

CURSO : Algoritmos y Estructura de Datos
PROFESOR : Sigfrido Alva
SEMESTRE : 2023 – 01
CICLO : Segundo
SECCIÓN : T2IC
GRUPO : 00
DURACIÓN : 50 minutos
FECHA : Viernes 28 de abril del 2023

NOTA

ALUMNO (A) : Hidetochi Junior Rodriguez Sandoval

CASO DE LABORATORIO 1

Consideraciones generales:

- La solución debe ser presentada en el archivo **AED_ApellidosNombres_Solucion_CL1.pdf**.
- Lea cuidadosamente la rúbrica para cumplir con todo lo solicitado.
- Si dos o más alumnos presentan la misma solución, recibirán la nota 99 (cero anulado).

LOGRO

Al término del curso, el alumno implementa los conceptos de encapsulamiento, herencia y polimorfismo de la programación orientada a objetos en la construcción de aplicaciones java básicas que realizan mantenimiento de datos almacenados en arreglo de objetos y archivos planos.

Consolidado

Pregunta	Puntaje		Llenar solo en caso de recalificación justificada	
	Máximo	Obtenido	Sustento	Puntaje
1	7			
2	13			
Nota Recalificada				

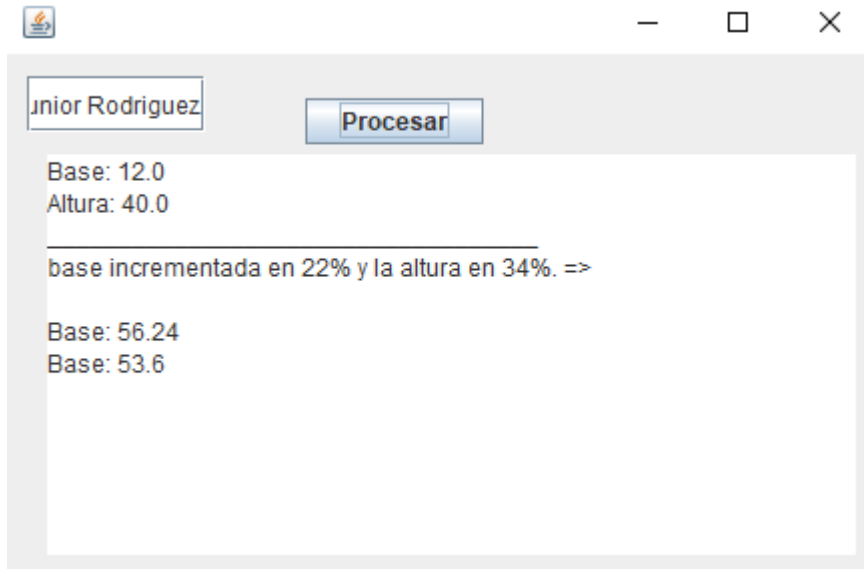
Problema 1 (7 puntos)

Diseñe la clase **Rectangulo** en el paquete **c11** con los atributos privados: *base (double)* y *altura (double)*. Usando parámetros con nombres iguales a los de los atributos, implemente lo siguiente:

- Un constructor que inicialice todos los atributos con valores recibidos a través de parámetros.
- Métodos de acceso set/get para todos los atributos.
- Un método que retorne el área total ($\text{base} * \text{altura}$).
- Un método que retorne el perímetro del rectángulo ($2 * (\text{base} + \text{altura})$).

En la clase **frmRectangulo**, ante la pulsación del botón Procesar::

