

CURSO : Algoritmos y Estructura de Datos
PROFESOR : Sigfrido Alva
SEMESTRE : 2023 – 01
CICLO : Segundo
SECCIÓN : T2IC
GRUPO : 00
DURACIÓN : 50 minutos
FECHA : 26 de mayo del 2023

NOTA

ALUMNO (A) :

CASO DE LABORATORIO 2

Consideraciones generales:

- La solución debe ser presentada en el archivo **AED_Solucion_ApellidosNombres_CL2.pdf**.
- El color del texto debe ser legible, de lo contrario, la solución no será calificada.
- 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	20			
Nota Recalificada				

Problema (20 puntos)

Dada la clase **ArregloSueldos** ubicada en el paquete **examen**:

```
package examen;
```

```
public class ArregloSueldos {  
    private double sueldo[];  
    private int indice;
```

```

public ArregloSueldos() {
    sueldo = new double[10];
    indice = 0;
}

public int tamaño() {
    return indice;
}

public double obtener(int p) {
    return sueldo[p];
}

public void adicionar(double s) {
    if (indice == tamaño())
        ampliarArreglo();
    sueldo[indice] = s;
    indice++;
}

private void ampliarArreglo() {
    double aux[] = sueldo;
    sueldo = new double[indice + 10];
    for (int i = 0; i < indice; i++)
        sueldo[i] = aux[i];
}

public void eliminarTodo() {
    indice = 0;
}

public void eliminarFinal() {
    indice--;
}
}

```

Considerando que el arreglo será ingresado progresivamente con datos tomados de la GUI y que podría no encontrarse completamente lleno, implemente:

En la clase **ArregloSueldos**:

- Un método denominado **posicionUltimoSueldoMenorQue1000**, que busque y retorne la posición del último sueldo menor que 1000. En caso que no exista, retornar -1.
- Un método denominado **sumaSueldosMayoresQue850**, que retorne la suma de todos los sueldos mayores que 850.
- Un método denominado **reemplazarUltimoSueldoMenorQue1000**, que reemplace el último sueldo menor que 1000, por la suma de los sueldos mayores que 850. Si el reemplazo es posible, retorne **true**; en caso contrario, retorne **false**.
- Un método denominado **incrementarSueldosMenoresQue1200** que incremente en 15% todos los sueldos menores que 1200 y retorne la cantidad de incrementos efectuados.
- Un método denominado **eliminarUltimoSueldoMenorQue1000**, que elimine el último sueldo menor que 1000. Si la eliminación es posible, retorne **true**; en caso contrario, retorne **false**.
-

En la clase **Problema**:

- Ante la pulsación del botón **Reemplazar último sueldo menor que 1000**, invoque al método **reemplazarUltimoSueldoMenorQue1000** e invoque al método **listar**. En caso de que no exista ningún sueldo menor que 1000, muestre el mensaje "No existe ningún sueldo menor que 1000".

- Ante la pulsación del botón **Incrementar sueldos menores que 1200** invoque al método **incrementarSueldosMenoresQue1200** e invoque al método **listar**. Muestre, además, un mensaje indicando cuantos sueldos fueron incrementados.
- Ante la pulsación del botón **Eliminar último sueldo menor que 1000**, invoque al método **eliminarUltimoSueldoMenorQue1000** e invoque al método **listar**. En caso de que no exista ningún sueldo menor que 1000, muestre el mensaje "".

```
ArregloSueldos.java  *Problema.java
1 package examen;
2
3 public class ArregloSueldos {
4     private double sueldo[];
5     private int indice;
6     private int sueldosIncrementados;
7
8     public ArregloSueldos() {
9         sueldo = new double[10];
10        indice = 0;
11        sueldosIncrementados = 0;
12    }
13
14    public int getSueldosIncrementados() {
15        return sueldosIncrementados;
16    }
17
18    public void setSueldosIncrementados(int sueldosIncrementados) {
19        this.sueldosIncrementados = sueldosIncrementados;
20    }
21
22
23    public double[] getSueldo() {
24        return sueldo;
25    }
26
27    public void setSueldo(double[] sueldo) {
28        this.sueldo = sueldo;
29    }
30
31    public int getIndice() {
32        return indice;
33    }
34
35    public void setIndice(int indice) {
36        this.indice = indice;
37    }
38
39    public int tamaño() {
40        return indice;
41    }
42 }
```

```
ArregloSueldos.java  *Problema.java

41     }
42
43     public double obtener(int p) {
44         return sueldo[p];
45     }
46
47     public boolean ingresarSueldo (double sueldoIngresante) {
48
49         if(sueldo[indice] == 0) {
50             sueldo[indice] = sueldoIngresante;
51             indice++;
52             return true;
53         } else {
54             return false;
55         }
56
57     }
58
59     public void adicionar(double s) {
60         if (indice == tamaño())
61             ampliarArreglo();
62         sueldo[indice] = s;
63         indice++;
64     }
65
66     private void ampliarArreglo() {
67         double aux[] = sueldo;
68         sueldo = new double[indice + 10];
69         for (int i = 0; i < indice; i++)
70             sueldo[i] = aux[i];
71     }
72
73     public void eliminarTodo() {
74         indice = 0;
75     }
76
77     public void eliminarFinal() {
78         indice--;
79     }
80 }
```

```

80
81 public int posicionUltimoSueldoMenorQue1000 () {
82
83     for (int i = 0; i < sueldo.length; i++) {
84         if(sueldo[i] < 1000 ) {
85             return sueldo.length - 1;
86         } else {
87             return -1;
88         }
89     }
90     return 0;
91 }
92
93 public double sumaSueldosMayoresQue850() {
94     double acum = 0;
95
96     for (int i = 0; i < sueldo.length; i++) {
97         if(sueldo[i] > 850) {
98             acum = acum + sueldo[i];
99         }
100
101     }
102     return 0;
103 }
104
105 public boolean reemplazarUltimoSueldoMenorQue1000 () {
106     int index = posicionUltimoSueldoMenorQue1000();
107
108     if(index != -1) {
109         sueldo[index] = sumaSueldosMayoresQue850();
110         return true;
111     }
112
113     return false;
114
115 }
116
117 public double incrementarSueldosMenoresQue1200 () {
118     for (int i = 0; i < sueldo.length; i++) {

```

```

119         if(sueldo[i] < 1200) {
120             sueldosIncrementados++;
121             return sueldo[i] * 1.15;
122         }
123     }
124
125     return 0;
126 }
127
128 public boolean eliminarUltimoSueldoMenorQue1000 () {
129     int index = posicionUltimoSueldoMenorQue1000();
130
131     if(index != -1) {
132         sueldo[index] = 0;
133     }
134
135     return false;
136 }

```

PROBLEMA

RÚBRICA

Problema	20.0 puntos	15.0 puntos	10.0 puntos	5.0 puntos
20 Puntos	Implementa correctamente los ocho métodos solicitados.	Implementa correctamente seis métodos solicitados.	Implementa correctamente cuatro métodos solicitados.	Implementa correctamente dos métodos solicitados.