

TRABAJO PRÁCTICO N° 4

Programación Orientada a Objetos

Alumno: Ignacio Figueroa – 45.406.120

Tecnicatura Universitaria en Programación – UTN

Materia: Programación II

Comisión: 7

Objetivo General

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

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.

Conclusiones

Atributos de clase Empleado

```
Source History
1 package dev.ignaciofigueroa;
2
3 /**
4  *
5  * @author Ignacio Figueroa
6  */
7 public class Empleado {
8
9     private int id;
10    private String nombre;
11    private String puesto;
12    private double salario;
13    static int totalEmpleados = 0;
14    private static int contadorId = 0;
15
16 }
17
```

Constructores

```
6 /**
7  */
8 public class Empleado {
9
10    private int id;
11    private String nombre;
12    private String puesto;
13    private double salario;
14    static int totalEmpleados = 0;
15    private static int contadorId = 0;
16
17    public Empleado(String nombre, String puesto, double salario) {
18        this.nombre = nombre;
19        this.puesto = puesto;
20        this.salario = salario;
21        this.id = contadorId++;
22        totalEmpleados++;
23    }
24
25    public Empleado(String nombre, String puesto) {
26        this.id = ++contadorId;
27        this.nombre = nombre;
28        this.puesto = puesto;
29        this.salario = 100000; // salario por defecto
30        totalEmpleados++;
31    }
32
33    public Empleado(String nombre) {
34        this.id = ++contadorId;
35        this.nombre = nombre;
36        this.puesto = "Sin asignar";
37        this.salario = 80000;
38        totalEmpleados++;
39    }
40 }
```

Métodos

```
public void actualizarSalario(double porcentaje) {  
    this.salario += this.salario * (porcentaje / 100);  
}  
  
public void actualizarSalario(int cantidadFija) {  
    this.salario += cantidadFija;  
}  
  
public static int mostrarTotalEmpleados() {  
    return totalEmpleados;  
}
```

Getters & Setters

```
public int getId() {  
    return 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;  
}
```

Override

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

Instancias en main

```
public class Tp4ProgramacionOrientadaAObjetos {

    public static void main(String[] args) {
        // Usando el constructor completo
        Empleado e1 = new Empleado("Juan Pérez", "Desarrollador", 150000);

        // Usando constructor sobrecargado (nombre + puesto)
        Empleado e2 = new Empleado("María López", "Diseñadora");

        // Usando constructor sobrecargado (solo nombre)
        Empleado e3 = new Empleado("Carlos Gómez");
    }
}
```

Demás métodos

```
e1.actualizarSalario(10.0);    // +10%
e2.actualizarSalario(20000);   // +20.000

// Mostrar info
System.out.println(e1);
System.out.println(e2);
System.out.println(e3);

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

}
```

Salida en consola

```
Output - tp4-programacion-orientada-a-objetos (run)

run:
Empleado{id=1, nombre=Juan Pérez, puesto=Desarrollador, salario=165000.0}
Empleado{id=2, nombre=María López, puesto=Diseñadora, salario=120000.0}
Empleado{id=3, nombre=Carlos Gómez, puesto=Sin asignar, salario=80000.0}

Total de empleados creados: 3
BUILD SUCCESSFUL (total time: 0 seconds)
```