МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

по дисциплине «Технологии разработки программного обеспечения»

# на тему: «Развитие представлений о разработке программ. Объектно-ориентированной программирование в Java»

Выполнил: студент гр. ИТП-21

Рожина А.К.

Принял: преподаватель

Малиновский И.Л.

Гомель 2023

**Цель работы:** ознакомиться с основными принципами объектно-ориентированного программирования и с их реализациями на языке Java.

**Практическое задание**

**Задание.**

1. Разработать UML-диаграмму иерархии классов, согласно варианта (таблица 1).

2. При наименовании компонентов руководствоваться соглашением о наименовании.

3. При описании иерархии использовать наследование и композицию.

4. На основе UML-диаграммы разработать иерархию классов на языке Java.

5. Весь код должен быть снабжен элементами документирования.

6. Разработанную иерархию поместить в .jar файл для дальнейшего использования в качестве библиотечных классов.

7. Создать консольное приложение для демонстрации работы созданных классов.

8. Составить отчет о проделанной работе.

Таблица 1

|  |  |
| --- | --- |
| **Вариант** | Условие задачи |
| 3 | 1.1 Создать класс Car, Engine и Driver.  1.2 Класс Driver содержит поля - ФИО, стаж вождения.  1.3 Класс Engine содержит поля - мощность, производитель.  1.4 Класс Car содержит поля - марка автомобиля, класс автомобиля, вес, водитель типа Driver, мотор типа Engine.  1.5 Вывести водителей со стажем более 5 лет.  1.6 Вывести автомобили советского производства |

**Ход работы**

**Порядок выполнения задания.**

На рисунке 1 показана UML-диаграмма задания.

