**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 1**

по дисциплине:

«Визуальные средства разработки программных приложений»

на тему:

«Создание собственных классов и интерфейсов в *Java*»

Выполнил: студент гр. ИТП-31

Расшивалов Н.И.

Принял: ассистент

Михалевич В.Г.

Гомель 2021

**Цель работы:** изучить способы создания собственных классов и интерфейсов в *Java.*

**Задание:** Вариант 8. Условия задания на рисунках 1 – 2.

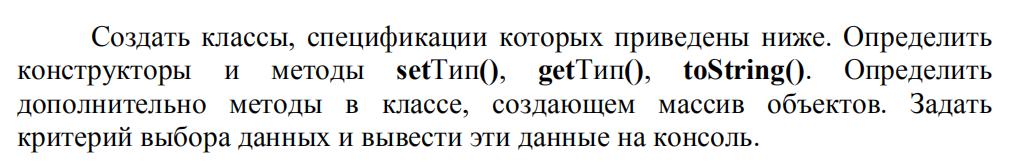


Рисунок 1 – Условие задания

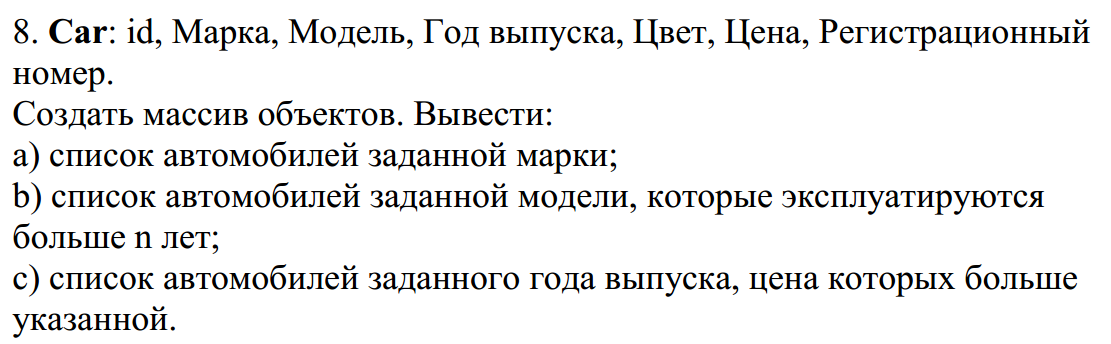


Рисунок 2 – Вариант задания

В классе Main было написано заполнение базы тестовыми значениями, далее были написаны методы, решающие данные задания.

Результат выполнения задания представлен на рисунке 3.

