Este código simula la gestión de vehículos de distintas marcas, permitiendo agregar vehículos a una lista y realizar varias operaciones sobre ellos, como eliminar duplicados y agruparlos por marca. A continuación te explico cada sección del código de manera ordenada.

**Importación de Librerías**

import .util.\*;

import modelo.Marca;

import modelo.Vehiculo;

Se importan las librerías necesarias:

* .util.\*: Contiene clases útiles como ArrayList, HashSet, HashMap, LinkedList, Random, Scanner, etc.
* modelo.Marca y modelo.Vehiculo: Se importan las clases Marca y Vehiculo que se usan para crear objetos con los datos de las marcas de los vehículos y los vehículos mismos.

**Clase Principal y Método main**

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Aquí comienza la clase Main y el método main que ejecutará el programa. Se crea un objeto Scanner para permitir la entrada de datos por parte del usuario.

**Paso 1: Array de nombres de marcas**

String[] nombresMarcas = {"Toyota", "Ford", "Chevrolet", "Honda", "Volkswagen"};

Se define un array nombresMarcas con algunos nombres de marcas de vehículos.

**Paso 2: Crear ArrayList de marcas**

ArrayList<Marca> listaMarcas = new ArrayList<>();

Random random = new Random();

Se crea una lista (ArrayList) llamada listaMarcas para almacenar objetos de la clase Marca. También se instancia un objeto Random para generar códigos aleatorios para cada marca.

for (String nombre : nombresMarcas) {

int codigo = 100 + random.nextInt(900);

listaMarcas.add(new Marca(nombre, codigo));

}

Se recorre el array nombresMarcas, y por cada nombre de marca, se genera un código aleatorio entre 100 y 999, y se crea un objeto Marca que se agrega a listaMarcas.

System.out.println("Marcas disponibles:");

for (Marca marca : listaMarcas) {

System.out.println(" - " + marca);

}

Aquí se imprime cada marca disponible utilizando el método toString de la clase Marca. Si no se hubiera modificado toString(), se mostraría el hashcode de los objetos.

**Paso 3: Crear LinkedList para vehículos**

LinkedList<Vehiculo> listaVehiculos = new LinkedList<>();

Se crea una lista doblemente enlazada (LinkedList) llamada listaVehiculos para almacenar objetos de la clase Vehiculo.

String opcion;

do {

System.out.print("\nIngrese patente del vehículo: ");

String patente = scanner.nextLine();

System.out.print("Ingrese nombre de la marca: ");

String nombreMarca = scanner.nextLine();

Se inicia un ciclo do-while que permite ingresar los datos de los vehículos (patente y marca) de manera repetida. Se pide al usuario que ingrese la patente y el nombre de la marca.

Marca marcaSeleccionada = null;

for (Marca m : listaMarcas) {

if (m.getNombre().equalsIgnoreCase(nombreMarca)) {

marcaSeleccionada = m;

break;

}

}

Se busca la marca ingresada en la lista listaMarcas. Si se encuentra, se asigna a marcaSeleccionada.

if (marcaSeleccionada != null) {

listaVehiculos.add(new Vehiculo(patente, marcaSeleccionada));

System.out.println("Vehículo agregado.");

} else {

System.out.println("Marca no encontrada.");

}

Si la marca es válida (es decir, se encuentra en la lista de marcas), se crea un nuevo objeto Vehiculo con la patente y la marca seleccionada, y se agrega a la lista listaVehiculos.

System.out.print("¿Desea agregar otro vehículo? (s/n): ");

opcion = scanner.nextLine();

} while (opcion.equalsIgnoreCase("s"));

El ciclo continúa pidiendo más vehículos si el usuario responde "s" a la pregunta de si desea agregar otro vehículo.

**Paso 4: Mostrar patentes sin duplicados (Usando Set)**

Set<String> patentesUnicas = new HashSet<>();

for (Vehiculo v : listaVehiculos) {

patentesUnicas.add(v.getPatente());

}

Se crea un HashSet llamado patentesUnicas para almacenar las patentes de los vehículos, asegurando que no haya duplicados. Las Set no permiten elementos repetidos.

System.out.println("\nPatentes únicas:");

for (String patente : patentesUnicas) {

System.out.println("- " + patente);

}

Se imprime la lista de patentes únicas (sin duplicados) en consola.

**Paso 5: Agrupar vehículos por marca (Usando Map)**

Map<String, ArrayList<Vehiculo>> mapaVehiculosPorMarca = new HashMap<>();

Se crea un HashMap para agrupar los vehículos por marca. La clave es el nombre de la marca (String) y el valor es una lista de vehículos (ArrayList<Vehiculo>).

for (Vehiculo v : listaVehiculos) {

String nombreMarca = v.getMarca().getNombre();

if (!mapaVehiculosPorMarca.containsKey(nombreMarca)) {

mapaVehiculosPorMarca.put(nombreMarca, new ArrayList<>());

}

mapaVehiculosPorMarca.get(nombreMarca).add(v);

}

Se recorre la lista de vehículos y se agrupan en el mapa según la marca. Si la marca no existe en el mapa, se crea una nueva entrada.

System.out.println("\nVehículos agrupados por marca:");

for (Map.Entry<String, ArrayList<Vehiculo>> entrada : mapaVehiculosPorMarca.entrySet()) {

System.out.println("Marca: " + entrada.getKey());

for (Vehiculo v : entrada.getValue()) {

System.out.println(" - " + v.getPatente());

}

}

Se imprime la lista de vehículos agrupados por marca. Se muestran las marcas y sus vehículos asociados.

**Opcional: Mostrar cantidad de vehículos por marca**

System.out.println("\nCantidad de vehículos por marca:");

for (Map.Entry<String, ArrayList<Vehiculo>> entrada : mapaVehiculosPorMarca.entrySet()) {

System.out.println(entrada.getKey() + ": " + entrada.getValue().size());

}

Esta sección opcional muestra la cantidad de vehículos registrados para cada marca.

**Cierre del Scanner**

scanner.close();

Se cierra el objeto Scanner al final del programa para liberar recursos.