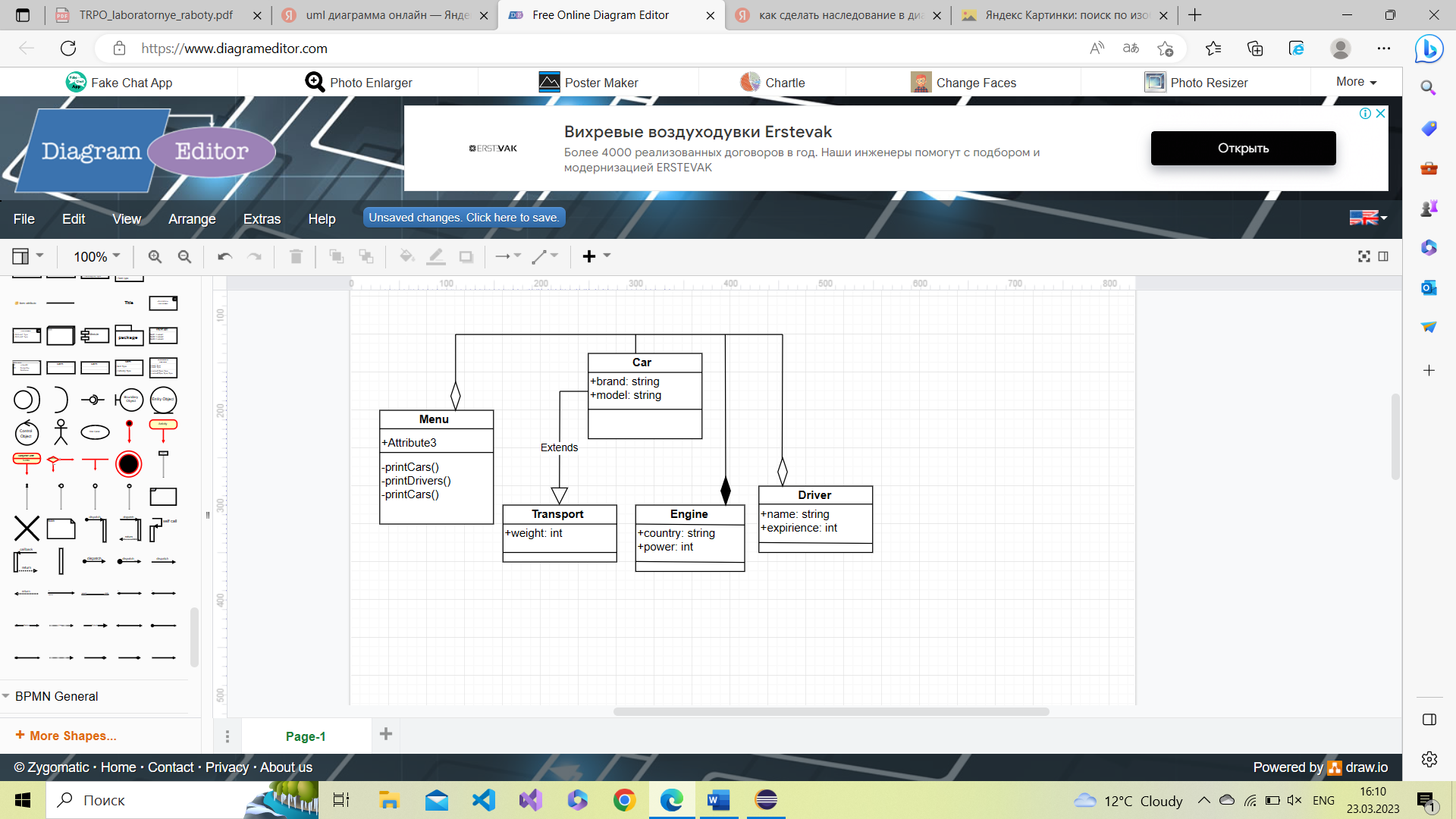


Рисунок 1 – UML-диаграмма поставленной задачи

На рисунке 2 показано меню основных функций

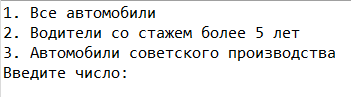


Рисунок 2 – Меню основных функций программы

На рисунке 3 показан результат выполнения пункта 1 меню

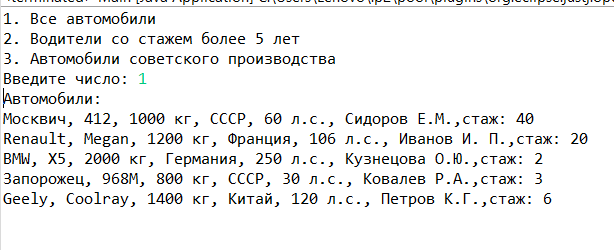


Рисунок 3 – Все автомобили

На рисунке 4 показан вывод водителей со стажем более 5 лет

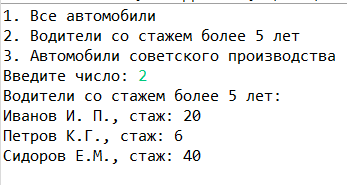


Рисунок 4 – вывод водителей со стажем более 5 лет

На рисунках 5 показан вывод автомобилей советского производства

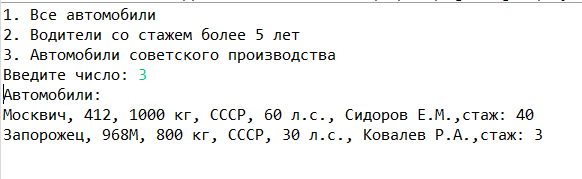


Рисунок 5 – вывод автомобилей советского производства

**Вывод:** в процессе работы ознакомились с основными принципами объектно-ориентированного программирования, а также с созданием классов и объектов на языке Java.

**ПРИЛОЖЕНИЕ А**

(обязательное)

**Текст программы**

**Main.java:**

package lab1;

import java.util.Scanner;

import library.Car;

import library.Driver;

public class Main {

public static void main(String[] args) {

String country = "СССР";

int experience = 5;

System.out.println("1. Все автомобили");

System.out.printf("2. Водители со стажем более %d лет", experience);

System.out.println();

System.out.println("3. Автомобили советского производства");

System.out.print("Введите число: ");

Scanner in = new Scanner(System.in);

int number = in.nextInt();

in.close();

var data = new Data();

switch (number) {

case 1:

printCars(data.getAllCars());

break;

case 2:

printDrivers(data.getDriversWithExperience(experience), experience);

break;

case 3:

printCars(data.getCarsFromCountry(country));

break;

default:

System.out.println("Неверные данные.");

}

}

private static void printCars(Car[] cars) {

System.out.println("Автомобили: ");

for (int i = 0; i < cars.length; i++) {

if (cars[i] != null) {

System.out.printf("%s, %s, %d кг, %s, %d л.с., %s,стаж: %d", cars[i].brand, cars[i].model, cars[i].weight, cars[i].getEngine().country, cars[i].getEngine().power, cars[i].getDriver().name, cars[i].getDriver().experience);

