Programación Orientada a Objetos

Examen - 2do Trimestre

1º DAW - Programación

Abril 1, 2025

Descripción del ejercicio

El evento más esperado del año, *La Velada 5*, está a punto de celebrarse. Este evento reúne a los creadores de contenido más populares en combates de boxeo que son transmitidos en directo a millones de espectadores. Para gestionar la organización y el seguimiento del evento, los estudiantes del instituto ADAITS han decidido desarrollar un programa que permita administrar los combates, participantes y espectadores. Sus profesores les orientan con el diagrama de clases que se muestra en la figura 1.

En apoyo al diagrama, se explican los métodos que han de ser implementados:

- **Participante**:
  - `presentarse` Método abstracto que debe ser implementado en las subclases. Imprime la presentación del participante, incluyendo su nombre y sus datos.
    - *Boxeador*: "Soy , boxeador de y compito en ."
    - *Entrenador*: "Soy , el entrenador de ."
    - *Árbitro*: "Soy , el árbitro de la pelea."
- **Evento**:
  - constructor `Evento()` crea un evento con una pelea estelar.
  - constructor `Evento(Pelea peleaEstelar)` crea un evento con una pelea estelar específica.
  - `iniciarEvento` Inicia el evento llamando al método `iniciarCombate` de la pelea estelar y simula un combate entre los boxeadores. Se debe mostrar el nombre de los boxeadores y el resultado del combate (ganador o empate). Si el combate es detenido por el árbitro, se debe mostrar un mensaje indicando que la pelea fue detenida.
- **Pelea**:
  - `iniciarCombate` Marca el inicio de la pelea entre dos boxeadores.
  - `continuarCombate` Calcula el atacante del siguiente movimiento, invocando al método `registrarGolpe` de la pelea. Devuelve `true` si el defensor **no** ha sido derrotado, `false` en caso contrario. El atacante se calcula multiplicando la velocidad con un valor aleatorio, el boxeador que obtenga el valor más alto será el atacante.
  - `registrarGolpe` Registra un golpe de un boxeador atacante a un defensor. Se deben invocar los métodos `esquivarGolpe`, `recibirGolpe` del defensor y `getPotencia` del atacante. Si el defensor no equiva el golpe, recibirá el daño del atacante.
  - `determinarGanador` Devuelve el boxeador ganador de la pelea. El ganador será el boxeador que no haya sido derrotado. Si ninguno de los boxeadores ha sido derrotado, el ganador es el boxeador con más resistencia.
  - `finalizarCombate` Finaliza el combate y muestra al ganador.
- **Boxeador**:
  - `calcularCategoria` Calcula la categoría del boxeador en base a su peso. Las categorías son:
    - PESO_PLUMA: 57 kg o menos
    - PESO_LIGERO: entre 57 y 70 kg
    - PESO_MEDIO: entre 70 y 85 kg

    * PESO_PESADO: más de 85 kg
  - `recibirGolpe` Reduce la resistencia del boxeador al recibir un golpe. La cantidad de resistencia perdida es 10% de la potencia del golpe del atacante.
  - `estaDerrotado` Indica si el boxeador ha sido derrotado. Si la resistencia llega a 0, el boxeador es derrotado.
  - `esquivarGolpe` Determina si el boxeador esquiva un golpe basado en su velocidad y experiencia. La probabilidad de esquivar es el 40% de la velocidad más el 50% de la resistencia.
  - `mejorarEstadisticas` Incrementa la potencia y velocidad del boxeador entre 3% y 7% en base a su resistencia. No es aleatorio, el porcentaje es proporcional a la resistencia del boxeador. Si la resistencia es 100% el incremento es 7%, si es 0% el incremento es 3%. La resistencia nunca puede ser superior a 100.
  - `mostrarEstado` Muestra el estado del boxeador, incluyendo su nombre, velocidad, resistencia y potencia de golpe.
  - Métodos `get` para obtener las propiedades del boxeador como `potencia`, `resistencia`, `velocidad`, `peso` y `categoria`.
- **Entrenador**:
  - `motivarBoxeador` Invioca el método `mejorarEstadisticas` del boxeador.
- **Arbitro**:
  - `detenerPelea` Invoca el método `finalizarCombate` de la pelea.

Nota: Se pueden crear métodos adicionales para facilitar la implementación de los métodos requeridos. Se recomienda implementar el método `toString` en cada clase para facilitar la visualización de los objetos.

Apartados

1. Creación de clases y jerarquía [2 puntos]
2. Variables, constructores, métodos genéricos (getter/setter y toString) [1 punto]
3. Correcto uso de Enum [1 punto]
4. Métodos de clase Arbitro y Entrenador [1 punto]
5. Métodos de la clase Pelea [2 punto]
6. Métodos de la clase Boxeador [2 punto]
7. Simular una pelea en el método de la clase Evento [1 punto]

Diagrama

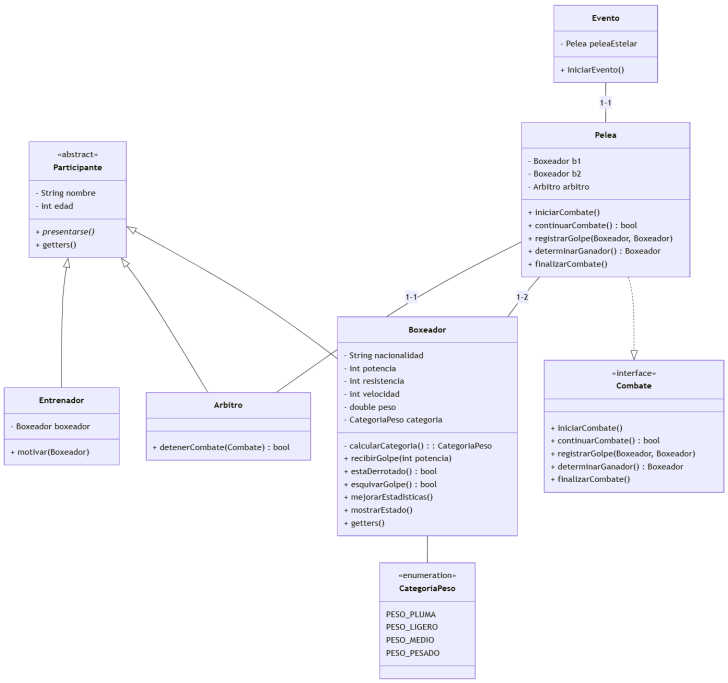

Figure 1: Diagrama de clases