|  |
| --- |
| ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  ΑΚΑΔΗΜΑΙΚΟ ΕΤΟΣ 2012 – 2013 Εαρινό εξαμηνο |
| Διαδίκτυο και Εφαρμογές |
|  |
| Άσκηση 3η : SOAP Web Service |
|  |
|  |

8Ο ΕΞΑΜΗΝΟ

**ΦΟΙΤΗΤΗΣ:**

ΚΟΓΙΑΣ ΜΑΡΙΟΣ-ΕΥΑΓΓΕΛΟΣ

**Σκοπός**

Σκοπός της συγκεκριμένης άσκησης είναι η δημιουργία μιας Web Service βασισμένης στο πρωτόκολλο SOAP. Το θέμα της εφαρμογής σχετίζεται με αυτοκίνητα. Δίνεται λοιπόν η δυνατότητα στο χρήστη να δει τη λίστα των διαθέσιμων αυτοκινήτων και τα χαρακτηριστικά τους, να προσθέσει και να διαγράψει από τη λίστα αυτή καθώς και να επιλέξει ένα συγκεκριμένο αυτοκίνητο.

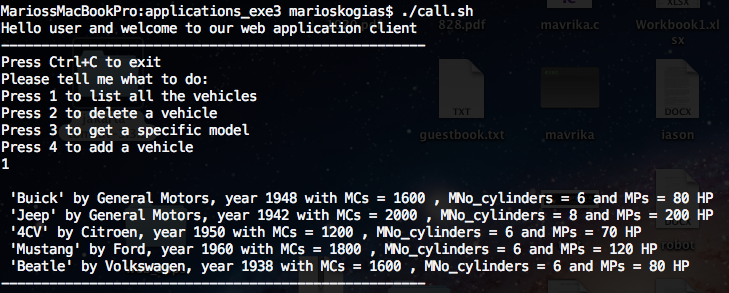
**Υλοποίηση**

Προφανώς, η υλοποίησή μας μπορεί να χωριστεί σε 2 μέρη, αυτό του server και αυτό του client. Το κομμάτι του server είναι αυτό που διαθέτει τις πληροφορίες μετά από αιτήματα που στέλνει ο client. Το πλέον σημαντικό πο αφορά τα web services είναι ότι και οποιοσδήποτε άλλος client πιυ ήξερε το όνομα και τον τρόπο επικοινωνίας με την εφαρμογή θα μπορούσε να έχει δοσοληψίες με αυτή, ανεξαρτήτως γλώσσας υλοποίησης. Η συγκεκριμένη υλοποίηση είναι σε java τόσο για το serverside κομμάτι όσο και για τον client.

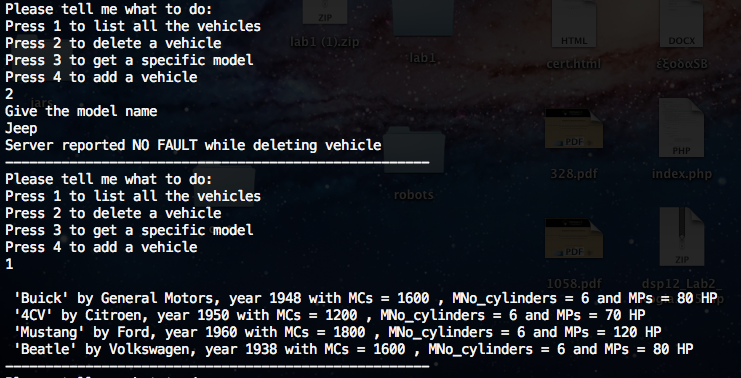
Σχετικά με την πληροφορία που αποθηκεύεται χρησιμοποιούνται 2 Java Beans ένα για τα οχήματα και ένα για τους κινητήρες τους και αυτά τοποθετούνται σε ένα HashTable ανάλογα με το όνομα του μοντέλου. JavaBeans ωστόσο είναι και αυτά που μεταφέρονται μέσω του SOAP. Επομένως, χρειάστηκε να ορίσουμε serializer για τη μεταφορά.

**Εκτέλεση**

Παρακάτω παρουσιάζονται screenshots από την εκτέλεση της εφαρμογής.



