

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Московский институт электроники и математики им. А. Н. Тихонова

Департамент компьютерной инженерии

Отчет по практическим работам №2 и №3
«Объектная модель в Java»

по курсу «Распределенные базы данных и сетевые вычисления»
Вариант №13

Выполнили:

Студенты группы МКС212

Журсунова Найля

Шабалина Анастасия Владимировна

Приняла:

Байбикова Татьяна Николаевна

Москва, 2022 г.

Оглавление

| | |
|---------------------------|----|
| 1 Задание | 3 |
| 2 Листинг..... | 4 |
| 3 Результаты работы | 10 |
| 4 Выводы | 11 |

1 Задание

В ходе работы необходимо изучить материал к данной работе и подготовить отчет. Для выполнения задания нами был взят класс Automobile.

В разработанной программе (наборе программ) нужно отобразить следующие моменты:

- конструкторы и их перегрузку;
- перегрузку методов;
- наследование (двух видов - быть чем-то и агрегацию);
- виртуальные методы;
- абстрактные классы;
- полиморфизм;
- интерфейсы;

Работа будет выполняться на языке программирования Java.

2 Листинг

2.1 Интерфейс Auto

```
package com.ddbase.auto;

public interface Auto {
    public void setMileage(int mileage); //функция установки пробега авто
    public void increaseMileage(int mileage);
    public void increaseMileage(); //функция увеличения пробега
    public int getMileage(); //функция получения пробега
    public void setColor(String color); //функция установки цвета авто
    public String getColor(); //функция получения цвета
}
```

2.2 Абстрактный класс Automobile реализующий интерфейс Auto

```
package com.ddbase.auto;

public abstract class Automobile implements Auto {
    protected String brand; //марка автомобиля
    protected int release; //год выпуска
    protected int carCapacity; //кол-во мест
    protected String fuelType; //тип топлива
    protected String gearBox; //тип коробки передач
    protected String drive; //тип привода
    protected int tankCapacity; //емкость топливного бака, л
    protected int mileage = 0; //пробег, км
    protected String color; //цвет автомобиля

    public Automobile() {
        this.brand = "Toyota";
        this.release = 2016;
        this.carCapacity = 5;
        this.fuelType = "Бензин";
        this.gearBox = "Автомат";
        this.drive = "Передний";
        this.tankCapacity = 50;
    }

    //перегрузка конструктора
    public Automobile(String brand, int release, int carCapacity,
                      String fuelType, String gearBox, String drive,
                      int tankCapacity)
    {
        this.brand = brand;
        this.release = release;
        this.carCapacity = carCapacity;
        this.fuelType = fuelType;
    }
}
```

```

        this.gearBox = gearBox;
        this.drive = drive;
        this.tankCapacity = tankCapacity;
    }

    //реализация методов интерфейса
    public void setMileage(int mileage) {
        this.mileage = mileage;
    }

    public void increaseMileage(int mileage) {
        this.mileage += mileage;
    }

    //перегрузка метода
    public void increaseMileage() {
        this.mileage += 100;
    }

    public int getMileage() {
        return this.mileage;
    }

    public void setColor(String color) {
        this.color = color;
    }

    public String getColor() {
        return this.color;
    }

    //виртуальный метод
    //функция печати информации про автомобиль
    public void printInfo() {
        System.out.println("Информация про класс Автомобиль!");
    }

    //функция вычисления возраста автомобиля
    public abstract int autoYear(); //абстрактный метод
}

```

2.3 Класс Engine

```

package com.ddbase.engine;

public class Engine
{
    protected int rpm; //количество оборотов в минуту, об/мин
    protected double mep; //среднее эффективное давление, МПа
    public int capacity; //объем двигателя, см3
}

```

```

public Engine() {
    this.rpm = 2000;
    this.mep = 0.85;
    this.capacity = 1500;
}

//перегрузка конструктора
public Engine(int rpm, double mep, int capacity) {
    this.rpm = rpm;
    this.mep = mep;
    this.capacity = capacity;
}

//функция вычисления мощности двигателя
public double enginePower() {
    return (this.capacity * this.mep * (this.rpm/120));
}
}

