**Ejercicios sesiones 22 23 y 24**

**Durante estos ejercicios se va a trabajar sobre las letras S, O y L de los principios SOLID, deberás:**

1. **Crea una clase coche aplicándole el principio de responsabilidad única (S).**

**Clase Coche**

package Ejercicio23;  
  
public class Coche {  
 //atributos  
 Integer ID;  
 String marca;  
 String modelo;  
 Integer year;  
  
 //constructores  
 public Coche(){}  
  
 public Coche(Integer ID, String marca, String modelo, Integer year) {  
 this.ID = ID;  
 this.marca = marca;  
 this.modelo = modelo;  
 this.year = year;  
 }  
 @Override  
 public String toString() {  
 return "Coche{" +  
 "ID=" + ID +  
 ", marca='" + marca + '\'' +  
 ", modelo='" + modelo + '\'' +  
 ", year=" + year +  
 '}';  
 }  
}

**Clase Coches:** Solo gestiona coches

package Ejercicio23;  
  
//SOLO gestiona solo coches  
  
import java.util.ArrayList;  
  
public class Coches {  
  
 private CochesDB cochesDB = new CochesDB();  
  
 public void create(Coche coche){  
 if (findOne(coche.ID) != null){  
 return;  
 }  
 cochesDB.insertar(coche);  
 }  
  
 public ArrayList<Coche> findAll(){  
 return cochesDB.convertirCocheenFicheroArrayList();  
 }  
  
 public Coche findOne(Integer ID) {  
 ArrayList<Coche> coches = cochesDB.convertirCocheenFicheroArrayList();  
  
 for (Coche cocheActual : coches) {  
 if (cocheActual.ID.equals(ID)) {  
 return cocheActual;  
 }  
 }  
 return null;  
 }  
  
 public void delete(Integer ID){  
 Coche coche = new Coche();  
 coche.ID = ID;  
 cochesDB.borrar(coche);  
  
 }  
}

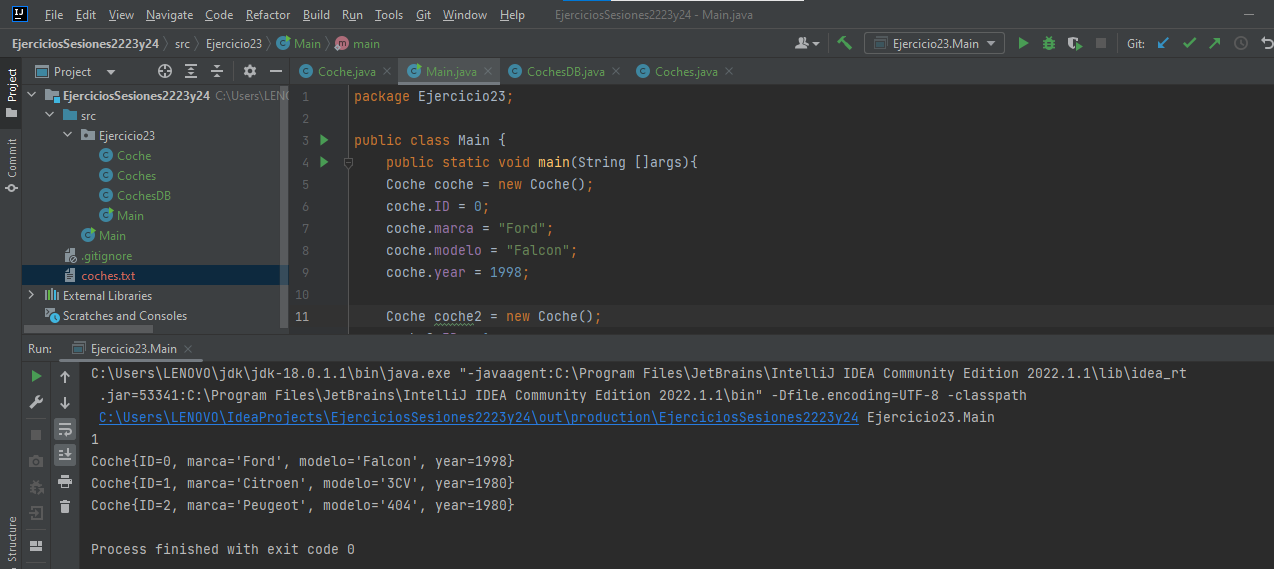
**Clase CochesDB:** Solo manipula los datos de la BD (convertir clase coche en fichero(ArrayList), insertar, separar con comas y borrar )

package Ejercicio23;  
  
