

PROGRAMACIÓN II

Trabajo Práctico N.º 4: POO

Estudiante: Emilia Gómez Juárez

Objetivo:

Comprender y aplicar conceptos de Programación Orientada a Objetos en Java, incluyendo el uso de this, constructores, sobrecarga de métodos, encapsulamiento y miembros estáticos, para mejorar la modularidad, reutilización y diseño del código.

Caso Práctico:

Sistema de Gestión de Empleados

Modelar una clase Empleado que represente a un trabajador en una empresa. Esta clase debe incluir constructores sobrecargados, métodos sobrecargados y el uso de atributos y métodos estáticos para llevar control de los objetos creados.

CLASE EMPLEADO

Atributos:

- ✓ int id: Identificador único del empleado.
- ✓ String nombre: Nombre completo.
- ✓ String puesto: Cargo que desempeña.
- ✓ double salario: Salario actual.
- ✓ static int totalEmpleados: Contador global de empleados creados.

REQUERIMIENTOS

1. Uso de this:
 - Utilizar this en los constructores para distinguir parámetros de atributos.
2. Constructores sobrecargados:
 - Uno que reciba todos los atributos como parámetros.

PROGRAMACIÓN II

- Otro que reciba solo nombre y puesto, asignando un id automático y un salario por defecto.
- Ambos deben incrementar totalEmpleados.
- 3. Métodos sobrecargados actualizarSalario:
 - Uno que reciba un porcentaje de aumento.
 - Otro que reciba una cantidad fija a aumentar.
- 4. Método toString():
 - Mostrar id, nombre, puesto y salario de forma legible.
- 5. Método estático mostrarTotalEmpleados():
 - Retornar el total de empleados creados hasta el momento.

TAREAS A REALIZAR

- a) Implementar la clase Empleado aplicando todos los puntos anteriores.
- b) Crear una clase de prueba con método main que:
 - Instancie varios objetos usando ambos constructores.
 - Aplique los métodos actualizarSalario() sobre distintos empleados.
 - Imprima la información de cada empleado con toString().
 - Muestre el total de empleados creados con mostrarTotalEmpleados().

PROGRAMACIÓN II

Clase empleado:

```
10 public class Empleado {
11     private int id;
12     private String nombre;
13     private String puesto;
14     private double salario;
15     private static int totalEmpleados = 0;
16
17     public Empleado(int id, String nombre, String puesto,
18         double salario) {
19         this.id = id;
20         this.nombre = nombre;
21         this.puesto = puesto;
22         this.salario = salario;
23         totalEmpleados++;
24     }
25
26     public Empleado(String nombre, String puesto) {
27         totalEmpleados++;
28         this.nombre = nombre;
29         this.puesto = puesto;
30         this.id = totalEmpleados;
31         this.salario = 950000.00;
32     }
33 }
```

```
35 public int getId() {
36     return id;
37 }
38
39 public void setId(int id) {
40     this.id = id;
41 }
42
43 public String getNombre() {
44     return nombre;
45 }
46
47 public void setNombre(String nombre) {
48     this.nombre = nombre;
49 }
50
51 public String getPuesto() {
52     return puesto;
53 }
54
55 public void setPuesto(String puesto) {
56     this.puesto = puesto;
57 }
58
59 public double getSalario() {
60     return salario;
61 }
62
63 public void setSalario(double salario) {
64     this.salario = salario;
65 }
```

PROGRAMACIÓN II

```
67 public void actualizarSalario(int porcentajeAumento) {
68     this.salario += (this.salario * porcentajeAumento/100);
69 }
70
71 public void actualizarSalario(double cantidadAumento) {
72     this.salario += cantidadAumento;
73 }
74
75 @Override
76 public String toString() {
77     return "Empleado{" +
78         "id=" + id +
79         ", nombre=" + nombre + '\n' +
80         ", puesto=" + puesto + '\n' +
81         ", salario=" + salario +
82         '}';
83 }
84
85 public static int mostrarTotalEmpleados() {
86     return totalEmpleados;
87 }
88
89 }
90
```

Main:

```
15 public static void main(String[] args) {
16     Empleado empleado1 = new Empleado(1, "Juan Molina", "Gerente", 3500000);
17     Empleado empleado2 = new Empleado("Pablo Miralles", "Administrativo");
18     Empleado empleado3 = new Empleado("Ramiro Juárez", "Cajero");
19     Empleado empleado4 = new Empleado(4, "Sofia Perez", "Administrativa", 1500000);
20     Empleado empleado5 = new Empleado("Aldana Fonda", "Cajera");
21
22     System.out.println("Datos de empleados 2 y 5 antes de actualizar sueldos");
23     mostrarEmpleado(empleado2);
24     mostrarEmpleado(empleado5);
25
26     empleado2.actualizarSalario(550000.00);
27     empleado5.actualizarSalario(15);
28
29     System.out.println("\nDatos de todos los empleados con sueldos actualizados");
30     mostrarEmpleado(empleado1);
31     mostrarEmpleado(empleado2);
32     mostrarEmpleado(empleado3);
33     mostrarEmpleado(empleado4);
34     mostrarEmpleado(empleado5);
35
36     System.out.println("\nEl total de empleados registrados es: "
37         + Empleado.mostrarTotalEmpleados());
38 }
39 public static void mostrarEmpleado(Empleado empleado) {
40     System.out.println(empleado.toString());
41 }
42 }
```

PROGRAMACIÓN II

Resultado en consola:

```
run:
Datos de empleados 2 y 5 antes de actualizar sueldos
Empleado{id=2, nombre='Pablo Miralles', puesto='Administrativo', salario=950000.0}
Empleado{id=5, nombre='Aldana Fonda', puesto='Cajera', salario=950000.0}

Datos de todos los empleados con sueldos actualizados
Empleado{id=1, nombre='Juan Molina', puesto='Gerente', salario=3500000.0}
Empleado{id=2, nombre='Pablo Miralles', puesto='Administrativo', salario=1500000.0}
Empleado{id=3, nombre='Ramiro Juárez', puesto='Cajero', salario=950000.0}
Empleado{id=4, nombre='Sofia Perez', puesto='Administrativa', salario=1500000.0}
Empleado{id=5, nombre='Aldana Fonda', puesto='Cajera', salario=1092500.0}

El total de empleados registrados es: 5
BUILD SUCCESSFUL (total time: 0 seconds)
```

REPOSITORIO

REMOTO:

<https://github.com/GomezJEmilia/UTN-Programacion2-TPs-EmiliaGJ.git>