```

2.4 Класс PassengerCar наследуемый от Automobile

```

package com.ddbase.autotype;

import java.util.Calendar;
import java.util.Date;
import com.ddbase.auto.*;
import com.ddbase.engine.Engine;

//наследование от класса Automobile (быть чем-то)
public class PassengerCar extends Automobile {
    protected String model; //модель легкового автомобиля
    protected String bodyType; //тип кузова
    protected double width; //ширина кузова, м
    protected double length; //длина кузова, м

    //наследование с использованием агрегации
    public Engine engine = new Engine();

    public PassengerCar() {
        super();
        this.model = "Corolla";
        this.width = 1.78;
        this.length = 4.63;
        this.bodyType = "Седан";
    }

    //перегрузка конструктора
    public PassengerCar(String brand, int release, int carCapacity,
                        String fuelType, String gearBox, String drive,
                        int tankCapacity, String model, String bodyType,

```

```

        double width, double length) {
    super(brand, release, carCapacity, fuelType, gearBox, drive,
        tankCapacity);
    this.model = model;
    this.width = width;
    this.length = length;
    this.bodyType = bodyType;
}

    public String carClass() { //функция определения класса легкового автомобиля
        String cl;
        if (this.width < 1.6 & this.length < 3.6)
            cl = "Класс А - особо малый класс.";
        else if ((this.width >= 1.5 & this.width <= 1.7) & (this.length >= 3.6 &
this.length < 3.9))
            cl = "Класс В - малый класс.";
        else if ((this.width >= 1.6 & this.width <= 1.75) & (this.length >= 3.9 &
this.length < 4.4))
            cl = "Класс С - первый средний класс.";
        else if ((this.width >= 1.7 & this.width < 1.8) & (this.length >= 4.4 &
this.length < 4.8))
            cl = "Класс D - второй средний класс.";
        else if (this.width >= 1.8 & (this.length >= 4.8 & this.length < 5))
            cl = "Класс Е - бизнес-класс.";
        else
            cl = "Класс F - представительский класс.";
        return cl;
    }

    @Override
    public int autoYear() { //переопределение метода
        return Calendar.getInstance().get(Calendar.YEAR) - this.release;
    }

    //Виртуальный метод в наследуемом классе
    public void printInfo() {
        System.out.println("Марка машины: " + this.brand + ". Модель: " + this.model
+ ". Год выпуска: " + this.release + " г. (" + autoYear() + " лет).");
        System.out.println("Вместимость: " + this.carCapacity + ". Цвет: " +
getColor() + ". Тип коробки передач: " + this.gearBox + ". Тип привода: " +
this.drive + ".");
        System.out.println("Тип топлива: " + this.fuelType + ". Емкость топливного
бака: " + this.tankCapacity + " л.");
        System.out.println("Объем двигателя: " + engine.capacity + " см3. Мощность
двигателя: " + engine.enginePower() + " кВт.");
        System.out.println("Длина кузова: " + this.length + " м. Ширина кузова: " +
this.width + "м. Тип кузова: " + this.bodyType + ".");
        System.out.println(carClass());
        System.out.println("Пробег: " + getMileage() + " км.");
    }
}

