

MANUAL TECNICO

PRACTICA 2

INTRODUCCION

Objetivo:

Otorgar soporte al usuario que comprenda y entienda cada uno de los métodos creados para el funcionamiento del programa.

Definiciones:

PRACTICA 2: es un programa que permite la interacción del usuario como si fuera una consola de Pokémon y sus diferentes menús y actividades.

CLASES

El presente manual esta creado con fines de que el usuario reconozca cada clase, método y librería utilizada en la creación del programa como así mismo saber cuál es su función (requisito obligatorio tener instalado java en el equipo para poder ejecutarlo).

- Clase "**Pokemones**"
Esta clase contiene todos los parámetros asignados para su uso en específico y también posee un método constructor como identificador con el de la clase correspondiente con el objetivo de obligar a y controlar cómo se inicializa una instancia de la clase y al final cuenta con sus respectivos métodos set y get.
- Clase "**Entrenadores**"
Esta clase contiene todos los parámetros asignados para su uso en específico y también posee un método constructor como identificador con el de la clase correspondiente con el objetivo de obligar a y controlar cómo se inicializa una instancia de la clase y al final cuenta con sus respectivos métodos set y get.
- Clase "**Pokeball**"
Esta clase contiene todos los parámetros asignados para su uso en específico y también posee un método constructor como identificador con el de la clase correspondiente con el objetivo de obligar a y controlar cómo se inicializa una instancia de la clase y al final cuenta con sus respectivos métodos set y get.
- Clase "**Gimnasios**"
Esta clase contiene todos los parámetros asignados para su uso en específico y también posee un método constructor como identificador con el de la clase correspondiente con el objetivo de obligar a y controlar cómo se inicializa una instancia de la clase y al final cuenta con sus respectivos métodos set y get.
- Clase "**Alimentos**"
Esta clase contiene todos los parámetros asignados para su uso en específico y también posee un método constructor como identificador con el de la clase correspondiente con el objetivo de obligar a y controlar cómo

se inicializa una instancia de la clase y al final cuenta con sus respectivos métodos set y get.

- Clase "[Pelea](#)"
Esta clase contiene todos los parámetros asignados para su uso en específico y también posee un método constructor como identificador con el de la clase correspondiente con el objetivo de obligar a y controlar cómo se inicializa una instancia de la clase y al final cuenta con sus respectivos métodos set y get.
- Clase "[MAIN](#)"
Esta clase contiene al método principal a ejecutarte y la programación de los diferentes métodos que se utilizaron para el funcionamiento correcto del programa.

METODOS

- Método "[Cargar_Pokemones](#)"
Método que realiza la lectura de un archivo csv y los datos que tiene el archivo los guarda en un objeto creado con capacidad máxima de 150 Pokemons.
- Método "[Cargar_Entrenadores](#)"
Método que realiza la lectura de un archivo csv y los datos que tiene el archivo los guarda en un objeto creado con capacidad máxima de 25 Pokemons.
- Método "[Pokeballs](#)"
Método que realiza la lectura de un archivo csv y los datos que tiene el archivo los guarda en un objeto creado con capacidad máxima de 4 Pokemons.

- Método "[Cargar_Gym](#)"
Método que realiza la lectura de un archivo csv y los datos que tiene el archivo los guarda en un objeto creado con capacidad máxima de 25 Pokemons.
- Método "[Cargar_Alimentos](#)"
Método que realiza la lectura de un archivo csv y los datos que tiene el archivo los guarda en un objeto creado con capacidad máxima de 25 Pokemons.
- Método "[Asignar_pokemons](#)"
Método que realiza la lectura de un archivo csv y los datos que tiene el archivo y realiza un proceso en el cual asigna una pokeball a un Pokémon realizando la conexión entre las id de cada uno.
- Método "[Asignar_pelea](#)"
Método que realiza simula la batalla entre dos pokemones y gana aquel que sea mayor vida al terminar tres rounds o si la vida de uno de ellos es cero.
- Método "[Asignar_pokeball](#)"
Método que realiza la asignación de una pokeball según su entrenador.
- Método "[Asignar_comida](#)"
Método que realiza la lectura de un archivo csv y los datos que tiene el archivo y realiza el proceso de asignación de comida al Pokémon sumándole la vida que ofrece la comida a la vida actual del Pokémon.
- Método "[ORDENARPOKEMONS](#)"
Método que realiza ordena el objeto pokemon según sus puntos de ataque de mayor a menor utilizando el método bubble sort.
- Método "[ORDENARCOMIDA](#)"
Método que realiza ordena el objeto comida según la vida que otorga de mayor a menor utilizando el método bubble sort.
- Método "[Reportes](#)"

Método que realiza la verificación de las diferentes rutas y así valida que se puedan imprimir los diferentes reportes llamados a los métodos asignados para cada reporte.

