Hoja de trabajo No. 6

Realizar: Operaciones con mapas.
Realizarse: INDIVIDUAL

Objetivos:
a. **Utilización de Java Collection Framework**: uso de la interface MAP y sus (tres) implementaciones.
b. Uso de algoritmos polimórficos proporcionados por el Java Collection Framework.
c. Explorar otras colecciones del Java Collection Framework.
d. Pruebas unitarias.

(Imagen tomada de Supergeek.cl)

Programa a realizar:

El programa lee un archivo que contiene la información de Pokemons. La descripción completa del archivo y sus datos los puede ver en:
https://www.kaggle.com/datasets/mohitbansal31s/pokemon-dataset/data

La información de cada pokemon debe guardarse en un Map. La llave será **Name** (la columna que tiene el nombre del pokemon).

Utilice el patrón de diseño Factory para seleccionar la implementación de MAP que usará su programa, en tiempo de corrida. El usuario debe seleccionar entre: 1)HashMap, 2)TreeMap, 3)LinkedHashMap.

El usuario podrá guardar una colección personalizada de Pokemon, usted debe seleccionar una implementación para almacenar esta información en disco, seleccione la estructura de datos más adecuada.

Luego de leer el archivo su programa debe permitir al usuario realizar las siguientes operaciones:

1. Agregar un pokemon a la colección del usuario. Para esto el usuario ingresa el nombre del pokemon que desea agregar a la misma. NOTA: si ese pokemon ya esta en la colección del usuario ya no se agrega y se le informa que esta tratando de repetir un pokemon. NOTA: Si el usuario solicita un pokemon que no se encuentra en los datos leídos, el programa debe mostrar un error.
2. Mostrar los datos de un pokemon. El usuario ingresará el nombre del pokemon y se muestran sus datos (ver la descripción del archivo, en donde se indican todos los datos que tiene un pokemon)
3. Mostrar el nombre, tipo1 (Type1: tipo primario del pokemon) que el usuario tiene en su colección, ordenadas por tipo1.
4. Mostrar el nombre y tipo1 de todos los pokemon existentes (que fueron leídos del archivo de entrada). Deben estar ordenadas por tipo1
5. Mostrar el nombre del pokemon que tiene la habilidad indicada por el usuario. El usuario ingresará la habilidad deseada y su programa mostrará todos los pokemon existentes (que fueron leídos del archivo de entrada) que poseen dicha habilidad.

Tareas:

a. Su programa principal debe usar **Patron de diseño Factory** para seleccionar la implementación de MAP a utilizar.

b. Debe dejar evidencia de todo el desarrollo en el repositorio de github o sistema similar para control de versiones. Indicar como acceder a su repositorio y si es necesario, agregar a su catedrático y auxiliar para que tengan acceso al mismo.

c. Pruebas unitarias: haga por lo menos dos pruebas unitarias. Usted puede seleccionar a cuales operaciones de su programa les hará las pruebas unitarias.

d. Calcule la complejidad de tiempo para la operación #4: mostrar el nombre y tipo1 de todos los pokemon, ordenados por tipo1. Cómo llega a ese resultado. (Recordar que hay tres posibles implementaciones del Map, puede enfocarse en una de ellas en particular). Utilizando un “profiler” e indique cual es la mejor estructura para este problema.

Debe subir a Canvas todos los productos elaborados y los enlaces a su repositorio de github.

Calificación:

Aspecto	Puntos
Uso del repositorio: existen más de tres versiones guardadas, la última versión es igual a la colocada en Canvas.	5
Por lo menos dos pruebas unitarias por cada método	15
Patron Factory para seleccionar la implementación a usar en tiempo de corrida	15
Funcionamiento del programa para las operaciones 1, 2 y 3	25
Funcionamiento del programa para las operaciones 4 y 5	25
Explicación de porque se seleccionó la colección del JCF para guardar los pokemons del usuario (complejidad, operaciones soportadas, etc.)	5
Cálculo de la complejidad para de la operación #4 (pokemons ordenados por tipo1)	10
TOTAL:	100