//Manipula el almacen de datos  
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.PrintStream;  
import java.util.ArrayList;  
import java.util.Scanner;  
  
public class CochesDB {  
 public String ficheroCoche = "coches.txt";  
 public ArrayList<Coche> convertirCocheenFicheroArrayList() {  
 ArrayList<Coche> coches = new ArrayList();  
  
 try {  
 Scanner scanner = new Scanner(new File(ficheroCoche));  
  
 while (scanner.hasNext()) {  
 String cocheActual = scanner.next();  
 String[] partes = cocheActual.split(",");  
  
 Coche coche = new Coche();  
 coche.ID = Integer.*parseInt*(partes[0]);  
 coche.marca = partes[1];  
 coche.modelo = partes[2];  
 coche.year = Integer.*parseInt*(partes[3]);  
  
  
 coches.add(coche);  
 }  
 scanner.close();  
  
 } catch (Exception e) {  
 }  
 return coches;  
 }  
  
 public void insertar(Coche coche) {  
 try {  
 FileOutputStream fileOutputStream = new FileOutputStream(ficheroCoche, true);  
 PrintStream printStream = new PrintStream(fileOutputStream);  
 printStream.println(separarCochePorComas(coche));  
 printStream.flush();  
 printStream.close();  
 }  
 catch (Exception e) {  
 }  
 }  
  
 public String separarCochePorComas(Coche coche){  
 return coche.ID + ","  
 +coche.marca + ","  
 +coche.modelo + ","  
 +coche.year;  
 }  
  
 public void borrar(Coche coche){  
 ArrayList<Coche> coches = convertirCocheenFicheroArrayList();  
  
 for (int i = 0; i < coches.size(); i++){  
 if (coches.get(i).ID.equals(coche.ID)){  
 coches.remove(i);  
 }  
 }  
 try {  
 PrintStream printStream = new PrintStream(ficheroCoche);  
 for (Coche cocheActual : coches) {  
 String cocheComas = separarCochePorComas(cocheActual);  
 printStream.println(cocheComas);  
 }  
 printStream.flush();  
 printStream.close();  
 }  
 catch (Exception e) {  
 }  
 }  
 }

**Clase Main**

package Ejercicio23;  
  
public class Main {  
 public static void main(String []args){  
 Coche coche = new Coche();  
 coche.ID = 0;  
 coche.marca = "Ford";  
 coche.modelo = "Falcon";  
 coche.year = 1998;  
  
 Coche coche2 = new Coche();  
 coche2.ID = 1;  
 coche2.marca = "Citroen";  
 coche2.modelo = "3CV";  
 coche2.year = 1980;  
  
 Coche coche3 = new Coche();  
 coche3.ID = 2;  
 coche3.marca = "Peugeot";  
 coche3.modelo = "404";  
 coche3.year = 1980;  
  
 Coches coches = new Coches();  
 coches.create(coche);  
 coches.create(coche2);  
 coches.create(coche3);  
  
 Coche ID = coches.findOne(1);  
 System.*out*.println(ID.ID);  
  
 //coches.delete(1);  
 for(Coche a : coches.findAll()){  
 System.*out*.println(a);  
 }  
 }  
}

**Run**

****

1. **Continúa con el código extendiéndolo para aumentar su funcionalidad pero sin modificarlo aplicando el principio abierto/cerrado(O).**

Se agrega una clase de Estadísticas de bajas y altas extendiendo la clase de CochesDB. Tomamos las funcionalidades insertar y borrar de esta clase y se hacen las agregaciones (totalAltas, totalBajas) y se generan los getters.