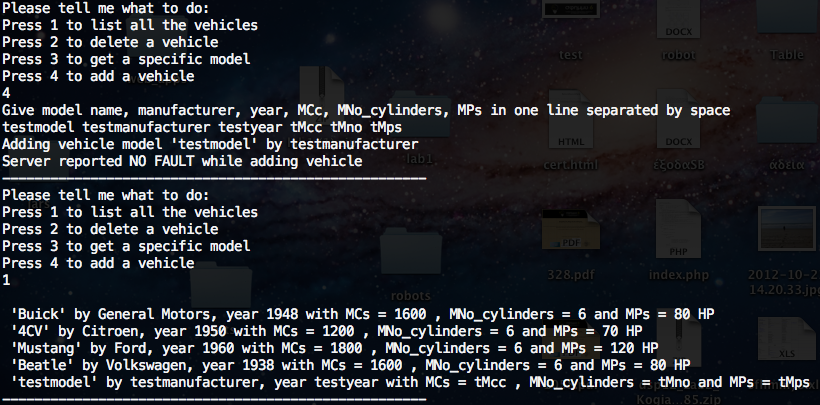
*Απαρίθμηση όλων των οχημάτων*

**

*Διαγραφή οχήματος*

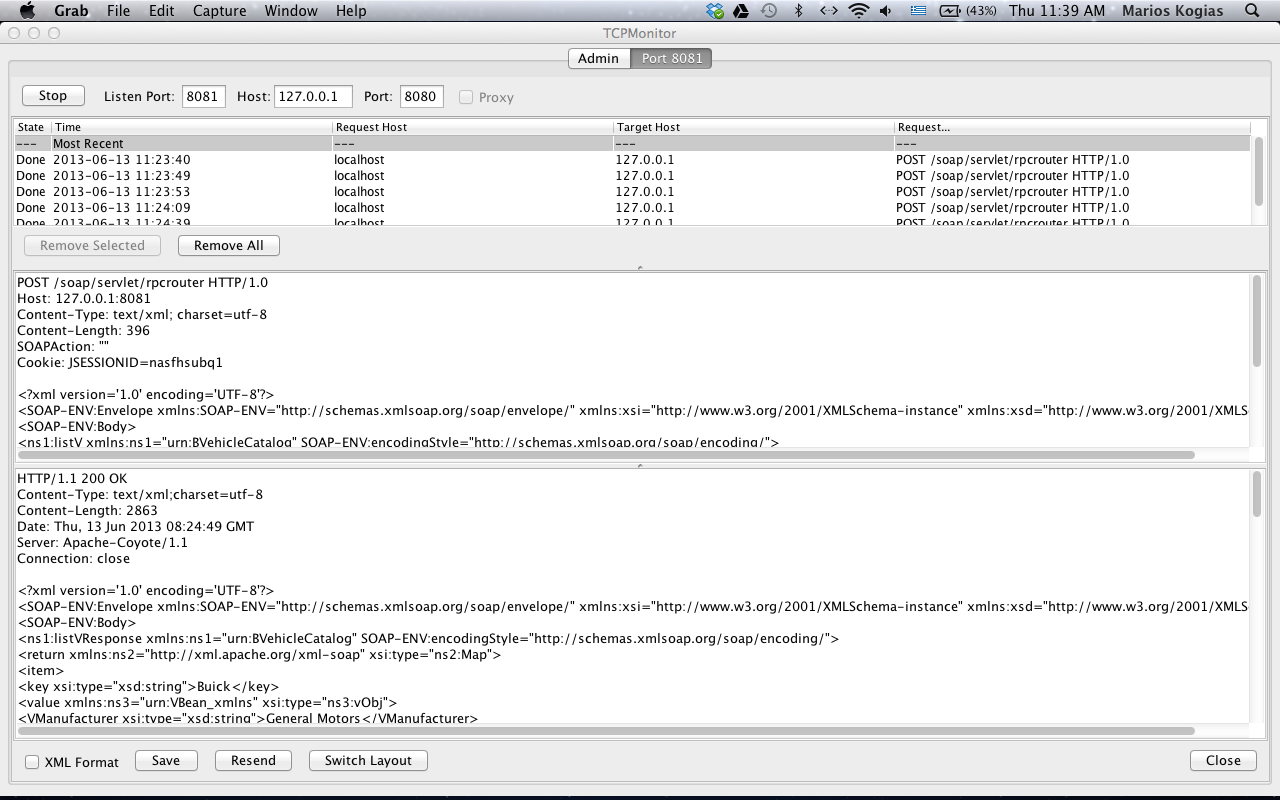
**

*Επιλογή οχήματος*

**

*Προσθήκη οχήματος*

Επιπλέον χρησιμοποιήσαμε το tcp monitor για να παρατηρήσουμε την ανταλλαγή πληροφορίας μέσω του XML.



