Ejercicio Herencia

Aprendiz

Erick Santiago Noreña Cifuentes

Instructor

Julio Roberto Galvis Cardozo

SENA

Centro De Diseño Y Metrología

Bogotá

2024

Contenido

Introducción

El desarrollo de clases orientadas a objetos es fundamental para la organización y optimización de proyectos en Java. En este ejercicio, se implementaron tres clases: Persona, Estudiante, y Profesor, aplicando principios de herencia y encapsulamiento. La clase Persona actúa como la clase base, mientras que Estudiante y Profesor heredan de esta, agregando atributos y comportamientos específicos. El ejercicio ya ha sido completado en código Java, lo que permite un manejo eficiente de las propiedades comunes y diferenciadas de estas entidades.

Estructura del Ejercicio

El proyecto se ha desarrollado con una estructura jerárquica que refleja el diagrama de clases proporcionado:

1. **Persona:** Es la clase principal que contiene atributos comunes a estudiantes y profesores, como el nombre y la edad. Se implementaron métodos para obtener y modificar estos atributos de manera encapsulada, garantizando la integridad de los datos.
2. **Estudiante:** Es una subclase de Persona que añade características particulares de los estudiantes, como la carrera y el legajo. Al heredar de Persona, también cuenta con los atributos comunes, lo que optimiza la reutilización de código.
3. **Profesor:** Similar a Estudiante, esta clase hereda de Persona, pero incluye atributos específicos de los profesores, como la materia que enseñan y el cargo que ocupan. Esto asegura que cada tipo de usuario del sistema tenga sus propios atributos sin duplicar código innecesariamente.

Planteamiento

El ejercicio se enfocó en demostrar cómo la herencia en Java permite crear jerarquías de clases más flexibles y escalables. Se implementaron los constructores, así como los métodos getter y setter para cada atributo, respetando el principio de encapsulamiento. Este enfoque simplifica la manipulación de objetos en el sistema y mantiene la claridad en el código.

Por ejemplo, al crear un objeto de tipo Estudiante, no solo es posible acceder a los métodos específicos de esta clase (como getCarrera o setLegajo), sino también a los métodos heredados de Persona, como getNombre y setEdad. Lo mismo ocurre para la clase Profesor.

Conclusión

El ejercicio ha demostrado con éxito cómo el uso de la herencia y el encapsulamiento en Java optimiza la estructuración del código. La clase Persona proporciona una base sólida con atributos comunes, mientras que Estudiante y Profesor agregan sus propias particularidades, garantizando un sistema extensible y mantenible. Este diseño facilita la gestión de distintos tipos de entidades (personas, estudiantes, profesores) dentro del sistema, asegurando que el código sea reutilizable y fácilmente modificable para futuras ampliaciones.