

## UF5. [PAC01] Solución.

Para la realización de las siguientes actividades deberéis ayudaros del libro de texto, capítulo 9, y consultar, si lo creéis necesario, internet.

Debéis subir un único archivo comprimido que contenga los ficheros .java o los proyectos creados, con los comentarios que creáis necesarios.

Los ejercicios tienen la siguiente puntuación: el ejercicio 1, 3 valen 2 puntos cada uno, el ejercicio 2 tiene un valor de 4 puntos, y los ejercicios 4 y 5 tienen un valor de 1 punto cada uno.

Se valorará positivamente el correcto uso del nombre en los nombres de variables, métodos y clases.

Recordad que la fecha límite para la entrega de esta PAC es el 27 de OCTUBRE.

**Sugerencia:** Para la realización de estos ejercicios es necesario utilizar Eclipse.

**Información:** Cualquier PAC copiada y/o en la que se haya utilizado “copy-paste” de código ya escrito será puntuada con un 0.

## Actividades

### Parte práctica

1. Defina una interfaz que denominará **ICalculadora** con los siguientes métodos públicos:
  - a. **sumar**: Devuelve **double** y no recibirá parámetros.
  - b. **sumar**: Devuelve **double** y recibirá un **double**.
  - c. **restar**: Devuelve **double** y no recibirá ningún parámetro.
  - d. **getNum1** y **getNum2**: Devuelven **double**.
  - e. **setNum1** y **setNum2**: Devuelven **void** y reciben un **double**.

```
package calculadora;
```

```
public interface ICalculadora {  
    public double getNum1();  
    public void setNum1(double num);  
    public double getNum2();  
    public void setNum2(double num);  
    public double sumar();  
}
```

```
    public double restar ();  
    public double sumar (double n3);  
}
```

2. Construya una clase **Calculadora** que implemente la interface **ICalculadora** y tenga dos atributos privados de tipo **double** denominados **num1** y **num2**.

La clase **Calculadora** implementará los siguientes métodos:

- Un **constructor** sin parámetros que inicie los números a 0.
- Un **constructor** que reciba un valor de tipo **double** para cada atributo.
- Un método **restar** que reste los números.
- Dos métodos **sumar**: uno de ellos sólo sumará los dos números, y el otro sumará un tercer número.
- Un método **potencia** que eleve uno de los números a una potencia, utilizando la clase **Math**.
- Un método **raizCuadrada** que realice la raíz cuadrada de uno de los números, utilizando la clase **Math**.

```
package calculadora;
```

```
public class Calculadora implements ICalculadora {  
    private double num1, num2;
```

```
    Calculadora() {  
        this.num1 = 0;  
        this.num2 = 0;  
    }
```

```
    Calculadora(double n1, double n2) {  
        this.num1 = n1;  
        this.num2 = n2;  
    }
```

```
    public double getNum1() {  
        return this.num1;  
    }
```

```
    public void setNum1(double num) {  
        this.num1 = num;  
    }
```

```
    public double getNum2() {  
        return this.num2;  
    }
```

```
    public void setNum2(double num) {  
        this.num2 = num;  
    }
```

```
    public double sumar() {  
        return this.num1 + this.num2;  
    }
```

```

public double restar () {
    return this.num1 - this.num2;
}

public double sumar (double n3) {
    return this.num1 + this.num2 + n3;
}

public double potencia (int p) {
    return Math.pow(this.num1, p);
}

public double raizCuadrada () {
    return Math.sqrt(this.num1);
}
}

```

3. Escriba un programa principal para probar todos los métodos anteriores.

a. Crear dos objetos:

