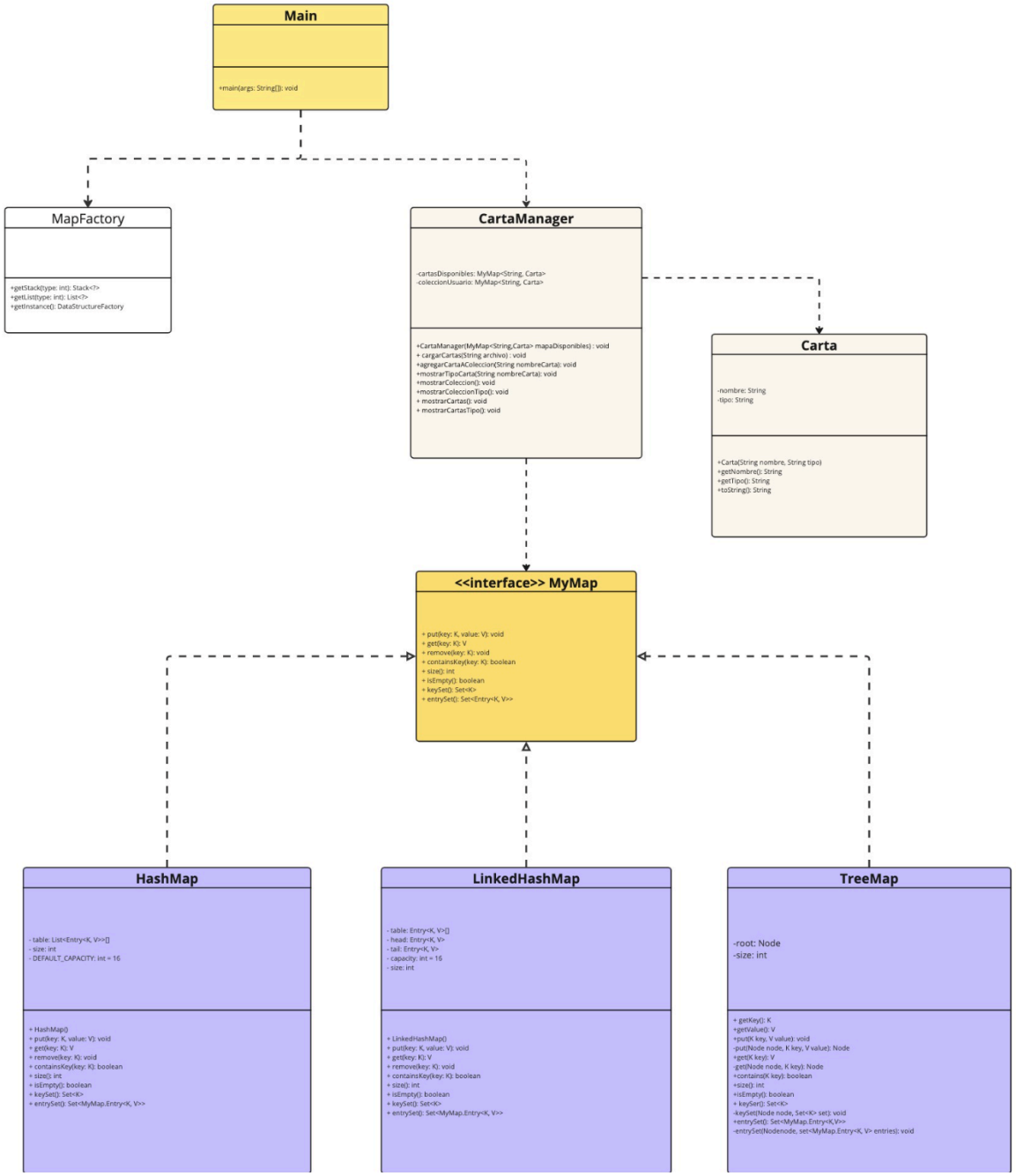
Guatemala 11 de marzo de 2024
Departamento de Ciencias de la Computación
Algoritmos y Estructura de Datos
Facultad de Ingeniería
Vianka Castro - 23201
Ricardo Godinez - 23247