*Ένα screenshot του tcp monitor.*

Παρακάτω θα δείξουμε ένα παράδειγμα ανταλλαγής πληροφορίας στο οποίο ο client ζητά την απαρρίθμηση όλων των οχημάτων και παίρνει πίσω την απάντηση.

POST /soap/servlet/rpcrouter HTTP/1.0

Host: 127.0.0.1:8081

Content-Type: text/xml; charset=utf-8

Content-Length: 396

SOAPAction: ""

Cookie: JSESSIONID=nasfhsubq1

<?xml version='1.0' encoding='UTF-8'?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<SOAP-ENV:Body>

<ns1:listV xmlns:ns1="urn:BVehicleCatalog" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

</ns1:listV>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

Το μήνυμα που έφτασε στον server από τον client

HTTP/1.1 200 OK

Content-Type: text/xml;charset=utf-8

Content-Length: 2863

Date: Thu, 13 Jun 2013 08:24:49 GMT

Server: Apache-Coyote/1.1

Connection: close

<?xml version='1.0' encoding='UTF-8'?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<SOAP-ENV:Body>

<ns1:listVResponse xmlns:ns1="urn:BVehicleCatalog" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

<return xmlns:ns2="http://xml.apache.org/xml-soap" xsi:type="ns2:Map">

<item>

<key xsi:type="xsd:string">Buick</key>

<value xmlns:ns3="urn:VBean\_xmlns" xsi:type="ns3:vObj">

<VManufacturer xsi:type="xsd:string">General Motors</VManufacturer>

<VModel xsi:type="xsd:string">Buick</VModel>

<VMotor xmlns:ns4="urn:MBean\_xmlns" xsi:type="ns4:mObj">

<MCc xsi:type="xsd:string">1600</MCc>

<MNo\_cylinders xsi:type="xsd:string">6</MNo\_cylinders>

<MPs xsi:type="xsd:string">80 HP</MPs>

</VMotor>

<VYear xsi:type="xsd:string">1948</VYear>

</value>

</item>

<item>

<key xsi:type="xsd:string">4CV</key>

<value xmlns:ns5="urn:VBean\_xmlns" xsi:type="ns5:vObj">

<VManufacturer xsi:type="xsd:string">Citroen</VManufacturer>

<VModel xsi:type="xsd:string">4CV</VModel>

<VMotor xmlns:ns6="urn:MBean\_xmlns" xsi:type="ns6:mObj">

<MCc xsi:type="xsd:string">1200</MCc>

<MNo\_cylinders xsi:type="xsd:string">6</MNo\_cylinders>

<MPs xsi:type="xsd:string">70 HP</MPs>

</VMotor>

<VYear xsi:type="xsd:string">1950</VYear>

</value>

</item>

<item>

<key xsi:type="xsd:string">Mustang</key>

<value xmlns:ns7="urn:VBean\_xmlns" xsi:type="ns7:vObj">

<VManufacturer xsi:type="xsd:string">Ford</VManufacturer>

<VModel xsi:type="xsd:string">Mustang</VModel>

<VMotor xmlns:ns8="urn:MBean\_xmlns" xsi:type="ns8:mObj">

<MCc xsi:type="xsd:string">1800</MCc>

<MNo\_cylinders xsi:type="xsd:string">6</MNo\_cylinders>

<MPs xsi:type="xsd:string">120 HP</MPs>

</VMotor>

<VYear xsi:type="xsd:string">1960</VYear>

</value>

</item>

<item>

<key xsi:type="xsd:string">testmodel</key>

<value xmlns:ns9="urn:VBean\_xmlns" xsi:type="ns9:vObj">

<VManufacturer xsi:type="xsd:string">testmanufacturer</VManufacturer>

<VModel xsi:type="xsd:string">testmodel</VModel>

<VMotor xmlns:ns10="urn:MBean\_xmlns" xsi:type="ns10:mObj">

<MCc xsi:type="xsd:string">tMcc</MCc>

<MNo\_cylinders xsi:type="xsd:string">tMno</MNo\_cylinders>

<MPs xsi:type="xsd:string">tMps</MPs>

</VMotor>

<VYear xsi:type="xsd:string">testyear</VYear>

</value>

</item>

<item>

<key xsi:type="xsd:string">Beatle</key>

<value xmlns:ns11="urn:VBean\_xmlns" xsi:type="ns11:vObj">

<VManufacturer xsi:type="xsd:string">Volkswagen</VManufacturer>

<VModel xsi:type="xsd:string">Beatle</VModel>

