**Exercici 2**

Exercici 2 bis

Fes un programa que agafant com a model Comarca, EscriureFitxerObject i LlegirFitxerObject:

1. Es creen objectes tipus Cotxes: marca, model, any i matricula.
2. Pugui guardar i recuperar objectes tipus Cotxe com un bloc.
3. Pugui guardar objectes tipus Cotxe d'un en un.
4. Pugui recuperar objectes tipus Cotxe segons un camp.

S'ha d'entregar PDF amb el següent contingut:

* Enllaç al repositori al núvol.
* Els fonts remarcant-ne el que s'ha modificat respecte dels codis d'exemple subministrats.
* Bolcats de pantalla amb DIFERENTS execucions (no només una) del programa.

**Equip:**

Raul Santos

Tiffany Fernández

**Enllaç:**

<https://github.com/TiffanyFA/M6-UF1>

**Codi:**

Hi ha un error amb el punter, hem seguit les teves indicacions (no passar el fitxer per paràmetre en els mètodes) però el problema persisteix. També hem provat de no treballar de forma modular i posar-ho tot al main i el que fa, és que afegeix bé el cotxe però no llista bé, només mostra el cotxe afegit, així que hem descartat aquesta opció i entreguem la primera opció on llista de forma correcte i afegeix el cotxe però després d’afegir-lo salta error amb el punter.

Aquest projecte consta de:

* + Classe Cotxe amb els seus atributs, constructor, setters i getters.
  + Classe Tester, el programa:

import java.io.\*;

import java.util.Scanner;

public class Tester {

// LLEGIR FITXER

public static void **llegirFitxerComplet**() throws IOException, ClassNotFoundException {

// Camp variable tipus Cotxe

**Cotxe cotxe;**

// Declaració del fitxer

File fitxer = new File("**cotxes.txt**");

// Crea el flux d'entrada

FileInputStream filein = new FileInputStream(fitxer);

// Connectar el flux de bytes al flux de dades

ObjectInputStream dataInCotxe = new ObjectInputStream(filein);

try {

while (true) {// Llegeix el fitxer

// Llegeix el Cotxe

**cotxe** = (**Cotxe**) **dataInCotxe**.readObject();

System.out.println**("Marca: " + cotxe.getMarca() + " Model: " + cotxe.getModel() + " Matricula: "**

**+ cotxe.getMatricula() + " Any: " + cotxe.getAny()**);

}

} catch (EOFException eo) {

}

**dataInCotxe.close()**;// Tanca el stream d'entrada

}

// LLEGIR FITXER FILTRAT

public static void **llegirFitxerConcret()** throws IOException, ClassNotFoundException {

Scanner teclat = new Scanner(System.in);

**String resposta**;

// Camp variable tipus Cotxe

**Cotxe cotxe**;

// Declaració del fitxer

File fitxer = new File("**cotxes.txt**");

// Crea el flux d'entrada

FileInputStream filein = new FileInputStream(fitxer);

// Connectar el flux de bytes al flux de dades

ObjectInputStream dataInCotxe = new ObjectInputStream(filein);

// demanar dades

System.out.println("**Indica la marca, el model o any dels cotxes a filtrar**:");

**resposta = teclat.next();**

try {

while (true) {// Llegeix el fitxer

// Llegeix el Cotxe

**cotxe = (Cotxe) dataInCotxe.readObject();**

**if (resposta.equalsIgnoreCase(cotxe.getMarca()) || resposta.equalsIgnoreCase(cotxe.getModel())**

**|| resposta.equalsIgnoreCase(cotxe.getAny())) {**

**System.out.println("Marca: " + cotxe.getMarca() + " Model: " + cotxe.getModel() + " Matricula: "**

**+ cotxe.getMatricula() + " Any: " + cotxe.getAny())**;

}

}

} catch (EOFException eo) {

}

filein.reset();

**dataInCotxe.close()**;// Tanca el stream d'entrada

}

// ESCRIURE FITXER

public static void **escriureFitxer()** throws IOException {

**Scanner teclat = new Scanner(System.in);**

**String marca;**

**String model;**

**String matricula;**

**String any;**

// Camp variable tipus Cotxe

**Cotxe cotxe**;

// Declaració del fitxer

File fitxer = new File("**cotxes.txt**");

// Crea el flux de sortida

FileOutputStream fileout = new FileOutputStream(fitxer, **true**);

// Connectar el flux de bytes al flux de dades

ObjectOutputStream dataOuCotxe = new ObjectOutputStream(fileout);

// Les dades per generar els objectes Cotxe

System.out.println("**Introdueix marca, model, matricula i any**:");

**marca = teclat.nextLine();**

**model = teclat.nextLine();**

**matricula = teclat.nextLine();**

**any = teclat.nextLine();**

// crear cotxe

**cotxe = new Cotxe(marca, model, matricula, any);**

// escriu a fitxer

dataOuCotxe.writeObject(cotxe);

dataOuCotxe.reset();

dataOuCotxe.close();// Tanca el stream de sortida

}

// CREAR FITXER

public static void **creaFitxer()** throws IOException {

// Camp variable tipus Cotxe

**Cotxe cotxe;**

// Declaració del fitxer

File fitxer = new File("**cotxes.txt**");

// Crea el flux de sortida

FileOutputStream fileout = new FileOutputStream(fitxer, **true**);

// Connectar el flux de bytes al flux de dades

ObjectOutputStream dataOuCotxe = new ObjectOutputStream(fileout);

// Les dades per generar els objectes Cotxe

**String marca[] = { "Opel", "Citroen", "Peugeot" };**

**String model[] = { "Corsa", "C3", "Ion" };**

**String matricula[] = { "1258GTP", "9820NFS", "3548SE" };**

**String any[] = { "2006", "2012", "2018" };**

// Recorre els arrays

for (int i = 0; i < marca.length; i++) {// Crea la Cotxe

**cotxe = new Cotxe(marca[i], model[i], matricula[i], any[i]);**

dataOuCotxe.writeObject(**cotxe**);// L'escriu al fixer

}

dataOuCotxe.close();// Tanca el stream de sortida

}

public static void main(String[] args) throws IOException, ClassNotFoundException {

// variables

**Scanner teclat = new Scanner(System.in);**

**int resposta;**

**boolean sortir = false;**

**Cotxe cotxe;**

// Declaració del fitxer

**File fitxer = new File("cotxes.txt");**

// creem fitxer per defecte

**creaFitxer();**

while (!sortir) {

// menu

**System.out.println("Tria una opció");**

**System.out.println("1 per mostrar els cotxes");**

**System.out.println("2 per guardar un nou cotxe");**

**System.out.println("3 per mostrar un tipus de cotxe");**

**System.out.println("4 per sortir");**

**resposta = teclat.nextInt();**

// crida metodes

**if (resposta == 1) {**

**llegirFitxerComplet();**

**} else if (resposta == 2) {**

**escriureFitxer();**

**} else if (resposta == 3) {**

**llegirFitxerConcret();**

**} else {**

**sortir = true;**

**}**

}

// tancar recursos

**teclat.close();**

}

}

**Bocats:**







