

## Sistema de gestión de eventos deportivos

Crear un sistema de gestión de eventos deportivos que permita almacenar y gestionar información sobre eventos, participantes y resultados.

1. Crea una interfaz llamada `Ganador` que declare los siguientes métodos:
  - `obtenerGanador()`: Método que devuelva un objeto de tipo `Participante` que represente al ganador del evento.
2. Crea una clase abstracta llamada `EventoDeportivo` que implemente la interfaz `Ganador` e incluya los siguientes atributos y métodos:

Atributos:

- `nombre` (String)
- `fecha` (utilizar clase más adecuada de `java.time`)
- `lugar` (String)
- `participantes` (ArrayList de objetos de tipo `Participante`)

Métodos:

- Constructor que reciba los atributos e inicialice los valores
- Getters y setters para cada atributo
- Método `inscribirParticipante(Participante participante)` que reciba un objeto de tipo `Participante` y lo añada a la lista de participantes
- Método `obtenerGanador()` que devuelva el objeto de tipo `Participante` que represente al ganador del evento. Este método debe ser implementado en las subclases.

3. Crea dos clases que hereden de la clase `EventoDeportivo`: `Carrera` y `TorneoDeFutbol`. Ambas clases deben implementar el método `obtenerGanador()`.

Para `Carrera`:

- Añade el atributo `distancia` (double) que represente la distancia de la carrera en kilómetros
- Modifica el constructor para incluir el nuevo atributo
- Implementa el método `obtenerGanador()` de forma que devuelva el participante con el menor tiempo registrado (asumiendo que cada participante tiene un atributo de tiempo registrado en segundos)

Para `TorneoDeFutbol`:

- Añade el atributo `equipos` (ArrayList de objetos de tipo `Equipo`)
- Modifica el constructor para incluir el nuevo atributo
- Implementa el método `obtenerGanador()` de forma que devuelva el equipo con la mayor cantidad de puntos acumulados (asumiendo que cada equipo tiene un atributo de puntos acumulados)

4. Crea una clase llamada `Participante` que contenga los siguientes atributos y métodos:

Atributos:

- nombre (String)
- apellido (String)
- edad (int)

Métodos:

- Constructor que reciba todos los atributos e inicialice los valores
- Getters y setters para cada atributo
- Método `toString()` que devuelva una cadena con la información del participante

5. Crea una clase llamada `Equipo` que contenga los siguientes atributos y métodos:

Atributos:

- nombre (String)
- jugadores (ArrayList de objetos de tipo `Participante`)
- puntos (int)

Métodos:

- Constructor que reciba el nombre del equipo e inicialice los valores (inicializar puntos en 0 y jugadores como un ArrayList vacío)
- Getters y setters para cada atributo
- Método `añadirJugador(Participante jugador)` que añada un jugador al equipo
- Método `eliminarJugador(Participante jugador)` que elimine un jugador del equipo por su nombre y apellido. Si el jugador no existe, lanza una excepción personalizada llamada `JugadorNoEncontradoException`.
- Método `toString()` que devuelva una cadena con la información del equipo, incluyendo los jugadores y los puntos acumulados

6. Crea una excepción personalizada llamada `JugadorNoEncontradoException` que extienda de `Exception` y tenga un mensaje de error personalizado. Además los participantes no podrán tener ningún atributo a nulo, de lo contrario se generará una excepción de tipo `ParticipanteNoValidoException`. Dichas instancias nunca deberán ser creadas. Controlar de la forma que se considere más adecuada que la edad de un `Participante` nunca podrá ser menor a 14.

7. Crea una clase principal `Main` que:

- Instancie eventos deportivos (carreras y torneos de fútbol)
- Añada participantes a los eventos
- Inscriba equipos en torneos de fútbol
- Añada y elimine jugadores de equipos, mostrando excepciones en caso de que un jugador no exista en el equipo
- Muestre información de los eventos, participantes, equipos y jugadores utilizando el método `toString()`
- Determine y muestre el ganador de cada evento utilizando el método `obtenerGanador()`