, a=-b;

Graphics g=cv.getGraphics();
if(x.getSource()==seno)
graficar(new Seno(),n,a,b,g,W,H);
else
graficar(new Coseno(),n,a,b,g,W,H);
}
}
Nota.
Funcion f=x.getSource()==seno?new Seno():new Coseno();
graficar(f,n,a,b,g,W,H);
```

Propuestos

1. Implementar una calculadora con la siguiente interfaz:

7	8	9	/
4	5	6	*
1	2	3	-
0	.	=	+
signo -	clear	OFF	

2. implementar juego del gato en un cuadrilado de 3x3 botones (primer click dibuja X, sgte O, sgte X, ... hasta completar fila, columna o diagonal)

3. implementar puzzle

8	4	2
1		5
3	7	6

Click sobre número lo mueve a lugar vacío, hasta ordenarlos