

Hoja de trabajo No. 6

Realizar: Operaciones con mapas.

Realizarse: Individual

Objetivos:

- Utilización de Java Collection Framework: uso de la interface MAP y sus (tres) implementaciones.
- Uso de algoritmos polimórficos proporcionados por el Java Collection Framework.
- Control de versiones del programa.

Programa a realizar:

Utilice el 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.

Se cuenta con una tienda online que vende sus productos en dos distintos mercados, al por mayor y al por menor. Lo han contratado para que usted y su equipo de trabajo puedan proponer la implementación de un programa que permita leer un archivo con la cantidad de productos que se tiene cargados en un inventario denominado inventario.txt; este archivo contiene clasificado los productos por las siguientes categorías:

Categoría	Descripción del producto
Mueble de terraza	Mesas de jardín
	Sillas de jardín
	Conjuntos mesas y sillas de jardín
	Mesas de Ping Pong exteriores
Sillones de masaje	Cojines y colchonetas de masaje
	Sillones relax y sofás de masajes
	Sillones de masajes avanzados
	Sofás camas
Bebidas	Cerveza tibetana Barley
	Té frio
	Coca cola 1 litro
	Coca cola 2 litros
Condimentos	Sirope de regaliz
	Especies Cajun del chef
	Mezcla Gumbo del chef
Frutas	Peras secas
	Pasas
	Manzana roja
	Manzana verde
Carnes	Res
	Pollo
	Cerdo
	Camarones
	Pescados
Lácteos	Queso de cabra
	Queso Manchego
	Leche descremada
	Leche deslactosada
	Leche entera

La estructura del archivo es la siguiente: Cada producto tiene asociado la categoría del producto, por lo que, usted puede visualizar la información de la siguiente manera:

Categoría | Producto

Bebidas | Té frio

El archivo del inventario se encuentra adjunto en el CANVAS.

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

1. Agregar un producto a la colección del usuario. Para esto el usuario ingresa la categoría del producto que desea agregar a la misma. NOTA: El usuario puede agregar más de un artículo de cada categoría. NOTA: Si el usuario ingresa un artículo que no se encuentra entre las categorías disponibles el programa debe mostrar un error.
2. Mostrar la categoría del producto. El usuario ingresará el nombre del producto y se muestra la categoría del artículo.
3. Mostrar los datos del producto, categoría y la cantidad de cada artículo que el usuario tiene en su colección.
4. Mostrar los datos del producto, categoría y la cantidad de cada artículo que el usuario tiene en su colección, ordenadas por tipo.
5. Mostrar el producto y la categoría de todo el inventario.
6. Mostrar el producto y la categoría existentes, ordenadas por tipo.

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. Use un profiler para evaluar el tiempo de ejecución de su programa para mostrar las cartas. Corra su programa con las tres implementaciones y muestre los tiempos de ejecución de cada una de ellas. Diga cuál es la más rápida con el profiler.
- d. Calcule la complejidad de tiempo para la implementación HashMap, para mostrar todas las cartas. Indique como llegó a ese resultado.

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

Calificación: su programa debe funcionar para ser calificado.

Aspecto	Puntos
Uso del repositorio: existen más de tres versiones guardadas, la última versión es igual a la colocada en Canvas.	10
Patron Factory para seleccionar la implementación a usar en tiempo de corrida	15
Funcionamiento del programa para las operaciones 1, 2, 3 y 5	40
Funcionamiento del programa para las operaciones 4 y 6	15
Análisis de tiempo en ejecución con profiler y la complejidad calculada para HashMap.	20
TOTAL:	100