```

2.5 Класс Truck наследуемый от Automobile

```
package com.ddbase.autotype;

import java.util.Calendar;
import java.util.Date;
import com.ddbase.auto.*;
import com.ddbase.engine.Engine;

//наследование от класса Automobile (быть чем-то)
public class Truck extends Automobile {
    protected String model; //модель грузового автомобиля
    protected String bodyType; //тип кузова
    protected int loadCapacity; //грузоподъемность, т
    protected int numberChassis; //кол-во осей

    //наследование с использованием агрегации
    public Engine engine = new Engine(1800, 0.85, 6000);

    public Truck(String brand, int release, int carCapacity,
                  String fuelType, String transmission, String drive,
                  int tankCapacity, String model, String bodyType,
                  int loadCapacity, int numberChassis) {
        super(brand, release, carCapacity, fuelType, transmission, drive,
              tankCapacity);
        this.model = model;
        this.loadCapacity = loadCapacity;
        this.bodyType = bodyType;
        this.numberChassis = numberChassis;
    }

    //функция разбиения грузовых автомобилей по грузоподъемности
    public String truckLoadCapacity() {
        String LoadCapacity;
        if (this.loadCapacity < 2)
            LoadCapacity = "Малая грузоподъемность";
        else if (this.loadCapacity >= 2 & this.loadCapacity < 5)
            LoadCapacity = "Средняя грузоподъемность";
        else if (this.loadCapacity >= 5 & this.loadCapacity < 16)
            LoadCapacity = "Большая грузоподъемность";
        else
            LoadCapacity = "Особо большая грузоподъемность";
        return LoadCapacity;
    }

    @Override
    public int autoYear() {
        return Calendar.getInstance().get(Calendar.YEAR) - this.release;
    }
}
```



```

        //Виртуальный метод в качестве полиморфизма
        public void printInfo() {
            System.out.println("Марка машины: " + this.brand + ". Модель: " + this.model
+ ". Год выпуска: " + this.release + " г. (" + autoYear() + " лет).");
            System.out.println("Вместимость: " + this.carCapacity + ". Цвет: " +
getColor() + ". Тип коробки передач: " + this.gearBox + ". Тип привода: " +
this.drive + ".");
            System.out.println("Тип топлива: " + this.fuelType + ". Емкость топливного
бака: " + this.tankCapacity + " л.");
            System.out.println("Объем двигателя: " + engine.capacity + " см3. Мощность
двигателя: " + engine.enginePower() + " кВт.");
            System.out.println("Количество осей: " + this.numberChassis + ".
Грузоподъемность: " + this.loadCapacity + "т (" + truckLoadCapacity() + ").");
            System.out.println("Пробег: " + getMileage() + " км.");
        }
    }
}

```

2.6 Класс Main

```

package com.ddbase.main;

import com.ddbase.auto.Automobile;
import com.ddbase.autotype.*;

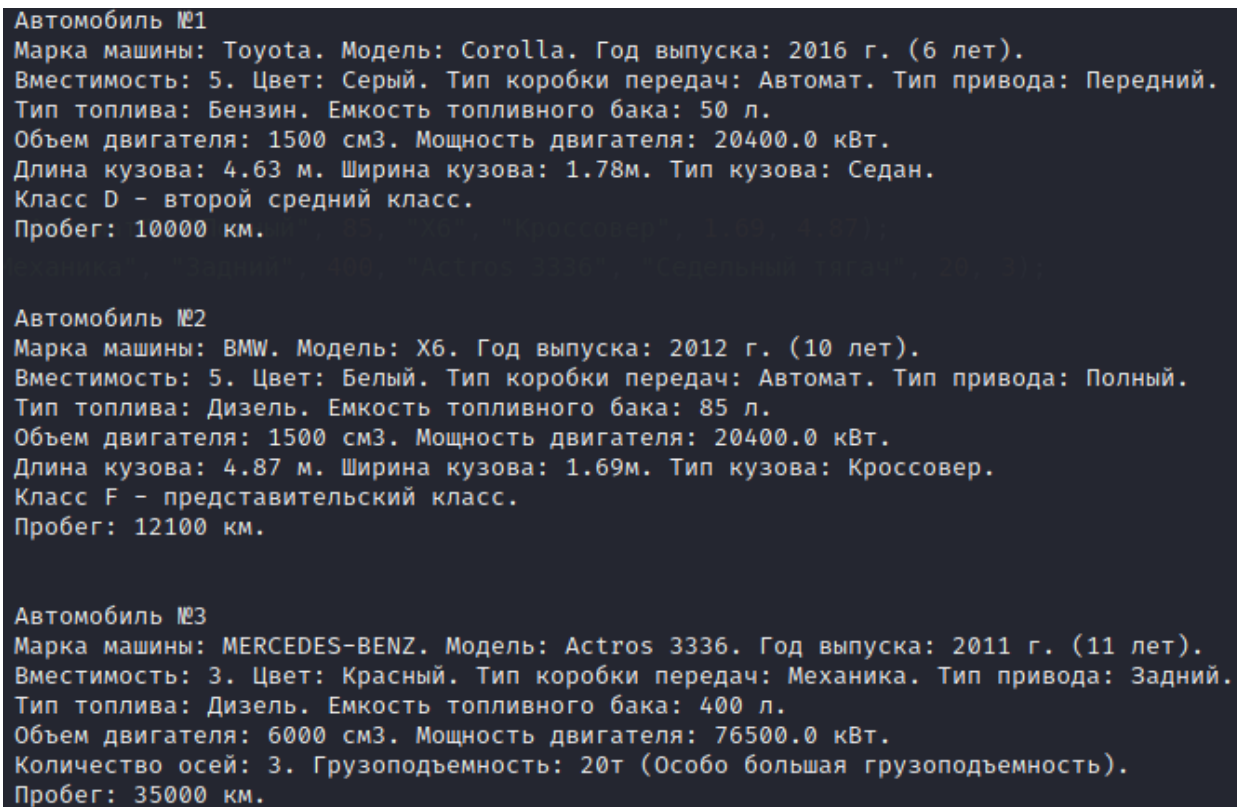
public class Main {
    public static void main(String[] args) {
        PassengerCar car1 = new PassengerCar(); //вызов конструктора по умолчанию
        //вызов конструктора с параметрами
        PassengerCar car2 = new PassengerCar("BMW", 2012, 5, "Дизель", "Автомат",
"Полный", 85, "X6", "Кроссовер", 1.69, 4.87);
        Truck truck1 = new Truck("MERCEDES-BENZ", 2011, 3, "Дизель", "Механика",
"Задний", 400, "Actros 3336", "Седельный тягач", 20, 3);
        car1.setMileage(10000);
        car1.setColor("Серый");
        car2.setMileage(12000);
        car2.increaseMileage();
        car2.setColor("Белый");
        truck1.setMileage(30000);
        truck1.increaseMileage(5000);
        truck1.setColor("Красный");

        Automobile[] auto = new Automobile[] {car1, car2, truck1};
        for(int i = 0; i < auto.length; i++) {
            System.out.println("Автомобиль №" + (i+1));
            auto[i].printInfo(); // полиморфный вызов метода
            System.out.println("\n");
        }
    }
}