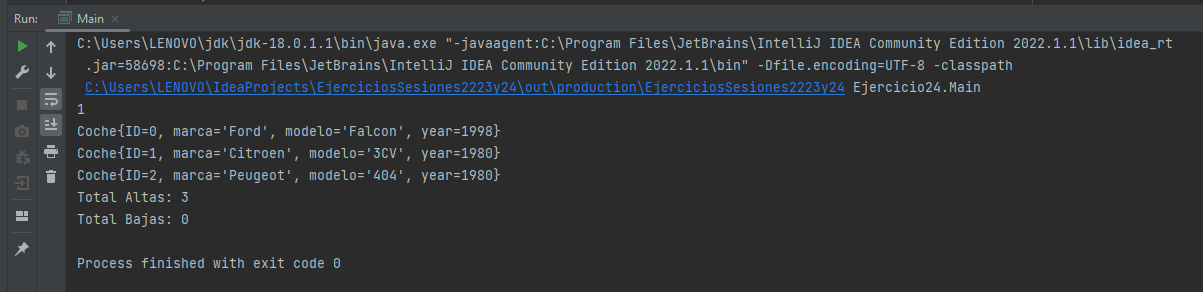
Clase CochesDBEstadisticas

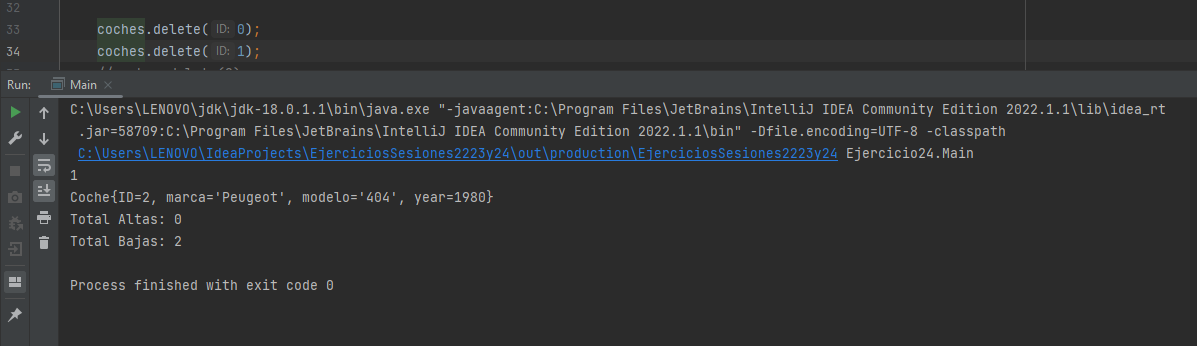
package Ejercicio24;  
  
public class CochesDBEstadisticas extends CochesDB{  
 private int totalAltas = 0;  
 private int totalBajas = 0;  
  
 @Override  
 public void insertar(Coche coche) {  
 super.insertar(coche);  
 totalAltas++;  
 }  
 @Override  
 public void borrar(Coche coche) {  
 super.borrar(coche);  
 totalBajas++;  
 }  
 public int getTotalAltas() {  
 return totalAltas;  
 }  
 public int getTotalBajas() {  
 return totalBajas;  
 }  
}