<VMotor xmlns:ns12="urn:MBean\_xmlns" xsi:type="ns12:mObj">

<MCc xsi:type="xsd:string">1600</MCc>

<MNo\_cylinders xsi:type="xsd:string">6</MNo\_cylinders>

<MPs xsi:type="xsd:string">80 HP</MPs>

</VMotor>

<VYear xsi:type="xsd:string">1938</VYear>

</value>

</item>

</return>

</ns1:listVResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

*Το μήνυμα απάντησης του server*

**Κώδικας Υλοποίησης**

*Server*

/\*

\*

\* BVCatalog.java

\*

\*/

package bvShop;

import java.util.Hashtable;

public class BVCatalog {

/\*\* The vehicles as Hashtable \*/

/\*\* the car Model is the key for finding entries \*/

private Hashtable<String,VehicleBean> catalog;

public BVCatalog() {

System.out.println("called the constructor");

/\*\* the old catalog remains \*/

/\*\* but now the catalog (Hashtable) will contain also beans !!! \*/

catalog = new Hashtable<String,VehicleBean>();

/\*\* Some content - we NOW polulate with some objects!! \*/

addV(new VehicleBean("Buick", "General Motors", "1948",new MotorBean("1600","6","80 HP")));

addV(new VehicleBean("Mustang", "Ford", "1960",new MotorBean("1800","6","120 HP")));

addV(new VehicleBean("4CV", "Citroen", "1950",new MotorBean("1200","6","70 HP")));

addV(new VehicleBean("Jeep", "General Motors", "1942",new MotorBean("2000","8","200 HP")));

addV(new VehicleBean("Beatle", "Volkswagen", "1938",new MotorBean("1600","6","80 HP")));

}

/\*\* FIRST METHOD TO BE EXPOSED - addV \*/

/\*\* input argument is now a single ('complex') object !! \*/

/\*\* nothing is returned \*/

public void addV(VehicleBean vObj) {

if (vObj == null) {

throw new IllegalArgumentException(

"The object provided cannot be null.");

}

/\*\* entry format for the Hashtable: key,data \*/

/\*\* vObj.getVModel gets the 'key' out of the bean object \*/

catalog.put(vObj.getVModel(), vObj);

System.out.println("Addition at server side: " + vObj.getVModel());

}

/\*\* SECOND METHOD TO BE EXPOSED - getVehicleBean \*/

/\*\* input argument is a String \*/

/\*\* a ('complex') object is returned \*/

public VehicleBean getVehicleBean(String model) {

if (model == null) {

throw new IllegalArgumentException("carModel cannot be null.");

}

// Return the requested vehicle - NOW AS AN OBJECT !!!!

return (VehicleBean) catalog.get(model);

}

/\*\* THIRD METHOD TO BE EXPOSED - listV \*/

/\*\* NO input argument - a whole table is returned \*/

/\*\* ... , and that with ('complex') objects as entries !!!! \*/

public Hashtable<String,VehicleBean> listV() {

System.out.println("in listing...");

System.out.println(catalog);

return catalog;

}

/\*\* FOURTH METHOD TO BE EXPOSED - delVehicleBean \*/

/\*\* input argument is a string \*/

/\*\* nothing is returned \*/

public void delVehicleBean(String model) {

System.out.println("deleting...");

catalog.remove(model);

}

}

/\*

\*

\* MotorBean.java

\*

\*/

package bvShop;

public class MotorBean {

String MCc;

String MNo\_cylinders;

String MPs;

public MotorBean() {

}

public MotorBean(String mcs,String mno\_cylinders, String mps) {

MCc = mcs;

MNo\_cylinders = mno\_cylinders;

MPs = mps;

}

public String getMCc() {

return MCc;

}

public void setMCc(String mCc) {

MCc = mCc;

}

public String getMNo\_cylinders() {

return MNo\_cylinders;

