

PROGRAMACIÓN II

Trabajo Práctico 4: Programación Orientada a Objetos II

Alumno: Mauro Zavatti

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.
 - 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

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.
 - Aplique los métodos `actualizarSalario()` sobre distintos empleados.
 - Imprima la información de cada empleado con `toString()`.
 - Muestre el total de empleados creados con `mostrarTotalEmpleados()`.

```
package tp.pkg4.poo;
```

```
public class Empleado {
```

```
    private int id;
```

```
    private String nombre;
```

```
    private String puesto;
```

```
    private double salario;
```

```
    private static int totalEmpleados = 0;
```

```
    private static int contadorId = 1;
```

```
    // Constructor con todos los atributos
```

```
    public Empleado(int id, String nombre, String puesto, double salario) {
```

```
        this.id = id;
```

```
        this.nombre = nombre;
```

```
        this.puesto = puesto;
```

```
        this.salario = salario;
```

```
        totalEmpleados++;
```

```
    }
```

```
    // Constructor con nombre y puesto (ID automático y salario por defecto)
```

```
    public Empleado(String nombre, String puesto) {
```

```
        this.id = contadorId++;
```

```
        this.nombre = nombre;
```

```
        this.puesto = puesto;
```

```
        this.salario = 50000; // valor por defecto
```

```
        totalEmpleados++;
```

```
    }
```

```
    // Sobrecarga actualizarSalario con porcentaje
```

```
public void actualizarSalario(double porcentaje) {  
    this.salario += this.salario * (porcentaje / 100);  
}
```

```
// Sobrecarga actualizarSalario con monto fijo
```

```
public void actualizarSalario(int monto) {  
    this.salario += monto;  
}
```

```
// Método estático
```

```
public static int mostrarTotalEmpleados() {  
    return totalEmpleados;  
}
```

```
@Override
```

```
public String toString() {  
    return "Empleado{" +  
        "id=" + id +  
        ", nombre=" + nombre + " +  
        ", puesto=" + puesto + " +  
        ", salario=" + salario +  
        "}";  
}  
  
}
```

```
package tp.pkg4.poo;
```

```
public class EmpleadoMain {  
    public static void main(String[] args) {  
        Empleado e1 = new Empleado(100, "Ana", "Gerente", 80000);  
        Empleado e2 = new Empleado("Luis", "Programador");  
        Empleado e3 = new Empleado("Carla", "Diseñadora");  
  
        e2.actualizarSalario(10); // aumento 10%  
        e3.actualizarSalario(5000); // aumento fijo  
  
        System.out.println(e1);  
        System.out.println(e2);  
        System.out.println(e3);  
  
        System.out.println("Total empleados: " +  
Empleado.mostrarTotalEmpleados());  
    }  
}
```