Міністерство освіти і науки України

Національний технічний університет

«Харківський політехнічний інститут»

Кафедра «Обчислювальна техніка та програмування»

ЗВІТ

Про виконання лабораторної роботи №8

“Основи введення/виведення Java SE”

Виконавець:

студент гр. КІТ-120В

Олексієнко Микита

Харків 2021

Лабораторна робота №8.

**Основи введення/виведення Java SE**

**Мета**

- Оволодіння навичками управління введенням/виведенням даних з використанням класів платформи Java SE.

**1.Вимоги**

**1.1 Розробник:**

Олексієнко Микита Віталійович

студент групи КІТ-120В

15 варіант

**1.2 Загальне завдання**

Реалізувати програму відповідно до індивідуального завдання.

1. Забезпечити можливість збереження і відновлення масива об'єктів рішення завдання лабораторної роботи №7.
2. Забороняється використання стандартного протокола серіалізації.
3. Продемонструвати використання моделі Long Term Persistence.
4. Забезпечити діалог з користувачем у вигляді простого текстового меню.
5. При збереженні та відновленні даних забезпечити діалоговий режим вибору директорії з відображенням вмісту і можливістю переміщення по підкаталогах.

**1.3 Прикладні задачі**

Автомобіль: марка, рік випуску, технічні характеристики у вигляді "параметр - значення" (серед яких витрата палива в міському і заміському циклі), технічний стан, ціна.

**2. Опис програми**

**2.1 Засоби ООП:**

У лабороторного роботі 8 , я використовую клас Program, який э головним класом та в якому безпосередньо оголошую приватні поля і публічні статичні методи. Також я оголосив клас Auto, який має всі перераховані вище поля та методи обробки полей.

**2.2 Ієрархія та структура класів:**

* Program.java
* Auto.java
* Garage.java
* Menu.java

**2.3 Важливі фрагменти програми:**

package ua.khpi.oop.alekseenko08;  
  
import java.beans.XMLEncoder;  
import java.beans.XMLDecoder;  
import java.io.IOException;  
import java.io.FileOutputStream;  
import java.io.FileInputStream;  
  
public class Garage {  
 int carsCount = 0;  
 private Auto[] cars;  
  
 public void addCar ( Auto car ) {  
 if (carsCount == 0) {  
 carsCount++;  
 cars = new Auto[carsCount];  
 cars[0] = car;  
 } else {  
 Auto[] temp = cars;  
 carsCount++;  
 cars = new Auto[carsCount];  
  
 for ( int i = 0; i < carsCount - 1; i++ ) {  
 cars[i] = temp[i];  
 }  
  
 cars[carsCount-1] = car;  
 }  
 }  
 public boolean removeCar ( int number ) {  
 for ( int i = 0; i < carsCount; i++ ) {  
 if ( i == number ) {  
 // удаление со сдвигом+  
  
 Auto[] temp = new Auto[carsCount-1];  
  
 //arraycopy(1 массив, исходная позиция в 1 массиве, 2 массив, исходная позиция во 2 массиве, длина)  
  
 System.arraycopy(cars, 0, temp, 0, i);  
 System.arraycopy(cars, i + 1, temp, i, carsCount - i - 1);  
  
 cars = temp;  
 carsCount--;  
 return true;  
 }  
 }  
 return false;  
 }  
 public void carList () {  
 if ( cars == null ) {  
 System.err.println("No cars in your garage!\n");  
 return;  
 }  
  
 for ( Auto car : cars ) {  
 System.out.println(car.toString());  
 }  
  
 System.out.print("\n");  
 }  
  
 public void save ( String path ) {  
 System.out.println("SAVING...\n");  
 try (XMLEncoder xmlEncoder = new XMLEncoder(new FileOutputStream(path))) {  
 xmlEncoder.writeObject(cars);  
 xmlEncoder.flush();  
 }  
 catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void load ( String path ) {  
 System.out.println("LOADING...\n");  
 try (XMLDecoder xmlDecoder = new XMLDecoder(new FileInputStream(path))) {  
 cars = (Auto[]) xmlDecoder.readObject();  
 }  
 catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

package ua.khpi.oop.alekseenko08;  
  
import java.io.File;  
import java.util.Scanner;  
  
