

Prog. Orientada a Objetos

Carrera Programador full-stack

Ejercitacion + Repaso

Selección de Fútbol

Desarrolla un sistema de gestión para una selección de fútbol utilizando los principios de Programación Orientada a Objetos (POO) en TypeScript. Deberás implementar los siguientes elementos, relaciones y UML correspondiente:

Crea una *Clase Abstracta* llamada *Persona* con los siguientes atributos:

nombre: string (+)

edad: number (+)

Debe tener un método abstracto `obtenerDetalles(): string (+)` que devuelva una descripción de la persona.

Clases que heredan de Persona (Jugador, DirectorTecnico, MedicoDeportivo)

Clase Jugador:

Atributos adicionales:

posicion: string (+)

numero: number (+)

goles: number (-)

Implementa el método `obtenerDetalles() (+)` que devuelva una descripción del jugador.

Implementa una interfaz `IJugador` que contenga los métodos:

`anotarGol(): void (+)`

`obtenerGoles(): number (+)`

Selección de Fútbol

Clase DirectorTecnico:

Atributos adicionales:

nivelDeExperiencia: string (+)

Implementa el método obtenerDetalles() que devuelva una descripción del cuerpo técnico.

Clase MedicoDeportivo:

Atributos adicionales:

especialidad: string (+)

Implementa el método obtenerDetalles()(+) que devuelva una descripción del cuerpo médico.

Seleccion de Futbol

Clase Seleccion:

Crea una clase llamada Seleccion que contenga:

Un array de Jugador que representará la plantilla de jugadores. (+)

Un atributo cuerpoTecnico que sea una instancia de CuerpoTecnico. (+)

Un atributo cuerpoMedico que sea una instancia de CuerpoMedico. (+)

Debe tener métodos para:

agregarJugador(jugador: Jugador): void - Para agregar un jugador a la selección.

quitarJugador(numero: number): void - Para quitar un jugador de la selección usando su número.

editarJugador(numero: number, nuevosDatos: Partial<Jugador>): void - Para editar las características de un jugador.

listarJugadores(): void - Para mostrar la lista de jugadores y sus detalles.