Sistema de Gestión de Eventos

Desarrolla un sistema para gestionar eventos de una empresa. El sistema debe permitir crear diferentes tipos de eventos (conferencias, talleres), agregar asistentes, y mostrar información detallada de cada evento.

Requisitos:

Clases:

- **Evento:**
  - Atributos: Nombre, fecha de inicio, fecha de fin, lugar, capacidad máxima, lista de asistentes (vector de objetos Asistente).
  - **Métodos:**
    - Constructor para inicializar los atributos.
    - Métodos get y set para acceder y modificar los atributos.
    - Método para agregar un asistente a la lista.
    - Método para mostrar la información del evento (nombre, fecha, lugar, número de asistentes).
- **Conferencia:**
  - Hereda de Evento.
  - Atributos adicionales: Temas, ponentes (vector de objetos Persona).
  - Métodos adicionales:
    - Agregar un tema.
    - Agregar un ponente.
- **Taller:**
  - Hereda de Evento.
  - Atributos adicionales: Duración (en horas), materiales necesarios.
  - ***Métodos adicionales:***
    - Agregar un material.

- **Asistente:**
    - ***Atributos:*** Nombre, apellido, correo electrónico.
    - ***Métodos:***
        - Constructor para inicializar los atributos.
        - Métodos get y set para acceder y modificar los atributos.
- **Persona: (Clase auxiliar)**
    - ***Atributos***: Nombre, apellido.
    - ***Métodos***:
        - Constructor para inicializar los atributos.
        - Métodos get y set para acceder y modificar los atributos.

Funcionalidades:

- Crear eventos de tipo conferencia o taller.
- Agregar asistentes a un evento.
- Mostrar la información detallada de un evento, incluyendo la información específica de cada tipo (ponentes, materiales).
- Buscar eventos por nombre o fecha.

Consideraciones adicionales:

- ***Encapsulación***: Utilizar modificadores de acceso (private) para proteger los datos y controlar el acceso a ellos.
- ***Herencia***: Utilizar la herencia para modelar la relación entre Evento, Conferencia y Taller.
- ***Polimorfismo***: Sobrecargar el método mostrar Información() en las clases hijas para mostrar la información específica de cada tipo de evento.

  Manejo de fechas: Utilizar una clase de manejo de fechas (por ejemplo, java.util.Date o java.time.LocalDate) para representar las fechas de inicio y fin de los eventos.
- **Validación de datos:** Implementar validaciones para asegurar que los datos ingresados sean válidos (por ejemplo, que la fecha de inicio sea anterior a la fecha de fin).
-

Objetivos de aprendizaje:

- Aplicar los conceptos de POO: herencia, polimorfismo, encapsulación.
- Diseñar clases y métodos para resolver un problema real.
- Utilizar estructuras de datos como vectores.
- Implementar algoritmos de búsqueda y ordenamiento (opcional).

Este ejercicio ofrece una base sólida para que practiques los conceptos fundamentales de la programación orientada a objetos en Java.

Exitos …

Carrera 43 A # 34 – 155. Torre Norte, Almacentro. Oficina 701.

Medellín, Colombia