PROGRAMACIÓN II

Trabajo Práctico 4: Programación Orientada a Objetos II (Resolución)

Alumno: Mauro Gaspar
Comisión: 4

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 aplicando encapsulamiento 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.

TAREAS A REALIZAR
1. Implementar la clase Empleado aplicando todos los puntos anteriores.
2. Crear una clase de prueba con método main que:
   ○ Instancie varios objetos usando ambos constructores.

```java
public class Empleado {
    private int id;
    private String nombre;
    private String puesto;
    private double salario;


    // Atributos estáticos para control global
    private static int totalEmpleados = 0; // contador de instancias creadas
    private static int contadorId = 0;     // usado para generar ids automáticos

    // Constructor que recibe todos los atributos (uso de "this")
    public Empleado(int id, String nombre, String puesto, double salario) {
        this.id = id; // this para diferenciar atributo/parametro
        this.nombre = nombre;
        this.puesto = puesto;
        this.salario = salario;
        totalEmpleados++; // incrementa contador global
        // Aseguramos que el contadorId avance si se pasó un id mayor
        if (id > contadorId) {
            contadorId = id;
        }
    }

    // Constructor sobrecargado: sólo nombre y puesto
    // Asigna id automático y salario por defecto
    public Empleado(String nombre, String puesto) {
        this.id = ++contadorId; // id automático (pre-incremento)
        this.nombre = nombre;
        this.puesto = puesto;
        this.salario = 10000.0; // salario por defecto (ejemplo)
        totalEmpleados++;
    }

    // Getter y Setters
    public int getId() {
        return id;
    }
```

```java
class TestEmpleado {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("¿Cuántos empleados querés cargar? ");
        int n = sc.nextInt();
        sc.nextLine(); // limpiar buffer

        Empleado[] empleados = new Empleado[n]; // arreglo para guardar empleados

        for (int i = 0; i < n; i++) {
            System.out.println("\nEmpleado #" + (i + 1));

            System.out.print("Nombre: ");
            String nombre = sc.nextLine();

            System.out.print("Puesto: ");
            String puesto = sc.nextLine();

            System.out.print("¿Deseás ingresar salario inicial? (s/n): ");
            String resp = sc.nextLine();

            if (resp.equalsIgnoreCase("s")) {
                System.out.print("Salario: ");
                double salario = sc.nextDouble();
                sc.nextLine();
                empleados[i] = new Empleado(i + 1, nombre, puesto, salario);
            } else {
                empleados[i] = new Empleado(nombre, puesto); // salario por defecto
            }
        }

        // Mostrar todos los empleados cargados
        System.out.println("\n*** Lista de empleados cargados ***");
        for (Empleado e : empleados) {
            System.out.println(e);
        }
```

```
    // Mostrar todos los empleados cargados
    System.out.println("\n*** Lista de empleados cargados ***");
    for (Empleado e : empleados) {
        System.out.println(e);
    }

    // Aplicar actualizaciones de salario a todos
    System.out.print("\n¿Querés actualizar los salarios por porcentaje o monto fijo? (p/m):
    String opcion = sc.nextLine();


    if (opcion.equalsIgnoreCase("p")) {
        System.out.print("Ingrese el porcentaje de aumento: ");
        double porc = sc.nextDouble();
        for (Empleado e : empleados) {
            e.actualizarSalario(porc); // usa versión porcentaje
        }
    } else {
        System.out.print("Ingrese el monto fijo a sumar: ");
        int monto = sc.nextInt();
        for (Empleado e : empleados) {
            e.actualizarSalario(monto); // usa versión fija
        }
    }

    // Mostrar lista después de actualización
    System.out.println("\n=== Empleados después de actualización ===");
    for (Empleado e : empleados) {
        System.out.println(e);
    }

    System.out.println("\nTotal empleados creados: " + Empleado.mostrarTotalEmpleados());

    sc.close();
}
```

tp4 (run) × NetBeansProjects - D:\Mauro\Documents\NetBeansProjects ×

```
run:
¿Cuántos empleados querés cargar? 2

Empleado #1
Nombre: Tomas
Puesto: Desarrollador
¿Deseás ingresar salario inicial? (s/n): n

Empleado #2
Nombre: Mauro
Puesto: Analista
¿Deseás ingresar salario inicial? (s/n): s
Salario: 50000

*** Lista de empleados cargados ***
Empleado {id=1, nombre='Tomas', puesto='Desarrollador', salario=10000,00}
Empleado {id=2, nombre='Mauro', puesto='Analista', salario=50000,00}

¿Querés actualizar los salarios por porcentaje o monto fijo? (p/m): p
Ingrese el porcentaje de aumento: 15

=== Empleados después de actualización ===
Empleado {id=1, nombre='Tomas', puesto='Desarrollador', salario=11500,00}
Empleado {id=2, nombre='Mauro', puesto='Analista', salario=57500,00}

Total empleados creados: 2
BUILD SUCCESSFUL (total time: 1 minute 10 seconds)
```