

Ejercicio: 002__001

Refactoriza el código realizado en el ejercicio 001__012:

Todos los objetos usados en nuestro zoo (área, recinto, animal, encargado, enfermería...) deberán pasar a ser clases que definamos mediante function y luego instanciamos mediante new.

Añade todas las funciones de cada clase (por ejemplo en animal: modificarSalud, ejecutarCicloAnimal, alimentar... etc) al prototype de la clase, para no repetir las funciones en cada instancia de dicha clase

No olvides realizar este proceso con todas las clases que haya en nuestro Zoo.

Ejercicio: 002__002

1) Haciendo uso de funciones y new, realiza una "clase" Vikingo que almacene la información de un vikingo:

nombre
salud (0 - 1000)
potenciaAtaque (1 - 20)
valocidad (0 - 100)

2) Haz uso de prototype y añade un método .ataca(vikingo) a un vikingo para que ataque a su oponente.
el ataque quitara salud al vikingo atacado (la potencia de ataque del atacante)

3) Realiza una clase Batalla() que en su creación reciba dos vikingos.

Batalla tendrá un método iniciarPelea que realizará la pelea entre ambos vikingos.

Una batalla tendrá una serie de asaltos en los que:

atacará primero el que más valocidad tenga,
y quitará de salud al rival su potencia de ataque,
hasta que uno de los dos muera. (salud <= 0)

4) Crear la clase Arma() tenga un tipo: (espada/cuchillo...etc), una potencia (20 - 50) y un ataquesRestantes (0 -10).

5) Añade una propiedad armas a Vikingo para que pueda poseer varias armas para su batalla.

Añade el método addArma() para añadir armas a los vikingos,

6) Modifica la función ataca del vikingo, para que si tiene armas disponibles ataque con el arma más potente.

Cada vez que se use un arma, deberá restar uno a ataquesRestantes de ese arma.
Cuando el arma tenga 0 ataquesRestantes, el vikingo deberá abandonar el arma (añade la función abandonarArma al vikingo).

Ejercicio: 002__003

Partiendo del ejercicio anterior:

- 1) realiza una clase `soldadoVikingo` que herede de `vikingo` y tenga los métodos necesarios para la lucha
- 2) realiza una clase `ejército` que tenga un array de `soldadoVikingo`
- 3) realiza una clase `curanderoVikingo` que herede de `vikingo`, añade a la clase `ejército` un array de `curanderos` y añade el método `resucitar(vikingo)`, que permitirá recuperar la salud de los soldados muertos en combate. Los curanderos solo podrán resucitar 5 veces
- 4) añade un método `enfrentarEjercitos` en la clase `batalla`, que reciba dos ejércitos y los enfrente entre si.
Enfrentará el primer soldado de cada ejército contra el primer soldado del otro.
Cuando mueran vikingos si el ejército tiene curanderos, podrán revivir al vikingo.
Si no pueden revivirlo, deberá abandonar el array y pasara a enfrentarse el siguiente
La batalla acabará cuando uno de los dos ejércitos pierda todos sus soldados.

NOTA:

Parametros de los Vikingos:

Salud: 100

Potencia de ataque: aleatorio entre 1 y 20

Numero de armas: aleatorio entre 1 y 5

Parametros de armas:

NumeroDeAtaques: aleatorio entre 1 y 10

Potencia: aleatorio entre 20 y 50

Ejercito:

Numero de soldados: 25

Numero de curanderos: 5

Curanderos:

`vecesQuePuedeCurar`: 5 (restaura a un vikingo al 100% de salud)

Ejercicio 002__003 Bonus

1) Vamos a cambiar nuestra clase ejército.

La clase ejército deberá recibir un array de soldados (25) y un array de armas (50).

Durante su construcción el ejército deberá decidir cómo reparte las armas entre sus soldados.

También deberá ordenar el array de soldados decidiendo en que orden deben pelear.

2) crearemos dos ejércitos y después crearemos una batalla con esos dos ejércitos.

La clase batalla orquestará la batalla entre los dos ejércitos

Ejercicio: 002__004

1) Define una clase Persona que tenga los siguientes atributos:

Nombre:

Edad:

Nacionalidad:

Altura:

Peso:

Enfermo: true/false

2) Definir la clase Jugador que herede de persona y tenga los siguientes atributos:

Posición: (portero/defensa/mediocentro/delantero)

Numero:

Calidad: (0-100)

La posición de cada jugador es completamente aleatoria

El estar enfermo o no, es aleatorio (10% de probabilidad de estar enfermo)

3) Definir la clase Equipo que tenga:

- Array de jugadores

- Entrenador

Un equipo tendrá 22 jugadores creados aleatoriamente

4) Definir la clase Entrenador que herede de Persona y tenga los siguientes métodos:

- elegirPlantillaParaPartido() que elegirá de sus jugadores a los mejores para un partido:

 - 1 portero

 - 4 defensas

 - 4 mediocentros

 - 2 delanteros

5) Define la clase partido que se cree a partir de dos equipos.

La clase partido tendrá el método jugarPartido que hará que se dispute.

Lógica del partido:

Cada equipo hará 10 ataques que funcionarán de la siguiente manera

Por ejemplo:

Si ataca el equipo 1 se calculará:

$A = (\text{Suma de calidad de medio centros equipo 1}) - (\text{Suma de calidad de medio centros equipo 2})$

$B = (\text{Suma de calidad de delanteros 1}) - (\text{Suma de calidad de defensas equipo 2})$

$C = A + B - (\text{Suma de calidad de portero equipo 2})$

Fortuna = numero aleatorio entre 0 y 100

Para cada jugador que no esté en su puesto del equipo 1:

$C = C - 10$

Para cada jugador que no esté en su puesto del equipo 2:

$C = C + 10$

TOTAL = C + Fortuna

Si total es mayor que cero -> GOOOOOOOL

Si total es igual a cero -> PALO !!!

Si total es menor que cero -> Ná de ná

AL finalizar el partido deberá mostrarse el resultado