```

3 Результаты работы

По результатам выполнения программы были получены результаты, представленные на Рисунке 1.



Автомобиль №1
Марка машины: Toyota. Модель: Corolla. Год выпуска: 2016 г. (6 лет).
Вместимость: 5. Цвет: Серый. Тип коробки передач: Автомат. Тип привода: Передний.
Тип топлива: Бензин. Емкость топливного бака: 50 л.
Объем двигателя: 1500 см3. Мощность двигателя: 20400.0 кВт.
Длина кузова: 4.63 м. Ширина кузова: 1.78м. Тип кузова: Седан.
Класс D – второй средний класс.
Пробег: 10000 км.

Автомобиль №2
Марка машины: BMW. Модель: X6. Год выпуска: 2012 г. (10 лет).
Вместимость: 5. Цвет: Белый. Тип коробки передач: Автомат. Тип привода: Полный.
Тип топлива: Дизель. Емкость топливного бака: 85 л.
Объем двигателя: 1500 см3. Мощность двигателя: 20400.0 кВт.
Длина кузова: 4.87 м. Ширина кузова: 1.69м. Тип кузова: Кроссовер.
Класс F – представительский класс.
Пробег: 12100 км.

Автомобиль №3
Марка машины: MERCEDES-BENZ. Модель: Actros 3336. Год выпуска: 2011 г. (11 лет).
Вместимость: 3. Цвет: Красный. Тип коробки передач: Механика. Тип привода: Задний.
Тип топлива: Дизель. Емкость топливного бака: 400 л.
Объем двигателя: 6000 см3. Мощность двигателя: 76500.0 кВт.
Количество осей: 3. Грузоподъемность: 20т (Особо большая грузоподъемность).
Пробег: 35000 км.

Рисунок 1 – Результаты работы программы

4 Выводы

При выполнении практической работы были получены навыки работы с основными принципами ООП с использованием высокоуровневого языка программирования Java. Были освоены навыки создания классов и конструкторов для них с возможной их перегрузкой. В дополнении были разработаны методы с использованием перегрузки для предоставления более детального контроля будущему разработчику. Для организованной работы с наследниками класса был создан виртуальный метод, который является механизмом реализации полиморфизма.