Práctica 7. Tablas de dispersión abierta. Operaciones básicas.

Objetivos

- Practicar con la implementación de tablas de dispersión.
- Practicar el uso de las operaciones básicas de las listas.
- Analizar diferentes alternativas para la obtención de códigos de dispersión (función `hash()`).
- Practicar el uso de las operaciones básicas de las tablas de dispersión.

Introducción

En una tabla de dispersión abierta, cada elemento de la tabla es una lista que permite almacenar todos los elementos que comparten un mismo código de dispersión.

Como lista podría utilizarse cualquiera de las implementaciones vistas en teoría: basada en array, simplemente enlazada o doblemente enlazada.

En esta práctica utilizaremos la lista doblemente enlazada implementada en la práctica anterior.

Desarrollo

Implementación de las operaciones de acceso posicional

Se propone implementar una tabla de dispersión abierta que utilice como lista la desarrollada en la práctica anterior.

La tabla se implementará como una clase Java genérica con dos parámetros:

```
class TablaDispersion<K,V>
```

El primero de los parámetros corresponde a la clase que se desea utilizar como *llave* y el segundo a la clase que constituye los *valores* a almacenar en la tabla.

La tabla de dispersión deberá tener las siguientes operaciones[1]:

- Constructor: construye una tabla vacía con el tamaño de la tabla (número de listas) que se indica como parámetro.
- `V put(K key, V value)`: asocia la llave con el valor indicado. Retorna el antiguo valor asociado con esa llave o `null` si no existía ninguna asociación para ella.
- `V get(K key)`: retorna el valor asociado con la llave, o null si la tabla no contiene ninguna asociación para la llave indicada.
- `V remove(K key)`: elimina de la tabla el valor asociado con la llave indicada. Retorna el antiguo valor asociado con esa llave o `null` si no existe ninguna asociación para ella.
- `void clear()`: vacía la tabla (elimina todas las asociaciones).

Indicar la eficiencia temporal de cada una de las operaciones implementadas.

1. Para una definición más completa de las operaciones se puede ver la proporcionada en el API de Java (documentación de la clase `HashMap`)

Función hash

Para obtener el código de dispersión de la llave puede utilizarse el método `hashCode()` definido en la clase `Object` (y, por consiguiente, heredado por todas las clases Java).

Muchas clases Java sobreescriben el método `hashCode()`. Lee en el API de Java como se define el método `hashCode()` para las clases `Object`, `Integer`, `Double` y `String`.

Para las clases `Integer` y `String` compara la función *hash* utilizada por Java con las vistas en clase de teoría.

Programa de prueba

Escribir un programa que permita verificar el correcto funcionamiento de la tabla.

El programa deberá crear una tabla y aplicar sobre ella los métodos desarrollados comprobando a cada paso que el estado de la tabla es el esperado de acuerdo a la especificación de los métodos.

Criterios de Evaluación y Aclaraciones

La práctica se considerará realizada si se implementan correctamente los métodos indicados y se realiza un programa que pruebe todos los métodos desarrollados en un amplio abanico de situaciones posibles.

También es necesario responder a las cuestiones planteadas: eficiencia de los métodos y comparación de la funciones *hash* de `Integer` y `String` con las vistas en teoría.

La práctica se entregará a través de la plataforma moodle siguiendo las instrucciones en ella proporcionadas.