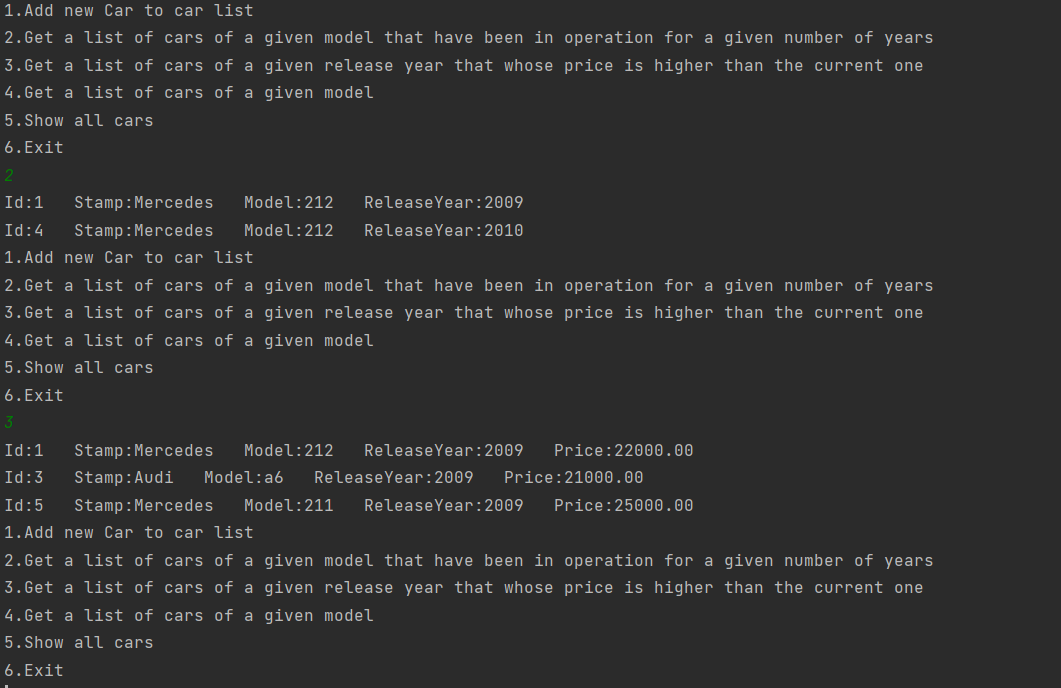


Рисунок 3 – Результат выполнения задания

**Вывод:** Были написаны аксессоры, классы и их реализации соответственно поставленному заданию.

**ПРИЛОЖЕНИЕ А**

Листинг созданных классов

package com.Lab\_1;  
  
import java.util.ArrayList;  
import java.util.List;  
  
**public class Car** {  
  
 int CarId ;  
 String Stamp;  
 String Model;  
 int ReleaseYear;  
 String Color;  
 double Price;  
 String Number;  
  
 public Car(int carId, String stamp, String model, int releaseYear, String color, double price, String number){  
 setCarId(carId);  
 setStamp(stamp);  
 setModel(model);  
 setReleaseYear(releaseYear);  
 setColor(color);  
 setPrice(price);  
 setNumber(number);  
 }  
  
 public int getCarId() {  
 return CarId;  
 }  
  
 public void setCarId(int carId) {  
 CarId = carId;  
 }  
  
 public String getStamp() {  
 return Stamp;  
 }  
  
 public void setStamp(String stamp) {  
 Stamp = stamp;  
 }  
  
 public String getModel() {  
 return Model;  
 }  
  
 public void setModel(String model) {  
 Model = model;  
 }  
  
 public int getReleaseYear() {  
 return ReleaseYear;  
 }  
  
 public void setReleaseYear(int releaseYear) {  
 ReleaseYear = releaseYear;  
 }  
  
 public String getColor() {  
 return Color;  
 }  
  
 public void setColor(String color) {  
 Color = color;  
 }  
  
 public double getPrice() {  
 return Price;  
 }  
  
 public void setPrice(double price) {  
 Price = price;  
 }  
  
 public String getNumber() {  
 return Number;  
 }  
  
 public void setNumber(String number) {  
 Number = number;  
 }  
  
}

package com.Lab\_1;  
  
import java.time.Year;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Scanner;  
  
**public class Main** {  
  
 public static void main(String[] args) {  
 List<Car> Cars = new ArrayList<>();  
 Initialize(Cars);  
 int choice = 0;  
 while (choice != 6){  
 choice = ShowMenu();  
 if (choice > 0 & choice <= 6){  
 System.out.flush();  
 Menu(choice,Cars);  
 }else {  
 System.out.println("Enter normal number man!");  
 }  
 }  
 }  
  
 public static List<Car> GetGivenCars(String stamp, List<Car> Cars) {  
 List<Car> GivenCars = new ArrayList<>();  
 for (var car:Cars){  
 if ( car.Stamp.equals(stamp)){  
 GivenCars.add(car);  
 }  
 }  
 return GivenCars;  
 }  
  
 public static List<Car> GetGivenModelCars(String model, int moreYear,List<Car> Cars){  
 List<Car> GivenModelCars = new ArrayList<>();  
 int year = Year.now().getValue();  
 for (var car:Cars){  
 if ( car.Model.equals(model) & year - car.ReleaseYear >= moreYear){  
 GivenModelCars.add(car);  
 }  
 }  
 return GivenModelCars;  
 }  
  
