**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Факультет Информационных технологий**

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 4**

**Дисциплина:** Технологии кроссплатформенного программирования

**Тема:** ООП в Java.

**Выполнила: студентка группы 211-725**

Алюбаева Карина Ислямбековна

**Дата**  03.10.2023

**Проверил: \_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_***

(Фамилия И.О., степень, звание) **(Оценка)**

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Москва**

**2023**

Отчет по 4-ой лабораторной.

1. Добавить гаражу возможность удаления из него машины, а к классу Autoдобавить поле с гос номером.

Изменения в классе Auto влияют на работу и код всех классов, я везде внесла изменения в строке с конструктором. Добавленные или измененные мной строки выделены желтым маркером.

**//класс Auto**

public class Auto {

    private String firm; // создаем закрытый член нашего класса с названием фирмы автомобиля

    private int maxSpeed; // закрытый член класса, содержащий максимальную скорость

    private String gosNumber; //гос номер, закрытый член класса

    public void setFirm(String firma) { // открытая функция (метод класса) для задания

        firm = firma; // значения фирмы автомобиля

    }

    public void setMaxSpeed(int speed) { // открытая функция (метод класса) для задания

        maxSpeed = speed; // значения максимальной скорости автомобиля

    }

    public void setGosNumber(String number) { // открытая функция (метод класса) для задания

        gosNumber = number; // значения гос номера

    }

    public int getMaxSpeed() { // открытая функция (метод класса) для вывода значения

        return maxSpeed; // максимальной скорости

    }

    public String getFirm() { // открытая функция (метод класса) для вывода значения

        return firm; // заданной фирмы

    }

    public String getGosNumber() { // открытая функция (метод класса) для вывода значения

        return gosNumber; // гос номера

    }

    public Auto() { // конструктор класса (без параметров)

        firm = "Без названия";

        maxSpeed = 0;

        gosNumber = "AAA000-00";

    }

    public Auto(String firma, int speed, String number) { // конструктор класса (с параметрами)

        firm = firma;

        maxSpeed = speed;

        gosNumber = number;

    }

}

**//класс Car**

public class Car extends Auto {// файл Car.java

    private String model;

    private int numDoors;

    private Boolean fullTime; // полный привод

    public Car() {

        super();// вызываем конструктор класса-родителя без параметров (см. класс Auto)

        model = ""; // добавляем инициализацию новых членов

        numDoors = 4;

        fullTime = false;

    }

    public Car(String firma, int speed, String number, String name, int n, Boolean f) {

        super(firma, speed, number); // вызываем конструктор класса-родителя с параметрами (см. класс Auto)

        model = name; // добавляем инициализацию новых членов

        numDoors = n;

        fullTime = f;

    }

    public void setModel(String name) {

        model = name;

    }

    public String getModel() {

        return model;

    }

    public void setNumDoors(int n) {

        numDoors = n;

    }

    public int getNumDoors() {

        return numDoors;

    }

    public void setFullTime(Boolean b) {

        fullTime = b;

    }

    public Boolean isFullTime() {

        return fullTime;

    }

public String toString(){ return getFirm() + " " + getMaxSpeed() + " " + model + " " + numDoors + " " + fullTime;

}

}

**//класс Truck**

import java.util.Scanner;

public class Truck extends Auto {

    private String model;

    private int power;

    private Boolean trailer;

    // c прицепом или без

    public Truck() {

        super();

        model = "";

        power = 0;

        trailer = false;

    }

    public Truck(String firma, int speed, String number, String name, int n, Boolean f) {

        super(firma, speed, number);

        model = name;

        power = n;

        trailer = f;

    }

    public void setModel(String name) {

        model = name;

    }

    public String getModel() {

        return model;

    }

    public void setPower(int n) {

        power = n;

    }

    public int getPower() {

        return power;

    }

    public void setTrailer(Boolean b) {

        trailer = b;

    }

    public Boolean isTrailer() {

        return trailer;

    }

    public void setAllInfo() {

        try (Scanner scanner = new Scanner(System.in)) {

            System.out.print("Введите фирму-производитель грузового авто: ");

            String nazv = scanner.next(); // метод next() позволяет вводить строки, но без пробелов

            setFirm(nazv);

            System.out.print("Введите максимальную скорость грузового авто: ");

            int s = scanner.nextInt();

            setMaxSpeed(s);

            System.out.print("Введите модель грузового авто: ");

            model = scanner.next();

            System.out.print("Введите мощность грузового авто: ");

            power = scanner.nextInt();

            System.out.print("Введите признак прицепа грузового авто (true/false): ");

            trailer = scanner.nextBoolean();

        }

        System.out.println();

    }

    public String toString() {

        return "\n\tГрузовик" + "\n\t" + "Фирма: " + getFirm() + "\n\t" + "Максимальная скорость: "

                + getMaxSpeed() + "\n\t" + "Модель: " + model + "\n\t" + "Мощность: " + power + "\n\t"

                + "Признак прицепа: " + trailer + "\n";

    }

}

**//класс GarageCar с добавленным методом удаления машины из списка**

import java.util.ArrayList;//нужно для работы с классом ArrayList

public class GarageCar {

    private ArrayList<Auto> masCar = new ArrayList<Auto>();// массив с машинами

    public void addCar(Auto m) {// метод для добавления машины в гараж

        masCar.add(m);

