

UTN-TUPaD-P2/módulo 4/tp4_POO at main ·
jazmin-herrera/UTN-TUPaD-P2

TP4: POO

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.

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.

6. Encapsulamiento en los atributos:

- Restringir el acceso directo a los atributos de la clase.
- Crear los métodos Getters y Setters correspondientes.

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()`.

```
public class Empleado {  
  
    /**  
     * @param args the command line arguments  
     */  
  
    private int id;  
    private String nombre;  
    private String puesto;  
    private double salario;  
  
    private static int totalEmpleados = 0;  
  
    Empleado(int i, String empleado_1, String vendedor, int i0) {  
        throw new UnsupportedOperationException("Not supported yet.");  
    }  
  
    public int getId() {  
        return id;  
    }  
  
    public void setId(int id) {  
        this.id = id;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
  
    public String getPuesto() {  
        return puesto;  
    }  
  
    public void setPuesto(String puesto) {  
        this.puesto = puesto;  
    }  
}
```

```

    public double getSalario() {
        return salario;
    }

    public void setSalario(double salario) {
        this.salario = salario;
    }

    public static int getTotalEmpleados() {
        return totalEmpleados;
    }

    public static void setTotalEmpleados(int totalEmpleados) {
        Empleado.totalEmpleados = totalEmpleados;
    }

    public Empleado(int id, String nombre, String puesto, double salario) {
        Empleado.totalEmpleados++;

        this.id = id;
        this.nombre = nombre;
        this.puesto = puesto;
        this.salario = salario;
    }

    public Empleado(String nombre, String puesto) {
        Empleado.totalEmpleados++;
        this.id = Empleado.totalEmpleados;

        this.nombre = nombre;
        this.puesto = puesto;
        this.salario = 900000.0;
    }

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

    public void actualizarSalario(){
        actualizarSalario(15);
    }
}

```

```

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

public static int mostrarTotalEmpleados(){
    return Empleado.totalEmpleados;
}
}

```

Main:

```

public class MainEmpleados {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {

        Empleado emp1 = new Empleado(1, "Empleado 1", "Vendedor", 850000);
        Empleado emp2 = new Empleado(2, "Empleado 2", "Vendedor", 950000);
        Empleado emp3 = new Empleado(3, "Empleado 3", "Encargado", 1500000);
        Empleado emp4 = new Empleado("Empleado 4", "Repositor");
        Empleado emp5 = new Empleado("Empleado 5", "Limpieza");
        Empleado emp6 = new Empleado("Empleado 6", "Deposito");

        // Actualizar salarios
        emp1.actualizarSalario();           // aumento por defecto 15%
        emp3.actualizarSalario(20);         // aumento 20%
        emp5.actualizarSalario();           // aumento por defecto 15%

        // Mostramos los datos de cada empleado
        System.out.println(emp1.toString());
        System.out.println(emp2.toString());
        System.out.println(emp3.toString());
        System.out.println(emp4.toString());
        System.out.println(emp5.toString());
        System.out.println(emp6.toString());

        // Mostramos el total de empleados
        System.out.println("\nTotal empleados: " + Empleado.mostrarTotalEmpleados());
    }
}

```