- Cree un objeto perteneciente a la clase **Rectangulo** con datos fijos.
- Muestre todos los datos del objeto mediante un método **listadoRectangulo**. Para ello, el método **listadoRectangulo** recibirá, a través de un parámetro, la dirección de memoria del objeto **Rectangulo** a listar.
- Incremente el la base en 22% y la altura en 34%.
- Muestre nuevamente todos los datos del objeto mediante el método **listadoRectangulo**



```
*Rectangulo.java  frmRectangulo.java  Arquitecto.java

1 package c11;
2
3 public class Rectangulo {
4
5     //Junior Rodriguez Sandoval
6
7     private double base, altura;
8
9     public Rectangulo(double base, double altura) {
10         this.base = base;
11         this.altura = altura;
12     }
13
14     public double getBase() {
15         return base;
16     }
17
18     public void setBase(double base) {
19         this.base = base;
20     }
21
22     public double getAltura() {
23         return altura;
24     }
25
26     public void setAltura(double altura) {
27         this.altura = altura;
28     }
29
30     public double RetornarAreaTotal() {
31         return this.base * this.altura;
32     }
33
34     public double RetornarPerimetroRectangulo() {
35         return 2*(this.base + this.altura);
36     }
37
38 }
39
```

```

1 package c11;
2
3 import java.awt.BorderLayout;
4
5 //Junior Rodriguez Sandoval
6
7 public class frmRectangulo extends JFrame {
8
9     private JPanel contentPane;
10
11     public static void main(String[] args) {
12         EventQueue.invokeLater(new Runnable() {
13             public void run() {
14                 try {
15                     frmRectangulo frame = new frmRectangulo();
16                     frame.setVisible(true);
17                 } catch (Exception e) {
18                     e.printStackTrace();
19                 }
20             }
21         });
22     }
23
24     public frmRectangulo() {
25         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
26         setBounds(100, 100, 450, 300);
27         contentPane = new JPanel();
28         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
29         setContentPane(contentPane);
30         contentPane.setLayout(null);
31
32         JTextArea textArea = new JTextArea();
33         textArea.setBounds(20, 50, 404, 200);
34         contentPane.add(textArea);
35
36         JButton btnProcesar = new JButton("Procesar");
37         btnProcesar.addActionListener(new ActionListener() {
38             public void actionPerformed(ActionEvent arg0) {
39
40                 Rectangulo rectangleOne = new Rectangulo(12, 40);
41                 listadoRectangulo(rectangleOne, false);
42                 rectangleOne.setAltura(((rectangleOne.getAltura() * 34) / 100) + rectangleOne.getAltura());
43                 rectangleOne.setBase(((rectangleOne.getBase() * 22) / 100) + rectangleOne.getBase());
44                 listadoRectangulo(rectangleOne, true);
45             }
46
47             public void listadoRectangulo(Rectangulo rectangulo, boolean bandera) {
48                 //Junior Rodriguez Sandoval
49
50                 if (bandera) {
51                     textArea.append("\nbase incrementada en 22% y la altura en 34%. =>" + "\n\nBase: " + rectangulo.getBase()
52                         + "\n\nAltura: " + rectangulo.getAltura());
53                     return;
54                 }
55
56                 textArea.append("\n\nJunior Rodriguez \n\nBase: " + rectangulo.getBase() + "\n\nAltura: " + rectangulo.getAltura()
57                     + "\n\n");
58             }
59         });
60         btnProcesar.setBounds(149, 22, 89, 23);
61         contentPane.add(btnProcesar);
62
63         JFormattedTextField frmtdtxtfldJuniorRodriguez = new JFormattedTextField();
64         frmtdtxtfldJuniorRodriguez.setText("Junior Rodriguez");
65         frmtdtxtfldJuniorRodriguez.setBounds(10, 11, 89, 28);
66         contentPane.add(frmtdtxtfldJuniorRodriguez);
67     }
68 }

```

Problema 2 (13 puntos)

Diseñe la clase **Arquitecto** en el paquete **c11** con los atributos privados: código (*int*), tarifa por contrato (*double*) y número de contratos (*int*). Usando parámetros con nombres iguales a los de los atributos, implemente:

- Una variable privada de clase que cuente la cantidad de objetos **Arquitecto** creados.
- Una variable privada de clase que acumule los sueldos netos de los objetos **Arquitecto** creados.

