

POKEMON



ANTES DE EMPEZAR:

Este ejercicio es para realizar de forma individual.

- ✓ Se valorará en la corrección que el programa esté probado (compila y ejecuta bien) y que esté claramente escrito y organizado (se respetan las reglas de estilo del lenguaje Java, nombres descriptivos, código no duplicado, ...). Leer atentamente la rúbrica.
- ✓ El ejercicio se realizará en IntelliJ.
- ✓ La entrega consistirá en comprimir la carpeta del proyecto en formato rar, zip o 7z y subirlo a la tarea correspondiente de la plataforma moodle.
- ✓ La fecha tope de entrega es el **viernes 22 noviembre hasta las 23:59h**. No se aceptarán entregas pasado el límite de entrega.
- ✓ Se anulará automáticamente la corrección del trabajo y se evaluará con un 0 si se detecta que ha sido copiado, dejado copiar a algún compañero/a o se han utilizado inteligencias artificiales.
- ✓ Se penalizará si no se siguen las normas de entrega del ejercicio
- ✓ El profesorado podrá convocar al alumno/a para defender oralmente el proyecto

Resumen del Proyecto

El proyecto es un sencillo juego de combate por turnos o rondas inspirado en Pokémon. Los jugadores crean su Pokémon indicando el nombre del mismo y el tipo (agua, fuego o tierra); seguidamente combatirán contra rivales y avanzan de nivel si ganan, hasta enfrentarse a un máximo de tres rivales. El juego, por lo tanto se divide en 3 combates o partidas, cada combate en rondas. En cada ronda, el Pokémon perdedor restará 1 a su aguante. El combate será ganado por el Pokémon que mantenga algún punto de aguante mayor a 0 mientras que su oponente se queda sin puntos de aguante. Si el Pokémon del jugador pierde todos sus puntos de aguante, pierde el combate, es eliminado y el juego termina; si gana el combate subirá de nivel y pasará al siguiente combate. Si el Pokémon jugador gana el tercer y último combate habrá ganado el juego y el jugador será proclamado **MAESTRO POKEMON**.

Clases en el Proyecto

1. **AppPokemon**: Es el punto de entrada del programa. Llama a la clase `InterfazPokemon` para iniciar el juego. (**Esta clase está hecha, no hay que modificarla**).
2. **InterfazPokemon**: Controla la lógica del juego, interactúa con el jugador para la creación de Pokémon y organiza los combates.
3. **Combate**: Define las reglas y mecánicas de una pelea entre dos Pokémon.
4. **Pokemon**: Representa a los Pokémon, incluyendo su nivel, tipo, aguante y lógica para calcular poder.

Especificaciones

1. Clases a implementar:

- Clase **Pokemon** (en esta clase sólo tienes que implementar los constructores y el método **calcularPoder**):

- **Atributos:**

- nombre (String): Nombre del Pokémon.
- tipo (String): Tipo del Pokémon (agua, fuego o tierra).
- nivel (int): Nivel actual del Pokémon.
- aguante (int): Resistencia del Pokémon (varía según nivel).

- **Métodos:**

- Constructor1: Inicializa un Pokémon con nombre, tipo ya especificar. El nivel es 1 cuando se crea el pokemon y el aguante se calculará según el nivel del pokemon, incluido en el constructor.
- Constructor2: Igual que el anterior pero con la diferencia de que en este se definirá también el nivel, para crear los Pokemon oponentes.
- **calcularPoder(Pokemon contrincante)**: Calcula y devuelve el poder de ataque contra otro Pokémon (contrincante) según reglas específicas. El poder se calculará según un número aleatorio entero dentro de los siguientes rangos:

NIVEL	RANGO
1	3 - 10
2	6 - 20
3	9 - 30
4	12 - 40

Según esta tabla se calculará el poder **únicamente** el método **random**, siendo capaz de calcular cualquier poder mayor a 3. Además, una vez calculado el poder, este puede verse alterado según el tipo de pokemon del objeto y el pokemon rival según la siguiente tabla:

