

Primer Certamen

Todas las preguntas tienen igual puntaje.

1. Escriba una aplicación Java que muestre dos botones. Cuando presionamos el botón de la izquierda incrementa el número mostrado en el botón de la derecha. Cuando presionamos el botón de la derecha incrementa el número de la izquierda.



```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class BotonesCruzados extends JFrame
{
    public BotonesCruzados(String s)
    {
        super(s);
        botonA= new JButton("0");
        botonB= new JButton("0");
        botonA.addActionListener( new ActionListener()
        {
            public void actionPerformed(ActionEvent event)
            {
                String texto=botonB.getText();
                texto=Integer.toString(Integer.parseInt(texto)+1);
                botonB.setText(texto);
            }
        });
        botonB.addActionListener( new ActionListener()
        {
            public void actionPerformed(ActionEvent event)
            {
                String texto=botonA.getText();
                texto = Integer.toString(Integer.parseInt(texto)+1);
                botonA.setText(texto);
            }
        });
        JPanel panel= new JPanel();
        panel.add(botonA);
        panel.add(botonB);
        getContentPane().add(panel);
        pack();
    }

    public static void main (String[ ] args)
    {
        BotonesCruzados botonesFrame = new BotonesCruzados("Pregunta_1");
        botonesFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        botonesFrame.show();
    }
    private JButton botonA;
    private JButton botonB;
}
```

```
}

```

2. Considere la aplicación mostrada a continuación.
 - a) Haga un dibujo que muestre lo observado inmediatamente después de iniciar la ejecución de este programa.
 - b) Describa qué hace el programa.
 - c) Reescriba el programa sin hacer uso de clases internas ni clases anónimas.

```
import javax.swing.*;
import java.awt.event.*;
public class Boton extends JFrame
{
    private JButton boton;
    public Boton ()
    {
        boton = new JButton("Bien");
        boton.addActionListener(
            new ActionListener()
            {
                public void actionPerformed(ActionEvent event)
                {
                    boton.setText(boton.getText()+".");
                }
            }
        );
        getContentPane().add(boton);
        pack();
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
    public static void main(String[] args)
    {
        new Boton().show();
    }
}

```

a)



- b) El programa cambia el texto del botón agregando un punto cada vez que éste es presionado. El texto cambia: Inicial: "Bien", luego "Bien.", luego "Bien..", etc.

c)

```
import javax.swing.*;
import java.awt.event.*;
public class BotonV2 extends JFrame
{
    private JButton boton;
    public BotonV2 ()
    {
        boton = new JButton("Bien");
    }
}

```

```

    boton.addActionListener( new BotonActionListener(boton));
    getContentPane().add(boton);
    pack();
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
public static void main(String[ ] args)
{
    new BotonV2().show();
}
}
class BotonActionListener implements ActionListener
{
    private JButton boton;
    public BotonActionListener(JButton b)
    {
        boton = b;
    }
    public void actionPerformed(ActionEvent event)
    {
        boton.setText(boton.getText()+".");
    }
}
}

```

3. Supongamos que contamos con la clase Línea vista en clases. Ahora se pide crear la clase Trazo. Un trazo es una línea que además incluye un espesor. Además se desea que arreglos de trazos puedan ser ordenado según el espesor de cada trazo.
- Presente su solución para la clase pedida.
 - Suponga que en algún punto de algún código de programa, se tienen tres instancias de Trazo ya creados con nombres *a*, *b*, y *c*. Se pide crear un arreglo con esas tres instancias y luego ordenarlo según el espesor de cada trazo. ¿qué código usaría para cumplir con esto?
 - Luego se pide que usted implemente el método public boolean equals(Object otherObject). ¿Qué método(s) pediría usted se incluyera(n) en la clase Línea? ¿Cuál sería su implementación de este método en su clase para Trazo?
- a) Se dice que los trazos son Líneas con cierto espesor. Además debemos implementar la interfaz Comparable para poder usar los métodos de ordenamiento ya existen en el lenguaje.

La clase línea vista en clases es:

```

class Linea
{
    private Punto puntoi, puntof;
    public Linea ()
    {
        puntoi = new Punto();
        puntof = new Punto();
    }
    public Linea (Punto i, Punto f)
    {
        puntoi=i;
    }
}

```

```

        puntof=f;
    }
    public Linea(int x1, int y1, int x2, int y2)
    {
        puntoi = new Punto(x1,y1);
        puntof = new Punto(x2,y2);
    }
    public float pendiente()
    {
        return (float)(puntof.getY()-puntoi.getY())/
            (puntof.getX()-puntoi.getX());
    }
    public boolean esParalelaA (Linea L)
    {
        return pendiente()==L.pendiente();
    }
    public boolean esPerpendicularA(Linea L)
    {
        return pendiente()*L.pendiente()==-1;
    }
}

```

Luego la clase Trazo puede ser:

```

class Trazo extends Linea implements Comparable
{
    public Trazo (Punto i, Punto f, int esp)
    {
        super(i,f);
        espesor=esp;
    }

    public int compareTo (Object obj)
    {
        Trazo t = (Trazo) obj; //asumo obj != NULL
        return (espesor - t.espesor);
    }
    private int espesor;
}

```

b) EL código podría ser:

```

Trazo [] arreglo = new Trazo[3]; // creamos el arreglo
arreglo[0]=a; // ponemos los elementos;
arreglo[1]=b;
arreglo[2]=c;
Arrays.sort(arreglo);

```

c) i) Para implementación del método public boolean equals(Object otherObject) en clase Trazo se requiere que la clase Linea también tenga implementado el método equals.

ii) luego la implementación de equals en la clase Trazo puede ser:

```
public boolean equals(Object otherObject)
{
    if (!super.equals(otherObject)) return false;
    Trazo other = (Trazo)otherObject;
    return espesor == other.espesor;
}
```

4.

a) Alguien dice: *Si para compilar un programa se usara: %javac P4.java entonces para ejecutar el programa siempre se debe usar: %java P4*

¿Es esto correcto?, justifique.

No es correcto porque no siempre la ejecución será así. P4.java puede tener más de una clase definida. Si la clase que contiene el main no es P4 sino otra dentro del archivo, entonces la ejecución debe usar la otra clase.

b) Supongamos que creamos una aplicación Java con clase principal (aquella que contiene el método main) CloseableFrame, pero por error el programa es llamado ClosableFrame.java. ¿Qué comando debe usar para compilar el programa? ¿Qué comando debe usar usted para ejecutar el programa?

Para compilar: javac ClosableFrame.java

Para ejecutar : java CloseableFrame

c) Considere la siguiente clase para un punto:

```
class Punto
{
    private int x,y; // atributos únicos
    //....otros métodos
}
```

Proponga el prototipo y la implementación para el método *swap*, el cual permite intercambiar dos instancias de Punto pasados como argumentos.

La única forma es intercambiar los atributos de ambos objetos. En el fondo se intercambia el estado de cada objeto. Dentro de la clase Punto incluir el método:

```
public static void swap (Punto a, Punto b)
{
    int t = a.x;
    a.x=b.x;
    b.x=t;
    t=a.y;
    a.y=b.y;
    b.y=t;
}
```