Examen 2ª Evaluación Java

Profesor: Luis Javier López López

EJERCICIO 1 - (1.5 puntos)

Realiza un método en UtilidadesFeria:

1. public List<Producto> getProductosCoincidente(List<Catering> caterings)
    a. El método devuelve la lista de productos que coinciden entre todos los caterings que se pasan como parámetro.

EJERCICIO 2 - (1 punto)

Realiza un método en UtilidadesFeria:

2. public Map<TipoProducto, List<Producto>> getProductosPorTipo(Caseta caseta)
    a. El método devuelve la lista de productos del catering de la caseta agrupados en forma de mapa tomando como clave el tipo del producto.

EJERCICIO 3 - (2 puntos)

Realiza un método en UtilidadesFeria:

3. public Map<Integer, List<Asistente>> getMayoresDeEdadPorAnyoNacimiento(List<Asistente> asistentes)
    a. El método devuelve en un mapa los asistentes de entre la lista que se pasa, que son mayores de edad (mayores de 18 años), agrupados por su año de nacimiento.

EJERCICIO 4 - (1.5 puntos)

Realiza un método en UtilidadesFeria:

4. public List<Asistente> getPorTipoYSocios(Feria feria, List<TipoAsistente> tipoAsistentes)
    a. Devuelve la lista de asistentes de la feria cuyo tipo coincide con alguno de los tipos pasados y con la condición de que sean socios de más de una caseta.

EJERCICIO 5 - (3 puntos)

Realiza un método en UtilidadesFeria:

5. public ResumenDia obtenerResumenDia(Asistente asistente, Map<Producto, Integer> consumo, Map<Atraccion, Integer> viajesAtraccion)
    a. Devuelve el resumen del día correspondiente al asistente que se pasa como parámetro. Se rellena de la siguiente manera:
        i. **asistente** -> El asistente que se pasa como parámetro.
        ii. **cantidadProductos** -> El mapa de consumo que se pasa como parámetro.
        iii. **privatotalGastadoPorTipoProducto** -> Se calcula a partir del mapa de cantidadProductos, agrupando los productos por tipo y sumando su precio por la cantidad que ha consumido el asistente de cada producto.
        iv. **viajesPorAtraccion** -> el mapa de viajesAtraccion que se pasa.
        v. **totalGastadoPorAtraccion** -> Se calcula a partir del mapa viajesPorAtraccion, se multiplica el precio por viaje de cada atracción por el número de viajes que ha hecho el asistente para dicha atracción.
        vi. **totalGastado**-> la suma total de todo lo gastado en productos y en atracciones. Cabe resaltar que lo gastado por por producto es su precio es el número de veces que lo ha consumido por su precio y lo gastado en las atracciones es el precio por viaje * el número de viajes realizados.

EJERCICIO 6 - (1 punto)

Realiza un método en UtilidadesFeria:

6. public Map<Catering, Double> sumImporteProductosCaterind(List<Catering> caterings)
    a. Devuelve un mapa que tiene como claves los caterings pasados y como valor la suma del importe de sus productos.