 public static List<Car> GetGivenPriceCars(Double price, int releaseYear,List<Car> Cars){  
 List<Car> GivenPriceCars = new ArrayList<>();  
 for (var car:Cars){  
 if ( car.ReleaseYear == releaseYear & car.Price > price){  
 GivenPriceCars.add(car);  
 }  
 }  
 return GivenPriceCars;  
 }  
  
 public static Car SetCar(){  
 Scanner in = new Scanner(System.in);  
 System.out.println("Enter carId");  
 int carId = in.nextInt();  
 System.out.println("Enter stamp");  
 String stamp = in.next();  
 System.out.println("Enter model");  
 String model = in.next();  
 System.out.println("Enter color");  
 String color = in.next();  
 System.out.println("Enter price");  
 double price = in.nextDouble();  
 System.out.println("Enter releaseYear");  
 int releaseYear = in.nextInt();  
 System.out.println("Enter number");  
 String number = in.next();  
 Car car = new Car(carId, stamp, model, releaseYear, color, price, number);  
 return car;  
 }  
  
 public static int ShowMenu(){  
 Scanner in = new Scanner(System.in);  
 System.out.println("1.Add new Car to car list");  
 System.out.println("2.Get a list of cars of a given model that " +  
 "have been in operation for a given number of years");  
 System.out.println("3.Get a list of cars of a given release year that " +  
 "whose price is higher than the current one");  
 System.out.println("4.Get a list of cars of a given model");  
 System.out.println("5.Show all cars");  
 System.out.println("6.Exit");  
 int choice = in.nextInt();  
 return choice;  
 }  
  
 public static void Menu(int choice, List<Car> Cars){  
 switch (choice) {  
 case (1):  
 Cars.add(SetCar());  
 break;  
 case (2):  
 List<Car> GivenModelCars = GetGivenModelCars("212",10,Cars);  
 for (var car:GivenModelCars) {  
 System.out.printf("Id:%d Stamp:%s Model:%s ReleaseYear:%d%n",  
 car.CarId,car.Stamp,car.Model, car.ReleaseYear);  
 }  
 break;  
 case (3):  
 List<Car> GivenPriceCars = GetGivenPriceCars(20000.0,2009,Cars);  
 for (var car:GivenPriceCars) {  
 System.out.printf("Id:%d Stamp:%s Model:%s ReleaseYear:%d Price:%.2f%n",  
 car.CarId,car.Stamp,car.Model, car.ReleaseYear,car.Price);  
 }  
 break;  
 case (4):  
 List<Car> GivenCars = GetGivenCars("Mercedes",Cars);  
 for (var car:GivenCars) {  
 System.out.printf("Id:%d Stamp:%s Model:%s%n",car.CarId,car.Stamp,car.Model);  
 }  
 break;  
 case (5):  
 ShowAllCars(Cars);  
 break;  
 default:  
 System.exit(0);  
 break;  
 }  
  
 }  
  
 public static void ShowAllCars(List<Car> Cars){  
 for (var car:Cars) {  
 System.out.printf("Id:%d Stamp:%s Model:%s ReleaseYear:%d Price:%.1f Color:%s Number:%s%n",  
 car.CarId,car.Stamp,car.Model, car.ReleaseYear,car.Price,car.Color,car.Number);  
 }  
 }  
  
 public static void Initialize(List<Car> Cars){  
 Car carOne = new Car(1,"Mercedes","212",2009,"black",22000,"7666");  
 Cars.add(carOne);  
 Car carTwo = new Car(2,"Bmw","m5",2010,"Silver",23000,"3333");  
 Cars.add(carTwo);  
 Car carThree = new Car(3,"Audi","a6",2009,"red",21000,"1111");  
 Cars.add(carThree);  
 Car carFour = new Car(4,"Mercedes","212",2010,"white",21000,"2222");  
 Cars.add(carFour);  
 Car carFive = new Car(5,"Mercedes","211",2009,"white",25000,"4444");  
 Cars.add(carFive);  
 Car carSix = new Car(6,"Mercedes","213",2008,"grey",21000,"6666");  
 Cars.add(carSix);  
 }  
}