import java.util.regex.\*;  
  
public class Menu {  
  
 private static final String ANSI\_RESET = "\u001B[0m";  
 private static final String ANSI\_YELLOW = "\u001B[33m";  
 private static final String ANSI\_GREEN = "\u001B[32m";  
 private static final String ANSI\_BLUE = "\u001B[34m";  
  
 private static Garage garage;  
 public static void create () {  
 garage = new Garage();  
  
 while (true) {  
 System.out.println(ANSI\_GREEN + "CHOOSE ANY TASK:" + ANSI\_RESET);  
 System.out.println(ANSI\_YELLOW + "[1] Add Car");  
 System.out.println("[2] Remove Car");  
 System.out.println("[3] Car List");  
 System.out.println("[4] Save");  
 System.out.println("[5] Load");  
 System.out.println("[6] Close" + ANSI\_RESET + "\n");  
  
 Scanner scan = new Scanner(System.in);  
 int chooseNumber = scan.nextInt();  
  
 Auto car = new Auto();  
  
 switch (chooseNumber) {  
 case 1:  
 if (carAdding()) {  
 System.out.println(ANSI\_GREEN + "ADDED!\n" + ANSI\_RESET );  
 } else {  
 System.err.println("Please try add correctly");  
 }  
 break;  
 case 2: garage.removeCar( inputCarNumber() ); break;  
 case 3: {  
 garage.carList();  
 break;  
 }  
 case 4: SAVING(); break;  
 case 5: LOADING(); break;  
 case 6:  
 return;  
 default: {  
 System.err.println("ERROR");  
 System.out.println("- Please select the correct command!");  
 }  
 }  
 }  
 }  
  
 private static boolean carAdding () {  
 Auto ADDED\_CAR = new Auto();  
  
 Scanner scan = new Scanner(System.in);  
  
 System.out.println("Please type your car model:");  
 String model = scan.nextLine();  
 if ( model.equals("") ) {  
 ERROR\_MESS();  
 return false;  
 } else ADDED\_CAR.setModel( model );  
  
 System.out.println("Please type your car release car:");  
 String year = scan.nextLine();  
 ADDED\_CAR.setReleaseYear( year.equals("") ? 0 : Integer.parseInt(year) );  
  
 System.out.println("Please type your car urban fuel:");  
 String urbanFuel = scan.nextLine();  
 ADDED\_CAR.setUrbanFuel( urbanFuel.equals("") ? 0 : Integer.parseInt(urbanFuel) );  
  
 System.out.println("Please type your car suburban fuel:");  
 String suburbanFuel = scan.nextLine();  
 ADDED\_CAR.setSubUrbanFuel( suburbanFuel.equals("") ? 0 : Integer.parseInt(suburbanFuel) );  
  
 System.out.println("Please choose your car state:");  
 System.out.println("[1] Good");  
 System.out.println("[2] Bad");  
 String technicalCondition = scan.nextLine();  
 ADDED\_CAR.setTechnicalCondition(Integer.parseInt(technicalCondition) == 1);  
  
 System.out.println("Please type your car price:");  
 String price = scan.nextLine();  
 ADDED\_CAR.setPrice( price.equals("") ? 0 : Integer.parseInt(price) );  
  
 garage.addCar(ADDED\_CAR);  
 return true;  
 }  
  
 private static int inputCarNumber () {  
 Scanner scan = new Scanner(System.in);  
 String number = scan.nextLine();  
  
 if ( number.equals("") ) {  
 return 0;  
 } else return Integer.parseInt(number);  
 }  
  
 private static void ERROR\_MESS () {  
 System.err.println("Error");  
 }  
  
 private static void SAVING() {  
 System.out.println("Enter the path to your file:");  
 String PATH = "C:/Users/FORCH/Desktop";  
  
 while (true) {  
 System.out.print(PATH + "/");  
 Scanner scan = new Scanner(System.in);  
 String command = scan.nextLine();  
  
 if ( command.equals("exit") ) {  
 System.out.println("");  
 return;  
 }  
  
  
 // REGEX -----------------------------------  
  
 Pattern CD\_PATTERN = Pattern.compile("cd\\s.+");  
 Matcher CD\_MATCHER = CD\_PATTERN.matcher(command);  
  