}

public void setMNo\_cylinders(String mNo\_cylinders) {

MNo\_cylinders = mNo\_cylinders;

}

public String getMPs() {

return MPs;

}

public void setMPs(String mPs) {

MPs = mPs;

}

public String toString() {

return MCc + " " + MNo\_cylinders + " " + MPs;

}

}

/\*

\*

\* VehicleBean.java

\*

\*/

package bvShop;

public class VehicleBean {

String VModel;

String VManufacturer;

String VYear;

public MotorBean VMotor;

/\*\* The following non-argument constructor MUST be always present \*/

/\*\* for this class to be a Java Bean \*/

public VehicleBean() {

}

/\*\* Another constructor CAN also be always present \*/

/\*\*

\* the following initializes all properties whenever a new object is

\* instantiated (with 'new') -

\*/

public VehicleBean(String model, String manu, String year,MotorBean motor) {

this.VModel = model;

this.VManufacturer = manu;

this.VYear = year;

this.VMotor = motor;

}

/\*\* get & set methods for all properties MUST be present \*/

public String getVModel() {

return VModel;

}

public void setVModel(String model) {

this.VModel = model;

}

public String getVManufacturer() {

return VManufacturer;

}

public void setVManufacturer(String manu) {

this.VManufacturer = manu;

}

public String getVYear() {

return VYear;

}

public void setVYear(String year) {

this.VYear = year;

}

public MotorBean getVMotor() {

return VMotor;

}

public void setVMotor(MotorBean motor) {

this.VMotor = motor;

}

// toString is an in-built method for every Java object. Outputs an

// informative string

public String toString() {

return "'" + VModel + "' by " + VManufacturer + " (" + VYear + ") and motor " + VMotor;

}

}

*Client*

import java.net.URL;

import java.util.Enumeration;

import java.util.Hashtable;

import java.util.Vector;

import org.apache.soap.Constants;

import org.apache.soap.Fault;

import org.apache.soap.SOAPException;

import org.apache.soap.encoding.SOAPMappingRegistry;

import org.apache.soap.encoding.soapenc.BeanSerializer;

import org.apache.soap.rpc.Call;

import org.apache.soap.rpc.Parameter;

import org.apache.soap.rpc.Response;

import org.apache.soap.util.xml.QName;

import java.util.Scanner;

import bvShop.VehicleBean;

import bvShop.MotorBean;

public class BVAdderLister {

public Call call;

public URL url;

public BVAdderLister(String address) {

/\*add the target url\*/

try {

url = new URL(address);

}

catch (Exception e) {

e.printStackTrace();

}

SOAPMappingRegistry reg = new SOAPMappingRegistry();

BeanSerializer serializer = new BeanSerializer();

reg.mapTypes(Constants.NS\_URI\_SOAP\_ENC,

new QName("urn:VBean\_xmlns","vObj"),

VehicleBean.class, serializer, serializer);

reg.mapTypes(Constants.NS\_URI\_SOAP\_ENC,

new QName("urn:MBean\_xmlns","mObj"),

MotorBean.class, serializer, serializer);

//Build the Call object

call = new Call();

//How to map, where to send, method to call, encoding "style"

call.setSOAPMappingRegistry(reg);

call.setTargetObjectURI("urn:BVehicleCatalog");

call.setEncodingStyleURI(Constants.NS\_URI\_SOAP\_ENC);

}

public void add(String model,String manufacturer,String year, String MCs,String MNo\_cylinders, String MPs) {

MotorBean motor = new MotorBean(MCs,MNo\_cylinders,MPs);

VehicleBean vObj = new VehicleBean(model, manufacturer, year,motor);

this.call.setMethodName("addV");

System.out.println("Adding vehicle model '" + model + "' by " + manufacturer);

// Set up the parameters of the call

Vector params = new Vector();

//in the instructions given to the 'Serializer' - see Depl. Descr.

params.addElement(new Parameter("vObj", VehicleBean.class, vObj, null));

this.call.setParams(params);

// Invoke the call

try {

Response response;

response = call.invoke(this.url, "");

//We do not expect something back, unless there is a fault!!

if (!response.generatedFault())

System.out.println("Server reported NO FAULT while adding vehicle");

else {

Fault fault = response.getFault();

System.out.println("Server reported FAULT while adding:");

System.out.println(fault.getFaultString());

