**Practica**

Crear un conjunto de clases que representan, de forma simplificada, a una hipotética empresa dedicada a vender un producto . A continuación, se describan las características básicas de estas clases:

1. Empleado. Clase básica que describe a un empleado , incluye sus datos personales (nombre, apellidos, DNI, direccion) y algunos datos tales como los años de antigüedad, teléfonos de contacto y su salario. Tendrá al menos, las

siguientes funciones miembro:

Imprimir

incrementar salario

2. Secretario. tiene despacho, número de fax e incrementa su salario un 5% anual. Tendrá al menos, las siguientes funciones miembro:

Imprimir (debe imprimir sus datos personales y su puesto en la empresa)

3 Vendedor, Tiene coche de la empresa (identificado por la matricula, marca y modelo), teléfono móvil, área de venta, porcentaje que se lleva de las ventas en concepto de comisiones. incrementa su salario un 10 % anual. Tendrá, al menos, las siguientes funciones miembros:

Imprimir (debe imprimir sus datos personales y su puesto en la empresa)

**Resumen del Código**

**1. Empleado.java**

* Define la clase Empleado con atributos como nombre, apellidos, DNI, dirección, años de antigüedad, teléfono y salario.
* Métodos para obtener y establecer estos atributos.
* Métodos para imprimir la información del empleado y para incrementar el salario.

**2. GestorEmpleados.java**

* Administra una lista de empleados.
* Métodos para agregar, actualizar, eliminar empleados y para imprimir la lista de empleados.

**3. Main.java**

* Configura la salida estándar con codificación UTF-8.
* Crea instancias de Empleado, Secretario y Vendedor.
* Usa GestorEmpleados para agregar, actualizar y eliminar empleados, e imprime la lista de empleados en cada paso.

**4. Secretario.java**

* Hereda de Empleado.
* Añade atributos específicos como despacho y fax.
* Sobrescribe el método imprimir para incluir información adicional y tiene un método para incrementar el salario en un 5%.

**5. Vendedor.java**

* Hereda de Empleado.
* Añade atributos específicos como matrícula del coche, marca, modelo, teléfono móvil, área de venta y comisión.
* Sobrescribe el método imprimir para incluir esta información adicional y tiene un método para incrementar el salario en un 10%.

Empleado.java

package Main;

import java.io.PrintStream;

public class Empleado {

private String nombre;

private String apellidos;

private String dni;

private String direccion;

private int añosAntiguedad;

private String telefono;

private double salario;

public Empleado(String nombre, String apellidos, String dni, String direccion, int añosAntiguedad, String telefono, double salario) {

this.nombre = nombre;

this.apellidos = apellidos;

this.dni = dni;

this.direccion = direccion;

this.añosAntiguedad = añosAntiguedad;

this.telefono = telefono;

this.salario = salario;

}

public String getNombre() { return nombre; }

public void setNombre(String nombre) { this.nombre = nombre; }

public String getApellidos() { return apellidos; }

public void setApellidos(String apellidos) { this.apellidos = apellidos; }

public String getDni() { return dni; }

public void setDni(String dni) { this.dni = dni; }

public String getDireccion() { return direccion; }

public void setDireccion(String direccion) { this.direccion = direccion; }

public int getAñosAntiguedad() { return añosAntiguedad; }

public void setAñosAntiguedad(int añosAntiguedad) { this.añosAntiguedad = añosAntiguedad; }

public String getTelefono() { return telefono; }

public void setTelefono(String telefono) { this.telefono = telefono; }

public double getSalario() { return salario; }

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

public void imprimir(PrintStream out) {

out.println("Nombre: " + nombre + " " + apellidos);

out.println("DNI: " + dni);

out.println("Dirección: " + direccion);

out.println("Años de antigüedad: " + añosAntiguedad);

out.println("Teléfono: " + telefono);

out.println("Salario: " + salario);

}

public void incrementarSalario(double porcentaje) {

salario \*= (1 + porcentaje / 100);

}

}

GestorEmpleados.java

package Main;

import java.io.PrintStream;

import java.util.ArrayList;

import java.util.List;

public class GestorEmpleados {

private List<Empleado> empleados;

public GestorEmpleados() {

empleados = new ArrayList<>();

}

public void agregarEmpleado(Empleado empleado) {

empleados.add(empleado);

}

public void actualizarEmpleado(String dni, Empleado empleadoActualizado) {

for (int i = 0; i < empleados.size(); i++) {

if (empleados.get(i).getDni().equals(dni)) {

empleados.set(i, empleadoActualizado);

return;

}

}

}

public void eliminarEmpleado(String dni) {

empleados.removeIf(empleado -> empleado.getDni().equals(dni));

