



Una empresa dedicada al montaje de equipos informáticos dispone de un catálogo de componentes. De cada componente se tiene la siguiente información:

- **Nombre:** nombre del componente, de tipo String, consultable.
- **Tipo:** tipo del componente, del tipo enumerado Tipo, consultable. Hay 10 tipos diferentes: PLACABASE, PROCESADOR, DISCODURO, TARJETAGRAFICA, MEMORIARAM, TARJETASONIDO, VENTILACION, FUENTEALIMENTACION, CAPTURADORAVIDEO y TORRE.
- **Suministrador:** nombre del suministrador, de tipo String, consultable.
- **Demora:** número de días en servir una petición del componente, de tipo Integer, consultable.
- **Precio:** precio en euros, de tipo Double, consultable y modificable.
- **Descuento:** porcentaje de descuento que se aplica al precio por pronto pago, de tipo Double, consultable y modificable.
- **Precio descontado:** precio del componente considerando el descuento, de tipo Double, consultable.
- **Combinadores:** nombres de otros componentes con los que combina frecuentemente en el montaje de equipos informáticos, de tipo Set<String>, consultable.

La información del catálogo se obtiene de un fichero CSV. A continuación, se muestra un ejemplo de línea de dicho fichero:

```
procesador1, PROCESADOR, suministrador1, 2, 233.5, 15.5, [placa1; placa3; memoria2; memoria4]
```

donde procesador1 es un componente de tipo PROCESADOR, suministrado por suministrador1, la demora de una petición de suministro de este componente es de 2 días, su precio en euros es 233.5, el descuento aplicado al componente por pronto pago es 15.5 % y sus combinadores más comunes en el montaje de equipos son [placa1; placa3; memoria2; memoria4], es decir, (procesador1, placa1), (procesador1, placa3), (procesador1, memoria2) y (procesador1, memoria4) son combinaciones que la empresa usa frecuentemente en el montaje de equipos informáticos. Muchas otras combinaciones serían posibles, pero no se registran por no ser usuales.

Ejercicio 1: Tipo base (1 punto)

Implemente el tipo *Componente* según la siguiente descripción:

Propiedades: todas las informaciones relevantes consideradas del componente.

Constructores:

- C1: recibe un parámetro por cada información contenida en la línea del fichero CSV.

Restricciones:

- R1: la demora no debe superar los 45 días.
- R2: el descuento es un porcentaje (comprendido entre 0.0 y 100.0).
- R3: los combinadores no deben incluir al componente en cuestión (es decir, ningún componente se combina consigo mismo).

Criterio de igualdad: dos componentes son iguales si tienen el mismo nombre.

Orden natural: los componentes se ordenan alfabéticamente por su nombre.

Representación como cadena: una cadena que contenga los valores de cada propiedad en el orden en el que se han definido en la descripción del tipo.

Ejercicio 2: Tipo contenedor (0.5 puntos)

Implemente el tipo *Catálogo* de acuerdo con la siguiente descripción:

Propiedades:

- **componentes**: conjunto de componentes del catálogo, de tipo Set<Componente>, consultable.

Constructores:

- C1: constructor sin parámetros. Crea un conjunto de componentes inicialmente vacío.
- C2: recibe una colección de componentes.

Criterio de igualdad: dos catálogos son iguales si lo son sus conjuntos de componentes.

Otras operaciones:

- *void añadirComponente(Componente c)*: añade el componente *c* al conjunto de componentes.
- *void eliminarComponente(String n)*: elimina el componente con nombre *n* del conjunto de componentes. No tiene efecto si este no existe.

Ejercicio 3: Factoría (1.5 puntos)

Implemente el tipo *FactoriaCatalogo* para leer los componentes del fichero CSV y almacenarlos en un catálogo.

La factoría debe contener los siguientes métodos:

- *Catalogo leerCatalogo(String rutaFichero)*: construye un objeto de tipo *Catalogo* con los datos del fichero CSV.
- *Componente parsearComponente(String lineaCSV)*: construye un objeto de tipo *Componente* a partir de una línea del fichero CSV.

Nota: pueden existir componentes que no tengan combinadores asociados (conjunto vacío de combinadores). Por ejemplo: torre1, TORRE, suministrador4, 3, 23.5, 1.5, [] (puede comprobarse en el fichero catalogo.csv suministrado).

Ejercicio 4: Tratamientos secuenciales (6 puntos)

Añada los siguientes tratamientos secuenciales al tipo contenedor. Debe resolver todos los métodos mediante streams, salvo que se le indique expresamente lo contrario. Puede utilizar, cuando lo necesite, las siguientes operaciones auxiliares (no hace falta implementarlas):

```
// Obtiene el objeto componente correspondiente al nombre del componente dado.  
private Componente componenteDesdeNombre(String nombreComponente)
```

```
// Obtiene la colección de objetos componentes correspondientes a los nombres de componentes dados.  
private Collection<Componente> componentesDesdeNombres(Collection<String> nombresComponentes)
```

1. *Map<Tipo, Long> numeroComponentesPorTipoDeSuministrador(String suministrador)*: relaciona cada tipo de componente con el número de componentes de dicho tipo suministrados por *suministrador*. **(1 punto)**
2. *Set<String> componentesConCombinadoresConPrecioMaximo(Double precio)*: obtiene los nombres de los componentes que cumplen la siguiente propiedad: un componente pertenecerá al resultado si y solo si cada uno de sus combinadores no supera el precio dado como parámetro (el precio considerado es sin descuento). **(1.25 puntos)**
3. *Double precioEquipoInformatico(Collection<String> nombresComponentes)*: calcula el precio de un equipo informático construido desde una colección de nombres de componentes dada, suponiendo para estos su precio con descuento. Todo nombre de componente hace referencia a un componente existente en el catálogo. **(1 punto)**

4. *List<String> combinadoresMasFrecuentesOrdenados(Integer n)*: devuelve una lista con los *n* combinadores más frecuentes. La lista estará ordenada de más frecuente a menos frecuente. **(1.5 puntos)**
5. *Set<String> combinacionDeCosteMinimo(Set<Tipo> tipos)*: devuelve un conjunto con los nombres de los componentes con menor precio de los tipos dados como parámetro en el conjunto tipos. El catálogo siempre contiene algún componente de los tipos dados. **Resuelva este método sin utilizar streams. (1.25 puntos)**

Escriba un test para el tipo contenedor. En este test se construirá el catálogo con los datos contenidos en el fichero CSV y se probarán todos los métodos del ejercicio 4. Cada método del ejercicio 4 debe tener su correspondiente método de test (tal como se ha hecho en clase). **(1 punto)**