}

} catch (Exception e) {

e.printStackTrace();

}

}

public void list() {

/\* Another method is now called\*/

this.call.setMethodName("listV");

/\* NO parameters here !!\*/

/\*(we cannot have a call with arguments as before)\*/

this.call.setParams(null);

// Invoke the call; here we expect something back !!

try {

Response response;

response = this.call.invoke(this.url, "");

/\*Extract the value returned in the form of a 'Parameter' Object\*/

Parameter returnValue = response.getReturnValue();

/\*Cast the 'Parameter' Object onto a Hashtabel Object\*/

Hashtable<String,VehicleBean> catalog = (Hashtable<String,VehicleBean>)returnValue.getValue();

Enumeration e = catalog.keys();

while (e.hasMoreElements()) {

String VModel = (String)e.nextElement();

VehicleBean vo = (VehicleBean)catalog.get(VModel);

System.out.println(" '" + vo.getVModel() + "' by " + vo.getVManufacturer() +

", year " + vo.getVYear() + " with MCs = " + vo.VMotor.getMCc() + " , MNo\_cylinders = "

+ vo.VMotor.getMNo\_cylinders() + " and MPs = "+ vo.VMotor.getMPs());

}

}

catch (Exception e) {

e.printStackTrace();

}

}

public void deleteCar(String model) {

this.call.setMethodName("delVehicleBean");

Vector params = new Vector();

params.addElement(new Parameter("model", String.class, model, null));

call.setParams(params);

try {

Response response;

response = this.call.invoke(this.url, "");

//We do not expect something back, unless there is a fault!!

if (!response.generatedFault())

System.out.println("Server reported NO FAULT while deleting vehicle");

else {

Fault fault = response.getFault();

System.out.println("Server reported FAULT while deleting:");

System.out.println(fault.getFaultString());

}

} catch (Exception e) {

e.printStackTrace();

}

}

public void getModel(String model) {

this.call.setMethodName("getVehicleBean");

Vector params = new Vector();

params.addElement(new Parameter("model", String.class, model, null));

this.call.setParams(params);

// Invoke the call; here we expect something back !!

try {

Response response;

response = this.call.invoke(this.url, "");

/\*Extract the value returned in the form of a 'Parameter' Object\*/

Parameter returnValue = response.getReturnValue();

/\*Cast the 'Parameter' Object onto a Hashtabel Object\*/

VehicleBean vo = (VehicleBean)returnValue.getValue();

System.out.println(" '" + vo.getVModel() + "' by " + vo.getVManufacturer() +

", year " + vo.getVYear() + " with MCs = " + vo.VMotor.getMCc() + " , MNo\_cylinders = " +

vo.VMotor.getMNo\_cylinders() + " and MPs = "+ vo.VMotor.getMPs());

} catch (Exception e) {

e.printStackTrace();

}

}

public static void main(String[] args) {

/\*create the object - get the link from the terminal\*/

BVAdderLister adderlister = new BVAdderLister(args[0]);

System.out.println("Hello user and welcome to our web application client");

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

System.out.println("Press Ctrl+C to exit");

while(true) {

System.out.println("Please tell me what to do:");

System.out.println("Press 1 to list all the vehicles");

System.out.println("Press 2 to delete a vehicle");

System.out.println("Press 3 to get a specific model");

System.out.println("Press 4 to add a vehicle");

try {

Scanner in = new Scanner(System.in);

int val = in.nextInt();

String line;

switch (val) {

case 1:

System.out.println();

adderlister.list();

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

break;

case 2:

System.out.println("Give the model name");

line = in.nextLine();

line = in.nextLine();

adderlister.deleteCar(line);

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

break;

case 3:

System.out.println("Give the model name");

line = in.nextLine();

line = in.nextLine();

adderlister.getModel(line);

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

break;

case 4:

System.out.println("Give model name, manufacturer, year, MCc, MNo\_cylinders, MPs in one line separated by space");

line = in.nextLine();

line = in.nextLine();

String splits[] = line.split(" ");

adderlister.add(splits[0],splits[1],splits[2],splits[3],splits[4],splits[5]);

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

break;

}

} catch (Exception e) {

System.out.println("An error occured");

System.exit(1);

}

}

}

}

Συνολικά τον κώδικα της εφαρμογής μπορείτε να βρείτε στο https://github.com/marioskogias/webApplications/tree/master/applications\_exe3