



Unidad 5 – Manejo de colecciones

Manejo de listas

Crear la clase Bebida con el atributo “marca”. Sobrecribir los métodos equals() y hashCode() para que se correspondan con este atributo. Crear una lista e iterar sobre los elementos de la misma.

Crear una nueva instancia de Bebida que tenga la misma marca que una del listado y pasarla como parámetro al método remove(Objecto o) de la lista. Volver a recorrer la lista para verificar que ya no esté el objeto.

Manejo de Set

Repetir el ejercicio anterior pero en lugar de utilizar una lista, utilizar un Set con la implementación HashSet.

Luego crear una instancia de Bebida con la misma marca que alguna de las que está dentro del Set y agregarla al listado. Verificar que el Set no haya agregado el elemento.

Manejo de SortedSet

Crear la clase Bebida con el atributo “marca”. Sobrecribir los métodos equals() y hashCode() para que se correspondan con este atributo. Implementar la interfaz Comparable para que las bebidas tengan como orden natural el orden alfabético de la marca.

Crear un SortedSet en base a un TreeSet y agregar 4 bebidas. Mostrar la colección y verificar que se vean en orden alfabético.

Crear otro TreeSet pero esta vez pasar como argumento una instancia de OrdenadorBebidas. Esta clase debe implementar Comparator<Bebida> y ordenar las bebidas por el orden alfabético inverso. Agregar 4 bebidas a este nuevo SortedSet y verificar que se muestren en el orden inverso.

Pizzería

Una pizzería quiere tener un sistema para registrar todos los pedidos a domicilio que realiza.

Los productos que vende son empanadas, pizzas y promociones, que son una suma de los otros productos a un precio total de 25% menos. Por ejemplo una promoción es una Pizza grande de muzarella, 3 empanadas de carne y 3 empanadas de pollo.

Se desea que el sistema, pueda:

- Registrar un pedido (que se pidió y a donde hay que llevarlo), así como que se calcule su precio automáticamente
- Que diga cual es el producto más vendido y su sabor. Tener en cuenta que en una promoción no se cuenta, sino que se toma en cuenta las pizzas y empanadas que efectivamente se venden. Por ejemplo: empanada de carne.
- Que calcule el total vendido.



Ordenamiento de listados

Crear la clase Persona con los atributos "Nombre", "Apellido", "Edad" y "DNI". Agregar 5 elementos a un List. Reordenar el List a través de `java.util.Collections.sort(List l, Comparator c)` según los siguientes criterios:

- Por apellido y nombre en orden alfabético
- Por nombre y apellido en orden alfabético inverso
- Por edad de menor a mayor
- Por DNI de mayor a menor

En todos los casos, luego de ordenar el listado, imprimirlo en pantalla para verificar el orden correcto.

Depurar colecciones

Crear un método que dado una colección del objeto Persona, elimine de dicha colección todas las personas menores de 21 años de edad.

Estadísticas de alumnos

A final de año, en un sistema de un colegio secundario están registrados los alumnos de cada curso, las fechas de las distintas ausencias y las notas de cada materia en cada uno de los 3 trimestres de este año. En cada falta se registra si está justificada o no.

Se desea crear un programa que muestre genere un Map de los cursos y la cantidad de alumnos que no van a repetir de año.

Si un alumno se queda libre, repite de grado. Esto se da cuando:

- Tiene más de 15 faltas no justificadas.
- Tiene más de 25 faltas totales.

Si un alumno desaprueba más de 2 materias, puede repetir (ya que le pueden quedar más de 2 materias libres). Un alumno no aprueba la materia cuando:

- Tiene menos de 6 en el 3er trimestre.
- Tiene un promedio menor a 6 entre los 3 trimestres