

Ejercicio utilizando polimorfismo, herencia, clases abstractas e interfaces

En este ejercicio se desarrolló un programa centrado en la gestión de empleados y diferentes tipos de empleados, aplicando conceptos fundamentales de POO, como herencia, polimorfismo, clases abstractas e interfaces. A continuación, se presenta un resumen general de las clases implementadas:

1. **Interfaz Beneficios:** Esta interfaz define dos métodos esenciales que todos los tipos de empleados deben implementar: `asignarBonos()`, que calcula el bono que recibe el empleado, y `asignarVacaciones()`, que asigna y muestra los días de vacaciones. Esta interfaz asegura que cualquier clase que la implemente tendrá estos métodos disponibles.
2. **Clase Empleado:** Es una clase abstracta que actúa como base para los distintos tipos de empleados. Incluye atributos como `id`, `nombre`, `apellidoPaterno`, `apellidoMaterno`, `edad`, `telefono`, y `puesto`. También tiene un contador estático para generar identificadores únicos para cada empleado. Esta clase define métodos para obtener y establecer estos atributos, y contiene un método abstracto `calcularSalario()` que debe ser implementado por sus subclases para calcular el salario del empleado.

Las **clases específicas representan los tipos de empleados**, los cuales extienden la clase `Empleado` y proporcionan detalles particulares para cada tipo de empleado.

- **EmpleadoAsalariado** representa a un empleado con un salario fijo. Tiene atributos adicionales como `salarioBase` y `antiguedadAnios`, y proporciona implementaciones para calcular bonos y asignar vacaciones en función de la antigüedad. También calcula el salario total, incluyendo los bonos.
- **EmpleadoConComision** es un empleado que recibe un salario base más una comisión por ventas realizadas. Incluye atributos como `ventasRealizadas` y `porcentajeComision`. Esta clase calcula los bonos basados en el rendimiento de ventas y calcula el salario total considerando tanto el salario base, la comisión y los bonos.
- **EmpleadoPorHoras** representa a un empleado que trabaja por horas. Tiene atributos como `numHoras`, `tarifaPorHora` y `asistenciaDiaria`. La clase calcula el salario total según las horas trabajadas y la tarifa por hora, y también maneja los bonos y vacaciones.

Finalmente, en la **clase Principal** se crean instancias de diferentes tipos de empleados, se almacenan en una lista y se recorren para mostrar su información, calcular sus salarios y asignarles vacaciones.