Hoja de Trabajo 6

Operaciones con Mapas

UML CLASES



Se solicitó utilizar un profiler para evaluar el tiempo de ejecución del programa que muestra las cartas, utilizando las implementaciones de HashMap, LinkedHashMap y TreeMap. Sin embargo, debido a que nuestras cuentas gratuitas de servicios de profiler expiraron, no pudimos realizar esta tarea. En su lugar, realizamos una investigación sobre las características y comportamientos de las tres implementaciones de Mapa.

HashMap

Características:

- Almacena los pares clave/valor en una tabla hash sin ningún orden específico.
- Ofrece un rendimiento de tiempo constante (O(1)) en operaciones básicas como get y put.
- Sin embargo en nuestro programa sería O(n) en el peor de los casos, donde n es el número total de cartas en el mapa. Esto se debe a la necesidad de acceder a cada elemento en el mapa y luego iterar sobre todos los elementos para mostrar las cartas.
- No mantiene un orden específico de inserción.

Consideraciones:

- Ideal si el rendimiento de tiempo constante es prioritario y el orden de inserción no es importante.
- Requiere controlar la carga para evitar una degradación del rendimiento.
- Aplicabilidad al problema:
- Podría ser adecuado si el tamaño de la colección de cartas es conocido y no se espera que varíe significativamente.
- La ausencia de un orden específico podría no ser ideal para mantener una relación coherente entre las cartas y sus tipos en el contexto de mostrar la colección del usuario.

LinkedHashMap

Características:

- Combina las características de un HashMap con una lista vinculada para mantener el orden de inserción.
- Proporciona un rendimiento similar a un HashMap con la ventaja adicional de mantener el orden de inserción.

Consideraciones:

- Útil cuando se requiere acceso rápido a los elementos y mantener un orden específico de inserción.
- Puede ofrecer un buen equilibrio entre rendimiento y funcionalidad.
- Aplicabilidad al problema:
- Sería una opción adecuada si se desea mantener el orden de inserción de las cartas leídas del archivo.
- Permite cumplir con la necesidad de mantener una relación coherente entre las cartas y sus tipos al mostrar la colección del usuario.

TreeMap

Características:

- Implementado como un árbol Rojo-Negro, que garantiza un ordenamiento automático de las claves.
- Ofrece un rendimiento logarítmico (O(log n)) en operaciones como contains, get o put.
-

Consideraciones:

- Útil cuando se necesita un orden específico de las claves y un rendimiento predecible en operaciones de búsqueda.
- Es más eficiente en términos de rendimiento cuando se espera una gran colección de elementos.

Decisión Final

Después de analizar las características y los requisitos de nuestro programa, decidimos utilizar la implementación de LinkedHashMap, ya que mantiene el orden de inserción de las cartas, lo que puede ser útil para mostrar las cartas en el orden en que se agregaron. Ofrece un equilibrio entre rendimiento y funcionalidad al proporcionar un acceso rápido a los elementos y mantener el orden de inserción, lo que facilita la coherencia entre las cartas y sus tipos al mostrar la colección del usuario.

Además, dado que no se ha especificado la necesidad de un rendimiento logarítmico o de un ordenamiento automático de las claves, un LinkedHashMap satisfaría los requisitos del problema de manera eficiente. Sin embargo, se reconoce que cada implementación de mapa tiene sus propias ventajas y desventajas en términos de rendimiento y comportamiento, y la elección final depende de los requisitos específicos del proyecto y las características del conjunto de datos.

Valdes, F. (2019, 8 noviembre). *Mapas Java: TreeMap vs HashMap vs LinkedHashMap*. https://es.linkedin.com/pulse/mapas-java-treemap-vs-hashmap-linkedhashmap-fernando-valdes