Tipo Pokemon objeto (this)	Tipo Pokemon Rival	Alteración del poder
Agua	Fuego	El poder aumenta 2 veces el nivel del propio objeto pokemon (this)
Agua	Tierra	El poder disminuye 2 veces el nivel del oponente.
Fuego	Agua	El poder disminuye 2 veces el nivel del oponente.
Fuego	Tierra	El poder aumenta 2 veces el nivel del propio objeto pokemon (this)
Tierra	Agua	El poder aumenta 2 veces el nivel del propio objeto pokemon (this)
Tierra	Fuego	El poder disminuye 2 veces el nivel del oponente.

Según lo visto en esta tabla se puede concluir:

- ✓ Agua vence a fuego.
- ✓ Fuego vence a tierra.
- ✓ Tierra vence a agua.

- subirNivel(): Incrementa el nivel del Pokémon y actualiza sus estadísticas.
- disminuirAguante(): Reduce en 1 el aguante del Pokémon.
- actualizarStats(): Recalcula el aguante basado en el nivel.
- toString(): Devuelve información detallada del Pokémon.

- **Clase Combate (deberás realizarla completa):**

- **Atributos:**

- pokemonJugador (Pokemon): Pokémon del jugador.
- pokemonRival (Pokemon): Pokémon del rival.

- **Métodos:**

- Constructor: Recibe el Pokémon del jugador y el rival.
- Ronda(): Resuelve una ronda, calculando el poder de ambos Pokémon y disminuyendo el aguante del perdedor (el que obtenga menos poder calculado). Devuelve el pokemon ganador de la ronda, si hay empate (ambos poderes resultan iguales), devuelve null.
- Ganador(): Devuelve el Pokémon ganador del combate o null si el combate aún no ha terminado. El ganador del combate será el Pokemon que tenga aguante mayor a 0 y el aguante del oponente haya llegado a 0.

- **Clase InterfazPokemon** (En esta clase deberás implementar las clases **Juego**, **Partida**, **menuCreacionPokemonJugador** y **siguientePokemonRival**):

- **Atributos:**

- teclado (Scanner): Para leer entradas del jugador.

- **Métodos:**

- **Juego():** Orquesta las partidas, permitiendo al jugador avanzar hasta 3 rivales o perder, en el segundo caso quedará eliminado. Se deberá tener en cuenta que si pierde la última ronda el jugador también queda eliminado, es decir, el jugador ganará el juego si gana los tres combates. **Cada vez que el pokemon jugador gane un combate subirá de nivel.**
 - **Partida(Pokemon PokemonJugador, Pokemon PokemonRival):** Este método gestiona una partida entre el Pokemon jugador y el pokemon rival. Se creará un combate entre ambos e irán haciendo rondas hasta que uno de ellos se quede sin aguante (aguante llegue a 0). Recordar que en cada ronda, se calculan los poderes de cada Pokemon y el pokemon que menos poder obtenga se le restará uno a su aguante. El pokemon que salga victorioso (su aguante no haya llegado a 0 y el de su oponente sí) será el objeto que devuelva este método.
 - **menuCreacionPokemonJugador():** Permite al jugador crear su Pokémon inicial. Para ello tendrá que darle un nombre y un tipo entre agua, tierra y fuego. Devuelve el Pokemon del jugador creado.
 - **siguientePokemonRival(int numero):** Devuelve un Pokémon rival según la ronda actual. Dicha relación se da en la siguiente tabla:

Int numero (el 1 representa el primer combate, el dos el segundo combate...)	POKEMON RIVAL		
	Nombre	Tipo	Nivel
1	Caterpie	tierra	1
2	Bulbasur	agua	2
3	Charmander	fuego	3

- **mostrarJuegoSuperado(), mostrarFinPartida():** Mensajes para el final del juego. Estos dos métodos ya están completados, no debes modificarlos.

- **Clase AppPokemon (Está completa, no hay que modificarla):**
 - **Métodos:**
 - `main(String[] args)`: Punto de entrada del programa.