# Практические задания

**День 2**. **ООП**

1. **Класс Phone.**

Создайте класс Phone, который содержит переменные number, model и weight.

Создайте три экземпляра этого класса.

Выведите на консоль значения их переменных.

Добавить в класс Phone методы: receiveCall, имеет один параметр – имя звонящего. Выводит на консоль сообщение “Звонит {name}”. getNumber – возвращает номер телефона. Вызвать эти методы для каждого из объектов.

Добавить конструктор в класс Phone, который принимает на вход три параметра для инициализации переменных класса - number, model и weight.

Добавить конструктор, который принимает на вход два параметра для инициализации переменных класса - number, model.

Добавить конструктор без параметров.

Вызвать из конструктора с тремя параметрами конструктор с двумя.

Добавьте перегруженный метод receiveCall, который принимает два параметра - имя звонящего и номер телефона звонящего. Вызвать этот метод.

Создать метод sendMessage с аргументами переменной длины. Данный метод принимает на вход номера телефонов, которым будет отправлено сообщение. Метод выводит на консоль номера этих телефонов.

Реализовать подсчет количества созданных телефонов (класс Phone) с помощью статической переменной.

1. **Создать класс Person, который содержит:**

переменные fullName, age;

методы move() и talk(), в которых просто вывести на консоль сообщение -"Такой-то Person говорит".

Добавьте два конструктора - Person() и Person(fullName, age).

Создайте два объекта этого класса. Один объект инициализируется конструктором Person(), другой - Person(fullName, age).

1. **Автомобили**

Создать класс Car в пакете com.company.vehicles, Engine в пакете com.company.details и Driver в пакете com.company.professions.

Класс Driver содержит поля - ФИО, стаж вождения.

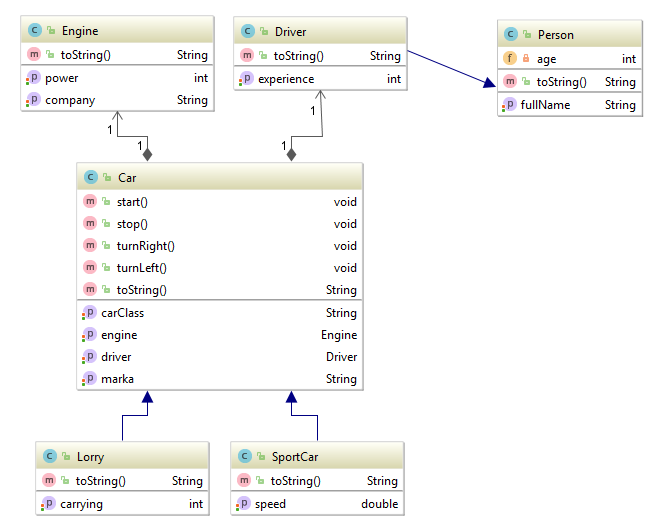
Класс Engine содержит поля - мощность, производитель.

Класс Car содержит поля - марка автомобиля, класс автомобиля, вес, водитель типа Driver, мотор типа Engine. Методы start(), stop(), turnRight(), turnLeft(), которые выводят на печать: "Поехали", "Останавливаемся", "Поворот направо" или "Поворот налево". А также метод toString(), который выводит полную информацию об автомобиле, ее водителе и моторе.

Создать производный от Car класс - Lorry (грузовик), характеризуемый также грузоподъемностью кузова.

Создать производный от Car класс - SportCar, характеризуемый также предельной скоростью.

Пусть класс Driver расширяет класс Person.



1. **Интернет магазин**

Создать класс Товар, имеющий переменные имя, цена, рейтинг.

Создать класс Категория, имеющий переменные имя и массив товаров. Создать несколько объектов класса Категория.

Создать класс Basket, содержащий массив купленных товаров.

Создать класс User, содержащий логин, пароль и объект класса Basket. Создать объект класса User.

1. **Фруктовая лавка**

Создать абстрактный класс Фрукт и классы Яблоко, Груша, Абрикос расширяющие его.

Класс Фрукт содержит:

а) переменную вес,

б) завершенный метод printManufacturerInfo(){System.out.print("Made in Russia");}

в) абстрактный метод, возвращающий стоимость фрукта, который должен быть переопределен в каждом классе наследнике. Метод должен учитывать вес фрукта.

Создать несколько объектов разных классов.

Подсчитать общую стоимость проданных фруктов.

А также общую стоимость отдельно проданных яблок, груш и абрикос.

1. **Интерфейс Инструмент**

Создать интерфейс Инструмент и реализующие его классы Гитара, Барабан и Труба.

Интерфейс Инструмент содержит метод play() и переменную String KEY ="До мажор".

Гитара содержит переменные класса количествоСтрун, Барабан - размер, Труба - диаметр.

Создать массив типа Инструмент, содержащий инструменты разного типа.

В цикле вызвать метод play() для каждого инструмента, который должен выводить строку "Играет такой-то инструмент, с такими-то характеристиками".