    }

    public void removeCar(Auto m) {

        masCar.remove(m);

    }

    public GarageCar() {

    }

    public Boolean findCar(Auto m) { // для выяснения – есть ли машина m в гараже

        return masCar.contains(m);

    }

    public GarageCar(ArrayList<Auto> n) {// конструктор для внесения существующего списка машин в гараж

        masCar = n;

    }

    public void printGarage() { // для вывода на экран списка машин в гараже

        System.out.println("В гараже: ");

        for (Auto a : masCar) {

            System.out.println("\t" + a.toString());

        }

    }

}

//тест

public class test {

    public static void main(String[] args) {

        Auto myAuto1 = new Auto("Ford", 180, "AAA000-55"); // создаем объект типа нашего класса

        System.out.println(myAuto1.getFirm() + " " + myAuto1.getMaxSpeed() + " " + myAuto1.getGosNumber());// вывод сведений в

    } // консоль

}

Результат работы программы тест:



//тест гараж

public class testGarage {

    public static void main(String[] args) {

        GarageCar myGarage = new GarageCar(); // создаем новый гараж

        Car myCar1 = new Car("Ford", 200, "aaa000", "Mustang", 2, false); // создаем легковую машину

        myGarage.addCar(myCar1); // добавляем ее в гараж

        myGarage.addCar(new Car("LADA", 140, "bbb010", "Kalina", 4, false));// добавляем еще одну машину

        Truck myTruck = new Truck("Dove", 160, "ccc100", "DTS", 700, true);// создаем грузовик

        myGarage.addCar(myTruck); // добавляем его в гараж

        myGarage.printGarage(); // выводи на экран содержимое гаража

        if (myGarage.findCar(myCar1)) { // ищем машину

            System.out.println("Да");

        } else {

            System.out.println("Нет");

        }

        //проверка работы метода удаления машины из гаража

        myGarage.removeCar(myCar1);

        if (myGarage.findCar(myCar1)) { // ищем машину

            System.out.println("Да");

        } else {

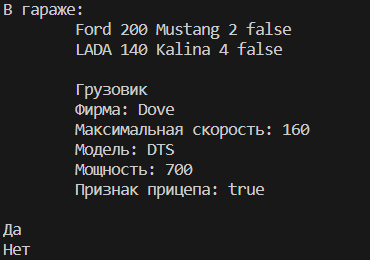
            System.out.println("Нет");

        }

    }

}

Результат работы программы:



После удаления машины из гаража, ее нельзя найти, значит она удалена.

2. Базовый класс – учащийся. Производные – школьник и студент. Создать класс Конференция, который может содержать оба вида учащихся. Предусмотреть метод подсчета участников конференции отдельно по школьникам и по студентам (использовать оператор instanceof).

Я создала класс Учащийся:

public class Student {

    private String name;

    public Student(String n) {

        name = n;

    }

    public String getName() {

        return name;

    }

}

И два производных от него – школьник и студент:

public class Schoolboy extends Student {

    private int grade;

    public Schoolboy(String name, int grade) {

        super(name);

        this.grade = grade;

    }

    public int getGrade() {

        return grade;

    }

}

public class CollegeStudent extends Student {

    private String major;

    public CollegeStudent(String n, String major) {

        super(n); //конструктор родительского класса

        this.major = major;

    }

    public String getMajor() {

        return major;

    }

}

К каждому производному классу добавила отличительные черты (номер класс и специальность)

Далее создала класс конференции со всеми, как мне показалось, необходимыми методами. Участники конференции будут записаны в список participants.

package Part2;

import java.util.ArrayList;

public class Conference {

    private ArrayList<Student> participants = new ArrayList<>();

    // Метод добавления студента (школьника или студента) в конференцию

    public void addParticipant(Student student) {

        participants.add(student);

    }

    // Метод подсчета школьников на конференции

    public int countSchoolboys() {

        int count = 0;

        for (Student participant : participants) {

            if (participant instanceof Schoolboy) {

                count++;

            }

        }

        return count;

    }

    // Метод подсчета студентов на конференции

    public int countCollegeStudents() {

        int count = 0;

        for (Student participant : participants) {

            if (participant instanceof CollegeStudent) {

                count++;

            }

        }

        return count;

    }

}

Далее создала класс, в котором протестировала работу программы:

package Part2;

public class ConferenceTest {

    public static void main(String[] args) {

        // Создание экземпляров школьников и студентов

        Schoolboy schoolboy1 = new Schoolboy("Маша", 8);

        Schoolboy schoolboy2 = new Schoolboy("Петя", 7);

        CollegeStudent student1 = new CollegeStudent("Егор", "Программист");

        CollegeStudent student2 = new CollegeStudent("Марина", "Летчик");

        // Создание конференции

        Conference conference = new Conference();

        // добавление участников

        conference.addParticipant(schoolboy1);

        conference.addParticipant(schoolboy2);

        conference.addParticipant(student1);

        conference.addParticipant(student2);

        // Подсчет

        int schoolboyCount = conference.countSchoolboys();

        int collegeStudentCount = conference.countCollegeStudents();

        System.out.println("Школьников: " + schoolboyCount);

        System.out.println("Студентов: " + collegeStudentCount);

    }

}

Вот результат:

