Consigna: Sistema de Gestión Universitaria usando Herencia

Objetivo:

Desarrollar un programa en Java que modele una estructura básica de una universidad y sus facultades. Este ejercicio pondrá en práctica los conceptos de herencia y polimorfismo en la Programación Orientada a Objetos (POO).

Descripción del Ejercicio

Se pide a los estudiantes modelar un sistema de gestión universitaria que incluya la estructura de una Universidad y varias Facultades. Cada facultad puede tener diferentes características, pero todas deben compartir ciertos atributos y métodos comunes que pertenecen a la universidad.

Requisitos:

Crear una clase base llamada Universidad:

Atributos:

nombre (nombre de la universidad)

direccion (dirección de la universidad)

anioFundacion (año de fundación de la universidad)

Métodos:

Constructor para inicializar los atributos.

mostrarInformacion(): un método que muestre la información de la universidad.

cantidadEstudiantesTotal(): un método abstracto que se implementará en las facultades para calcular el número total de estudiantes en cada facultad.

Crear una clase Facultad que herede de Universidad:

Esta clase debe ser una clase abstracta que servirá como base para cada facultad específica.

Atributos:

nombreFacultad (nombre de la facultad)

cantidadEstudiantes (cantidad de estudiantes en la facultad)

Métodos:

Constructor que inicialice el nombre de la facultad y la cantidad de estudiantes.

Implementar el método cantidadEstudiantesTotal() para devolver el número de estudiantes en esa facultad.

mostrarInformacionFacultad(): muestra el nombre de la facultad y la cantidad de estudiantes.

Crear clases específicas que hereden de Facultad:

Ejemplos de Facultades:

FacultadIngenieria

FacultadCiencias

FacultadArtes

Cada clase de facultad debe:

Contar con su propio constructor.

Sobreescribir el método mostrarInformacionFacultad() para añadir detalles específicos de cada facultad.

Implementar el método cantidadEstudiantesTotal().

Clase Principal Main:

En la clase Main, crear una instancia de la universidad y varias instancias de cada facultad específica (por ejemplo, una facultad de ingeniería, otra de ciencias, etc.).

Llamar a mostrarInformacion() para ver la información de la universidad.

Llamar a mostrarInformacionFacultad() para mostrar la información de cada facultad.

Calcular el total de estudiantes en todas las facultades sumando los resultados de cantidadEstudiantesTotal().