Clase Main

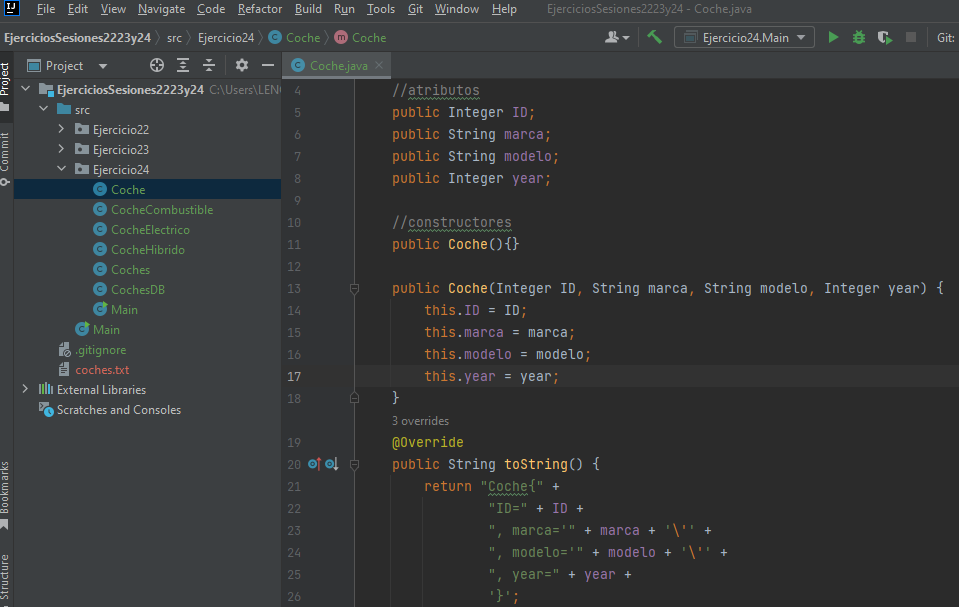
package Ejercicio24;  
  
public class Main {  
 public static void main(String []args){  
 Coche coche = new Coche();  
 coche.ID = 0;  
 coche.marca = "Ford";  
 coche.modelo = "Falcon";  
 coche.year = 1998;  
  
 Coche coche2 = new Coche();  
 coche2.ID = 1;  
 coche2.marca = "Citroen";  
 coche2.modelo = "3CV";  
 coche2.year = 1980;  
  
 Coche coche3 = new Coche();  
 coche3.ID = 2;  
 coche3.marca = "Peugeot";  
 coche3.modelo = "404";  
 coche3.year = 1980;  
  
 CochesDBEstadisticas cochesDB = new CochesDBEstadisticas();  
 Coches coches = new Coches();  
  
 coches.create(coche);  
 coches.create(coche2);  
 coches.create(coche3);  
  
 Coche ID = coches.findOne(1);  
 System.*out*.println(ID.ID);  
  
 //coches.delete(0);  
 //coches.delete(1);  
 //coches.delete(2);  
 for(Coche a : coches.findAll()){  
 System.*out*.println(a);  
 }  
 System.*out*.println("Total Altas: " + coches.cochesDB.getTotalAltas());  
 System.*out*.println("Total Bajas: " + coches.cochesDB.getTotalBajas());  
 }  
}

**Run**

****

****

1. **Crea subclases de la clase coche, por ejemplo un coche eléctrico, de combustible y uno híbrido y aplica el principio de substitución de Liskov (L), las subclases deben ser sustituibles por la superclase coche.**

****

**Clase Coche**

package Ejercicio24;  
  
public class Coche {  
 //atributos  
 public Integer ID;  
 public String marca;  
 public String modelo;  
 public Integer year;  
  
 //constructores  
 public Coche(){}  
  
 public Coche(Integer ID, String marca, String modelo, Integer year) {  
 this.ID = ID;  
 this.marca = marca;  
 this.modelo = modelo;  
 this.year = year;  
 }  
 @Override  
 public String toString() {  
 return "Coche{" +  
 "ID=" + ID +  
 ", marca='" + marca + '\'' +  
 ", modelo='" + modelo + '\'' +  
 ", year=" + year +  
 '}';  
 }  
}

**Clase CocheCombustible**

package Ejercicio24;  
  
public class CocheCombustible extends Coche {  
 public String motor;  
  
 public CocheCombustible() {  
 super();  
 }  
  
 public CocheCombustible(Integer ID, String marca, String modelo, Integer year, String motor) {  
 super(ID, marca, modelo, year);  
 this.motor = motor;  
 }  
  