- Método " **TOPCOMIDAS** "
Método que utiliza una función para generar un archivo y una función que se utiliza como lápiz para generar el reporte con todos los atributos correspondientes.
- Método " **TOPPOKEMONS** "
Método que utiliza una función para generar un archivo y una función que se utiliza como lápiz para generar el reporte con todos los atributos correspondientes.
- Método " **POKEMONSALVAJE** "
Método que utiliza una función para generar un archivo y una función que se utiliza como lápiz para generar el reporte con todos los atributos correspondientes.
- Método " **COMIDAASIGNADA** "
Método que utiliza una función para generar un archivo y una función que se utiliza como lápiz para generar el reporte con todos los atributos correspondientes.
- Método " **main** "
Método que contiene toda la interfaz del menú y es lo primero que ejecuta el programa al iniciar.

LIBRERIAS

- java.io. BufferedReader
Es una clase de Java para leer el texto de una secuencia de entrada (como un archivo) almacenando en el búfer caracteres que leen a la perfección caracteres, matrices o líneas.

- `java.io. File`
Es una clase de java que nos permite crear, eliminar archivos, directorios o bien para obtener información sobre archivos y directorios.
- `java.io.FileReader`
Es una clase de java que nos permiten leer y escribir "streams" de datos y de archivos bien para abrir un fichero de texto para leer usando la clase `FileReader`.
- `java.io.FileWriter`
Esta clase es usada para escribir caracteres en archivos. Su método `write()` permite escribir caracteres o strings a un fichero.
- `java.io.PrintWriter`
Es una clase que sirve para crear y escribir archivos en Java que soporta diferentes tipos de datos para imprimir desde los más primitivos como números, texto hasta arrays y objetos.
- `java.util.InputMismatchException`
Es una excepción lanzada por la clase `Scanner` cuando el elemento recibido no corresponde al tipo de dato esperado.
- `java.util.Scanner`
es una clase utilizada para obtener la entrada de los tipos como `int`, `double`, `string` etc.
- `java.time.ZonedDateTime`
es una clase que nos permite representar una fecha y hora con una zona horaria
- `java.time.format.DateTimeFormatter`
Este se utiliza para definir el tipo de formato que se maneja (dd/MMM/yy)

GLOSARIO

1. **Pokémon:** es una franquicia de medios que originalmente comenzó como un videojuego RPG, pero debido a su popularidad ha logrado expandirse a otros medios de entretenimiento como series de televisión, películas, juegos de cartas, ropa, entre otros, convirtiéndose en una marca que es reconocida en el mercado mundial.
2. **Pokeball:** un artilugio u objeto diseñado para servir dos funciones básicas en el mundo Pokémon, almacenar y transportar Pokémon, haciendo posible la captura de Pokémon.
3. **Serializar:** en una secuencia de bytes que pueden ser posteriormente leídos para restaurar el objeto original
4. **Clase en java:** es una plantilla que define la forma de un objeto; en ella se agrupan datos y métodos que operarán sobre esos datos.
5. **Poo:** es un paradigma de programación que viene a innovar la forma de obtener resultados.
6. **Encapsular:** se denomina encapsulamiento al ocultamiento del estado, es decir, de los datos miembro de un objeto de manera que solo se pueda cambiar mediante las operaciones definidas para ese objeto.
7. **Método get:** método que se emplea en las clases para mostrar el valor de un atributo.
8. **Método set:** método que se emplea en las clases para modificar el valor de un atributo
9. **Archivo Binario:** es un archivo informático que contiene información de cualquier tipo codificada en binario para el propósito de almacenamiento y procesamiento en ordenadores.
10. **Atributo:** también llamados datos o variables miembro son porciones de información que un objeto posee o conoce de sí mismo.