System.out.println();

}

}

}

private static void printDrivers(Driver[] drivers, int experience) {

System.out.printf("Водители со стажем более %d лет: ", experience);

System.out.println();

int i = 0;

while (drivers[i] != null) {

System.out.printf("%s, стаж: %d", drivers[i].name, drivers[i].experience);

System.out.println();

i++;

}

}

}

**Data.java:**

**package** lab1;

**import** library.Driver;

**import** library.Engine;

**import** library.Car;

**public** **class** Data {

Driver[] drivers;

Engine[] engines;

Car[] cars;

**public** Data(){

loadDrivers();

loadEngines();

loadCars();

}

**void** loadDrivers(){

**this**.drivers = **new** Driver[]{

**new** Driver("Иванов И. П.", 20),

**new** Driver("Петров К.Г.", 6),

**new** Driver("Сидоров Е.М.", 40),

**new** Driver("Ковалев Р.А.", 3),

**new** Driver("Кузнецова О.Ю.", 2)

};

}

**void** loadEngines() {

**this**.engines = **new** Engine[]{

**new** Engine("СССР", 60),

**new** Engine("Германия", 250),

**new** Engine("Франция", 106),

**new** Engine("СССР", 30),

**new** Engine("Китай", 120)

};

}

**void** loadCars() {

**var** car1 = **new** Car("Москвич", "412", 1000);

car1.setDriver(**this**.drivers[2]);

car1.setEngine(**this**.engines[0]);

**var** car2 = **new** Car("Renault", "Megan", 1200);

car2.setDriver(**this**.drivers[0]);

car2.setEngine(**this**.engines[2]);

**var** car3 = **new** Car("BMW", "X5", 2000);

car3.setDriver(**this**.drivers[4]);

car3.setEngine(**this**.engines[1]);

**var** car4 = **new** Car("Запорожец", "968М", 800);

car4.setDriver(**this**.drivers[3]);

car4.setEngine(**this**.engines[3]);

**var** car5 = **new** Car("Geely", "Coolray", 1400);

car5.setDriver(**this**.drivers[1]);

car5.setEngine(**this**.engines[4]);

**this**.cars = **new** Car[] {car1, car2, car3, car4, car5};

}

**public** Car[] getAllCars() {

**return** **this**.cars;

};

**public** Driver[] getDriversWithExperience(**int** exp) {

Driver[] filteredDrivers = **new** Driver[**this**.drivers.length];

**int** j = 0;

**for** (**int** i = 0; i < **this**.drivers.length; i++) {

**if** (**this**.drivers[i].experience > exp) {

filteredDrivers[j] = **this**.drivers[i];

j++;

}

}

**return** filteredDrivers;

}

**public** Car[] getCarsFromCountry(String country) {

Car [] filteredCars = **new** Car[**this**.cars.length];

**int** j = 0;

**for** (**int** i = 0; i < **this**.cars.length; i++) {

**if** (**this**.cars[i].getEngine().country == country) {

filteredCars[j] = **this**.cars[i];

j++;

}

}

**return** filteredCars;

}

}

**Car.Java**

**package** library;

**public** **class** Car **extends** Transport {

**public** String brand;

**public** String model;

**private** Engine engine;

**private** Driver driver;

**public** Car (String b, String m, **int** w){

**super**(w);

brand = b;

model = m;

}

**public** Engine getEngine() {

**return** engine;

}

**public** **void** setEngine(Engine engine) {

**this**.engine = engine;

}

**public** Driver getDriver() {

**return** driver;

}

**public** **void** setDriver(Driver driver) {

**this**.driver = driver;

}

}

**Driver.java**

**package** library;

**public** **class** Driver {

**public** String name;

**public** **int** experience;

**public** Driver (String n, **int** e){

name = n;

experience = e;

}

}

**Engine.java**

**package** library;

**public** **class** Engine {

**public** String country;

**public** **int** power;

**public** Engine (String c, **int** p){

country = c;

power = p;

}

}

**Transport.java**

package library;

public class Transport {

public int weight;

public Transport (int w){

weight = w;

}

}