 Pattern TOUCH\_XML\_PATTERN = Pattern.compile("touch\\s.+\\.xml");  
 Matcher TOUCH\_XML\_MATCHER = TOUCH\_XML\_PATTERN.matcher(command);  
  
 // -----------------------------------------  
 if ( command.equals("ls") ) {  
 File dir = new File(PATH);  
 for(File item : dir.listFiles()){  
 if(item.isDirectory()){  
 System.out.println( ANSI\_GREEN + "- " + item.getName() + ANSI\_RESET);  
 } else {  
 Pattern LS\_XML\_PATTERN = Pattern.compile(".+\\.xml");  
 Matcher LS\_XML\_MATCHER = LS\_XML\_PATTERN.matcher(item.getName());  
  
 if ( LS\_XML\_MATCHER.find() ) {  
 System.out.println(ANSI\_BLUE + "- " + item.getName() + ANSI\_RESET);  
 }  
 }  
 }  
 }  
 else if (CD\_MATCHER.find()) {  
 String cd\_string = command.substring(CD\_MATCHER.start() + 3, CD\_MATCHER.end());  
 PATH = PATH + "/" + cd\_string;  
 } else if (TOUCH\_XML\_MATCHER.find()) {  
 String file\_name = command.substring(TOUCH\_XML\_MATCHER.start() + 6, TOUCH\_XML\_MATCHER.end());  
  
 File file\_path = new File(PATH + "/" + file\_name);  
  
 if ( file\_path.exists() ) {  
 System.out.println(ANSI\_YELLOW + "WARNING" + ANSI\_RESET);  
 System.out.println("The file already exists at the given path. Recreate?");  
 System.out.println("[1] - yes");  
 System.out.println("[2] - no");  
  
 Scanner recreate = new Scanner(System.in);  
 int returned = recreate.nextInt();  
  
 if ( returned == 1 ) {  
 garage.save(PATH + "/" + file\_name);  
 }  
 } else garage.save(PATH + "/" + file\_name);  
  
 }  
  
 // -----------------------------------------  
 }  
 }  
 private static void LOADING() {  
 System.out.println("Enter the path to your file:");  
 String PATH = "C:/Users/FORCH/Desktop";  
  
 while (true) {  
 System.out.print(PATH + "/");  
 Scanner scan = new Scanner(System.in);  
 String command = scan.nextLine();  
  
 if ( command.equals("exit") ) {  
 System.out.println("");  
 return;  
 }  
  
 // REGEX -----------------------------------  
  
 Pattern CD\_PATTERN = Pattern.compile("cd\\s.+");  
 Matcher CD\_MATCHER = CD\_PATTERN.matcher(command);  
  
 Pattern CAT\_XML\_PATTERN = Pattern.compile("cat\\s.+\\.xml");  
 Matcher CAT\_XML\_MATCHER = CAT\_XML\_PATTERN.matcher(command);  
  
 // -----------------------------------------  
 if ( command.equals("ls") ) {  
 File dir = new File(PATH);  
 for(File item : dir.listFiles()){  
 if(item.isDirectory()){  
 System.out.println( ANSI\_GREEN + "- " + item.getName() + ANSI\_RESET);  
 } else {  
 Pattern LS\_XML\_PATTERN = Pattern.compile(".+\\.xml");  
 Matcher LS\_XML\_MATCHER = LS\_XML\_PATTERN.matcher(item.getName());  
  
 if ( LS\_XML\_MATCHER.find() ) {  
 System.out.println(ANSI\_BLUE + "- " + item.getName() + ANSI\_RESET);  
 }  
 }  
 }  
 }  
 else if (CD\_MATCHER.find()) {  
 String cd\_string = command.substring(CD\_MATCHER.start() + 3, CD\_MATCHER.end());  
 PATH = PATH + "/" + cd\_string;  
 } else if (CAT\_XML\_MATCHER.find()) {  
 String file\_name = command.substring(CAT\_XML\_MATCHER.start() + 4, CAT\_XML\_MATCHER.end());  
 garage.load(PATH + "/" + file\_name);  
 }  
  
 // -----------------------------------------  
 }  
 }  
}

**3. Висновок:**

* Оволодів навичками управління введенням/виведенням даних з використанням класів платформи Java SE.
* Розробив прикладну галузь.