Ejercicio 1

En el primer ejercicio he creado el constructor de una clase Jugador con los parámetros exigidos, teniendo en cuenta que el máximo de puntos de vida es una propiedad fija y no se define por parámetro.

Mostramos los datos de los jugadores por consola mediante un método llamado toString() dentro de Jugador, y compruebo su funcionamiento tras crear dos jugadores de ejemplo.

Ejercicio 2

Teniendo ya el constructor de la clase Zombi, creamos una subclase suya: Abominacion. Con “extends” hacemos que la nueva subclase herede de Zombi y creamos un método con un nuevo ataque, que se define llamando al método “atacar” de la clase padre con “super” tres veces.

Ejercicio 3

Dada la subclase PlantaCurativa que hereda de Consumible, implemento el código faltante. En el constructor de la subclase llamamos al “nombre” de la clase padre y añadimos un nuevo parámetro “poder” que mide la potencia curativa de la planta. En el método “consumir” aplicamos los efectos de la planta curativa curando la vida correspondiente al jugador, respetando siempre su máximo de puntos de vida. Y finalmente comprobamos todo mostrándolo por consola.

Ejercicio 4

Con el array de objetos dado, creamos una función jugadoresTop(array) que, gracias al método filter(), nos devuelve un nuevo array sin modificar el que se pasa por parámetro con los nombres de aquellos jugadores que tengan más de 100 puntos. Siendo la función flecha que pasamos como parámetro a filter(), la parte más esencial del ejercicio, hace que se vea fácil y limpio. Finalmente mostramos por consola los elementos del nuevo array para comprobar el funcionamiento de la función.

GitHub

Enlace a mi repositorio privado con la práctica resuelta:

adrismm/DWC_ASMM_Practica1 (github.com)