- i. El primero se inicializarán los dos números a 0 y después se añadirán los números según se reciban por teclado.
- ii. El segundo se inicializará directamente con los números introducidos por teclado.

```

package calculadora;

import java.util.Scanner;

public class Principal {
    static Scanner scan = new Scanner(System.in);

    public static void main (String[] args) {
        //Vamos a crear dos objetos calculadora para hacer uso de
        //los dos tipos de constructores
        //Primero creamos un objeto sin parámetros, por lo que sus
        //números serán ambos 0
        Calculadora calc = new Calculadora();
        double n1, n2;
        Calculadora calc2;
        int potencia;

        //Los primeros números se cargan en el objeto creado y se
        //modifican sus valores
        System.out.println("Primer número: ");
        calc.setNum1(scan.nextDouble());

        System.out.println("Segundo número: ");
        calc.setNum2(scan.nextDouble());

        //Ahora recogemos los datos para el segundo objeto
        System.out.println("Tercer número: ");
        n1 = scan.nextDouble();
    }
}

```

```

        System.out.println("Cuarto número: ");
        n2 = scan.nextDouble();

        //Creamos el objeto con los datos guardados anteriormente
        calc2 = new Calculadora(n1, n2);

        //Sumas
        System.out.println(calc.getNum1() + " + " + calc.getNum2()
+ " = " + calc.sumar());
        System.out.println(calc2.getNum1() + " + " +
calc2.getNum2() + " = " + calc2.sumar());

        //Utilizamos los datos guardados anteriormente para
realizar las sumas de tres elementos
        System.out.println(calc.getNum1() + " + " + calc.getNum2()
+ " + " + n1 + " = " + calc.sumar(n1));
        System.out.println(calc2.getNum1() + " + " +
calc2.getNum2() + " + " + n2 + " = " + calc2.sumar(n2));

        //Resta
        System.out.println(calc.getNum1() + " - " + calc.getNum2()
+ " = " + calc.restar());
        System.out.println(calc2.getNum1() + " - " +
calc2.getNum2() + " = " + calc2.restar());

        //Potencia de un número
        System.out.println("Potencia: ");
        potencia = scan.nextInt();
        System.out.println(calc2.getNum1() + " elevado a " +
potencia + " es " + calc2.potencia(potencia));

        //Raiz cuadrada
        System.out.println("La raiz cuadrado de " +
calc2.getNum1() + " es " + calc2.raizCuadrada());
    }
}

```

4. Crea el programa 'Hola Mundo' con una interfaz de Swing.

```

import java.awt.BorderLayout;
import java.awt.Container;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class HolaMundo extends JFrame {
    JFrame frame;
    JPanel panel;
    JLabel lbl1;
    Container contentPane;
}

```

```
HolaMundo() {
    frame = new JFrame("Hola Mundo con Swing");
    panel = new JPanel();
    lbl1 = new JLabel("HOLA MUNDO :");

    contentPane = frame.getContentPane();
}

public void inicializar() {
    frame.setSize(200, 200);
    frame.setVisible(true);

    panel.add(lbl1);

    contentPane.add(panel, BorderLayout.CENTER);
}

public static void main (String argv[]) {
    HolaMundo h = new HolaMundo();
    h.inicializar();
}
}
```

5. Crea una interfaz Swing que tenga dos botones y un TextView, y cada vez que se pulse en un botón se mostrará algo diferentes en ese TextView.

```
import java.awt.BorderLayout;
import java.awt.Container;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class Ej5 extends JPanel {
    JFrame frame;
    JPanel panel;
    JButton btn1;
    JButton btn2;
    JTextField txt;
    Container contentPane;

    Ej5() {
        frame = new JFrame("Botones con Swing");
        panel = new JPanel();
        btn1 = new JButton("Botón 1");
        btn2 = new JButton("Botón 2");
        txt = new JTextField( 20 );

        ActionListener actList = new ActionListener () {
            @Override
            public void actionPerformed(ActionEvent arg0) {
```

```
        // TODO Auto-generated method stub
        String nombre = ((JButton)
arg0.getSource()).getText();
        txt.setText(nombre + " pulsado");
    }
};

btn1.addActionListener(actList);
btn2.addActionListener(actList);

contentPane = frame.getContentPane();
}

public void inicializar() {
    frame.setSize(300, 100);
    frame.setVisible(true);

    panel.add(txt);
    panel.add(btn1);
    panel.add(btn2);

    contentPane.add(panel, BorderLayout.CENTER);
}

public static void main (String args[]) {
    Ej5 ej5 = new Ej5();
    ej5.inicializar();
}
}
```