

Práctica 4: Mapas

Ejercicio 1: Implementar el método `offset` de las clases `HashMapLP` (prueba lineal), `HashMapQP` (prueba cuadrática) y `HashMapDH` (doble hash), de manera que implementen la funcionalidad de prueba lineal, prueba cuadrática y doble hash, respectivamente.

Ejercicio 2: Implementar todos los métodos de las clases `HashMapSC` (resolución de colisiones por encadenamiento separado) así como todas las clases auxiliares necesarias para tener una funcionalidad equivalente a `HashMapLP`, `HashMapQP`, o `HashMapDH`. Para que el ejercicio se considere todas las clases deben tener los mismos métodos y el mismo comportamiento.

Ejercicio 3: Definir los test unitarios necesarios para asegurar el correcto funcionamiento de la clase `HashMapSC`.

Ejercicio 4: Pear es una conocida multinacional tecnológica que fabrica diferentes productos, especialmente reconocidos por la calidad de su servicio técnico. En concreto, desde hace varios años, Pear tiene varios productos estrella: el teléfono `yFone`, el ordenador de sobremesa `yMak`, y el `Makbuk`, la versión portátil de `yMak`. Se desea llevar un registro de los dispositivos que han comprado los usuarios a lo largo del tiempo en las `PearStore` de España, además de obtener cierto feedback de la valoración que hacen los clientes de cada uno de los productos. En concreto, desde Pear nos piden que desarrollemos un programa en Java que gestione los siguientes aspectos:

1. Leer de un fichero los datos sobre los productos que se han vendido a lo largo del tiempo, cuyo formato se describe al final del documento. Para ello, se implementará el método cuyo prototipo será:

```
public void loadFile(String pathToFile)
```

2. Llevar, para cada producto, un registro de qué `PearStore`'s lo han vendido y cuántas unidades de este han logrado vender. Como los productos llevan varios años siendo tendencia y se venden bastante bien, Pear fabrica uno de estos productos cada año, por lo que un dispositivo estará identificado por su nombre y su año de fabricación (es decir, puede haber un `yFon 2019` y un `yFon 2020` que no se contabilizarán conjuntamente). El registro debe tener en cuenta este detalle. Se deben implementar los métodos:
 - a. `public void addProduct(Product producto, Iterable<PearStore> listOfStores)`, que recibirá un producto y las `PearStore` en las que se ha vendido y añadirá la información al registro de productos.
 - b. `public void addSalesInPearStore(Product producto, PearStore store, int units, double score)`, que añadirá el número de unidades vendidas de un producto en un determinado `PearStore` y la puntuación (entre 0 y 5) que han asociado al producto los

clientes de esa tienda. Además, es importante tener en cuenta esta tienda en el registro de ventas del producto determinado.

3. Implementar el método

```
public double getScoreOfProduct(Product producto)
```

que devolverá la puntuación media asignada a un producto concreto por los clientes de todas las tiendas en las que se vendió.

4. Para realizar una mejor distribución del stock, Pear quiere saber en qué PearStore's se han vendido más unidades de un determinado producto. Para ello, se debe implementar el método:

```
public PearStore getGreatestSeller(Product producto)
```

que devolverá la PearStore con más unidades vendidas del product solicitado.

5. Se quiere saber las unidades totales que se han vendido de un determinado producto, para lo cual debe implementarse el método

```
public int getUnits(Product producto)
```

que devolverá el número de unidades vendidas del producto en el total de las tiendas en las que ha sido vendido.

6. Además, se quiere saber si ya existe un determinado producto registrado en la plataforma. Para ello, se debe implementar el método

```
public boolean productExists(Product product)
```

El formato del fichero que contendrá los datos será de la siguiente forma:

```
N
NombreProducto Año T
Tienda1 ID1 Unidades1 Score1
Tienda2 ID2 Unidades2 Score2
.
.
.
TiendaN IDN UnidadesN ScoreN
```

Donde:

-N indicará el número de productos que se describirán

-`NombreProducto` indica el nombre del producto para el que se deben registrar las ventas, `Año` será el año de fabricación del producto en cuestión y `T` el número de tiendas que vendrán a continuación.

-`TiendaX` es el nombre de la tienda (una tienda se identifica por su nombre y un código numérico), `IDX` será el identificador numérico de la tienda, `UnidadesX` será el número de unidades vendidas en esa tienda y `ScoreX` será la puntuación asignada por los clientes de esa tienda a ese producto.