 @Override  
 public String toString() {  
 return "CocheCombustible{" +  
 "motor='" + motor + '\'' +  
 ", ID=" + ID +  
 ", marca='" + marca + '\'' +  
 ", modelo='" + modelo + '\'' +  
 ", year=" + year +  
 '}';  
 }  
}

**Clase CocheElectrico**

package Ejercicio24;  
  
public class CocheElectrico extends Coche {  
 public String motor;  
  
 public CocheElectrico() {  
 super();  
 }  
  
 public CocheElectrico(Integer ID, String marca, String modelo, Integer year, String motor) {  
 super(ID, marca, modelo, year);  
 this.motor = motor;  
 }  
  
 @Override  
 public String toString() {  
 return "CocheElectrico{" +  
 "motor='" + motor + '\'' +  
 ", ID=" + ID +  
 ", marca='" + marca + '\'' +  
 ", modelo='" + modelo + '\'' +  
 ", year=" + year +  
 '}';  
 }  
}

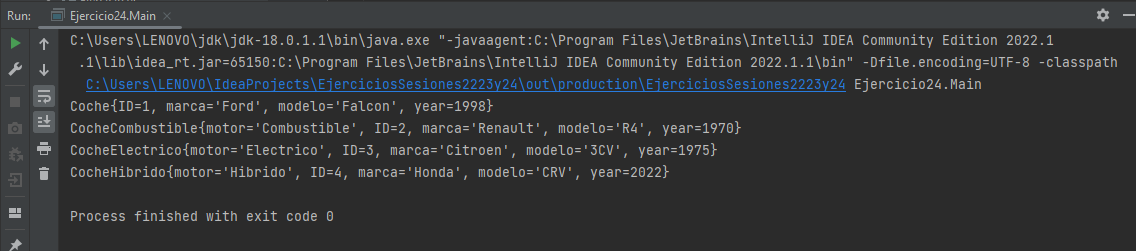
**Clase CocheHibrido**

package Ejercicio24;  
  
public class CocheHibrido extends Coche {  
 public String motor;  
  
 public CocheHibrido() {  
 super();  
 }  
  
 public CocheHibrido(Integer ID, String marca, String modelo, Integer year, String motor) {  
 super(ID, marca, modelo, year);  
 this.motor = motor;  
 }  
 @Override  
 public String toString() {  
 return "CocheHibrido{" +  
 "motor='" + motor + '\'' +  
 ", ID=" + ID +  
 ", marca='" + marca + '\'' +  
 ", modelo='" + modelo + '\'' +  
 ", year=" + year +  
 '}';  
 }  
}

**Clase Main**

package Ejercicio24;  
  
public class Main {  
 public static void main(String []args){  
  
 Coche coche1 = new Coche();  
 coche1.ID = 1;  
 coche1.marca = "Ford";  
 coche1.modelo = "Falcon";  
 coche1.year = 1998;  
  
 CocheCombustible coche2 = new CocheCombustible();  
 coche2.ID = 2;  
 coche2.marca = "Renault";  
 coche2.modelo = "R4";  
 coche2.year = 1970;  
 coche2.motor = "Combustible";  
  
 CocheElectrico coche3 = new CocheElectrico();  
 coche3.ID = 3;  
 coche3.marca = "Citroen";  
 coche3.modelo = "3CV";  
 coche3.year = 1975;  
 coche3.motor = "Electrico";  
  
 CocheHibrido coche4 = new CocheHibrido();  
 coche4.ID = 4;  
 coche4.marca = "Honda";  
 coche4.modelo = "CRV";  
 coche4.year = 2022;  
 coche4.motor = "Hibrido";  
  
 Coches coches = new Coches();  
  
 coches.create(coche1);  
 coches.create(coche2);  
 coches.create(coche3);  
 coches.create(coche4);  
  
 System.*out*.println(coche1);  
 System.*out*.println(coche2);  
 System.*out*.println(coche3);  
 System.*out*.println(coche4);  
  
 }  
}

**RUN**

****