}

public void imprimirEmpleados(PrintStream out) {

for (Empleado empleado : empleados) {

empleado.imprimir(out);

out.println(); // Línea en blanco entre empleados

}

}

}

Main.java

package Main;

import java.io.OutputStreamWriter;

import java.io.PrintStream;

import java.nio.charset.StandardCharsets;

public class Main {

public static void main(String[] args) {

// Configurar la salida estándar con codificación UTF-8

PrintStream out = new PrintStream(System.out, true, StandardCharsets.UTF\_8);

GestorEmpleados gestor = new GestorEmpleados();

// Crear instancias de Empleado

Empleado empleado1 = new Empleado("Juan", "Perez", "95845898", "Av. Chiclayo", 5, "965658597", 30000);

Secretario secretario = new Secretario("Maria", "Gonzalez", "79859458", "Luis Gonzales", 10, "930607598", 40000, "Despacho 1", "fax1");

Vendedor vendedor = new Vendedor("Pedro", "Sanchez", "75811584", "La Victoria", 8, "962526458", 50000, "1234ABC", "Ford", "Focus", "962526458", "Administrador", 10);

// Agregar empleados

gestor.agregarEmpleado(empleado1);

gestor.agregarEmpleado(secretario);

gestor.agregarEmpleado(vendedor);

// Imprimir todos los empleados

out.println("Empleados después de agregar:");

gestor.imprimirEmpleados(out);

// Actualizar un empleado

Empleado empleadoActualizado = new Empleado("Juan", "Perez", "95845898", "Nueva Dirección", 6, "965658597", 31000);

gestor.actualizarEmpleado("95845898", empleadoActualizado);

// Imprimir todos los empleados después de la actualización

out.println("Empleados después de actualizar:");

gestor.imprimirEmpleados(out);

// Eliminar un empleado

gestor.eliminarEmpleado("79859458");

// Imprimir todos los empleados después de eliminar

out.println("Empleados después de eliminar:");

gestor.imprimirEmpleados(out);

}

}

Secretario.java

package Main;

import java.io.PrintStream;

public class Secretario extends Empleado {

private String despacho;

private String fax;

public Secretario(String nombre, String apellidos, String dni, String direccion, int añosAntiguedad, String telefono, double salario, String despacho, String fax) {

super(nombre, apellidos, dni, direccion, añosAntiguedad, telefono, salario);

this.despacho = despacho;

this.fax = fax;

}

@Override

public void imprimir(PrintStream out) {

super.imprimir(out);

out.println("Despacho: " + despacho);

out.println("Fax: " + fax);

out.println("Puesto: Secretario");

}

public void incrementarSalario() {

super.incrementarSalario(5);

}

public String getDespacho() { return despacho; }

public void setDespacho(String despacho) { this.despacho = despacho; }

public String getFax() { return fax; }

public void setFax(String fax) { this.fax = fax; }

}

Vendedor.java

package Main;

import java.io.PrintStream;

public class Vendedor extends Empleado {

private String matricula;

private String marca;

private String modelo;

private String telefonoMovil;

private String areaVenta;

private double comision;

public Vendedor(String nombre, String apellidos, String dni, String direccion, int añosAntiguedad, String telefono, double salario, String matricula, String marca, String modelo, String telefonoMovil, String areaVenta, double comision) {

super(nombre, apellidos, dni, direccion, añosAntiguedad, telefono, salario);

this.matricula = matricula;

this.marca = marca;

this.modelo = modelo;

this.telefonoMovil = telefonoMovil;

this.areaVenta = areaVenta;

this.comision = comision;

}

@Override

public void imprimir(PrintStream out) {

super.imprimir(out);

out.printf("Matrícula del coche: %s%n", matricula);

out.printf("Marca del coche: %s%n", marca);

out.printf("Modelo del coche: %s%n", modelo);

out.printf("Teléfono móvil: %s%n", telefonoMovil);

out.printf("Área de venta: %s%n", areaVenta);

out.printf("Comisión: %.2f%n", comision);

}

public void incrementarSalario() {

super.incrementarSalario(10);

}

public String getMatricula() { return matricula; }

public void setMatricula(String matricula) { this.matricula = matricula; }

public String getMarca() { return marca; }

public void setMarca(String marca) { this.marca = marca; }

public String getModelo() { return modelo; }

public void setModelo(String modelo) { this.modelo = modelo; }

public String getTelefonoMovil() { return telefonoMovil; }

public void setTelefonoMovil(String telefonoMovil) { this.telefonoMovil = telefonoMovil; }

public String getAreaVenta() { return areaVenta; }

public void setAreaVenta(String areaVenta) { this.areaVenta = areaVenta; }

public double getComision() { return comision; }

public void setComision(double comision) { this.comision = comision; }

}