- En la clase **frmArquitecto**, ante la pulsación del botón Procesar:

- ```
//Junior R0driguez
JButton btnProcesar = new JButton("Procesar");
btnProcesar.addActionListener(new ActionListener() {
 public void actionPerformed(ActionEvent e) {
 Arquitecto arquitecto1 = new Arquitecto(12, 5, 23);
 listadoArquitecto(arquitecto1, 1);

 Arquitecto arquitecto2 = new Arquitecto(10, 3);
 listadoArquitecto(arquitecto2, 2);

 Arquitecto arquitecto3 = new Arquitecto();
 listadoArquitecto(arquitecto3, 3);

 listadoGeneral(arquitecto3);
 }

 public void listadoArquitecto(Arquitecto arquitecto, int index) {
 textArea.append("\n\tObjeto " + index + " " +
 "\n\nCodigo: " + arquitecto.getCodigo() +
 "\nNumero de contratos: " + arquitecto.getNumeroContratos() +
 "\nTarifa por contratos: " + arquitecto.getTarifaPorContrato() +
 "\nSueldo Bruto: " + arquitecto.SueldoBruto() +
 "\nDescuento: " + arquitecto.Descuento() +
 "\nSueldo Neto: " + arquitecto.SueldoNeto());
 }

 public void listadoGeneral(Arquitecto arquitecto3) {
 textArea.append("\n\nCantidad bjetos creados: " + Arquitecto.getCantidadObjetosArquitectoCreados()
 + "\nSueldo neto acumulado: " + Arquitecto.getAcumuladorSueldosNetosObjetosArquitectos());
 }
}
```

```

1 package c11;
2
3 public class Arquitecto {
4
5 //Junior Rodriguez Sandoval
6
7 private int codigo, numeroContratos;
8 private double tarifaPorContrato;
9 private static int cantidadObjetosArquitectoCreados = 0;
10 private static double acumuladorSueldosNetosObjetosArquitectos = 0.0;
11 private static final double PORDCO = 0.12;
12
13 public Arquitecto(int codigo, int numeroContratos, double tarifaPorContrato) {
14 Arquitecto.cantidadObjetosArquitectoCreados = Arquitecto.cantidadObjetosArquitectoCreados + 1;
15 Arquitecto.acumuladorSueldosNetosObjetosArquitectos += Arquitecto.acumuladorSueldosNetosObjetosArquitectos;
16 this.codigo = codigo;
17 this.numeroContratos = numeroContratos;
18 this.tarifaPorContrato = tarifaPorContrato;
19 }
20
21 public Arquitecto (int codigo, double tarifaPorContrato) {
22 this(codigo, 6 , tarifaPorContrato);
23 }
24
25 public Arquitecto() {
26 this(-1, 0, 0.0);
27 }
28
29 public int getCodigo() {
30 return codigo;
31 }
32
33 public void setCodigo(int codigo) {
34 this.codigo = codigo;
35 }
36
37 public int getNumeroContratos() {
38 return numeroContratos;
39 }
40
41 public void setNumeroContratos(int numeroContratos) {

```

```

43 }
44
45 public double getTarifaPorContrato() {
46 return tarifaPorContrato;
47 }
48
49 public void setTarifaPorContrato(double tarifaPorContrato) {
50 this.tarifaPorContrato = tarifaPorContrato;
51 }
52
53 public static int getCantidadObjetosArquitectoCreados() {
54 return cantidadObjetosArquitectoCreados;
55 }
56
57 public static void setCantidadObjetosArquitectoCreados(int cantidadObjetosArquitectoCreados) {
58 Arquitecto.cantidadObjetosArquitectoCreados = cantidadObjetosArquitectoCreados;
59 }
60
61 public static double getAcumuladorSueldosNetosObjetosArquitectos() {
62 return acumuladorSueldosNetosObjetosArquitectos;
63 }
64
65 public static void setAcumuladorSueldosNetosObjetosArquitectos(double acumuladorSueldosNetosObjetosArquitectos) {
66 Arquitecto.acumuladorSueldosNetosObjetosArquitectos = acumuladorSueldosNetosObjetosArquitectos;
67 }
68
69 public static double getPordcto() {
70 return PORDCO;
71 }
72
73 public double SueldoBruto() {
74 return this.tarifaPorContrato * this.numeroContratos;
75 }
76
77 public double Descuento() {
78 return this.PORDCO * SueldoBruto();
79 }
80
81 public void SueldoNeto() {
82 Arquitecto.acumuladorSueldosNetosObjetosArquitectos += SueldoBruto() - Descuento();

```

## RÚBRICA

| <b>Problema 1</b> | <b>7 puntos</b>                                                                                                                                             | <b>5 puntos</b>                                                                                                             | <b>3 puntos</b>                                                                                               | <b>1 puntos</b>                                                                                           |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| 7 Puntos          | Implementa la clase completamente. Asimismo, crea el objeto solicitado, y muestra y modifica sus datos.                                                     | Implementa la clase completamente (atributos, constructor, métodos de acceso y operaciones)                                 | Implementa la clase parcialmente (atributos, constructor y métodos de acceso)                                 | Implementa la clase parcialmente (atributos)                                                              |
| <b>Problema 2</b> | <b>13 puntos</b>                                                                                                                                            | <b>8 puntos</b>                                                                                                             | <b>5 puntos</b>                                                                                               | <b>3 puntos</b>                                                                                           |
| 13 puntos         | Implementa la clase completamente. Asimismo, crea los tres objetos solicitados y, muestra los datos de cada objeto y los valores de las variables de clase. | Implementa la clase completamente (atributos, variables de clase, todos los constructores, métodos de acceso y operaciones) | Implementa la clase parcialmente (atributos, variables de clase, todos los constructores y métodos de acceso) | Implementa la clase parcialmente (atributos, variables de clase, constructor general y métodos de acceso) |