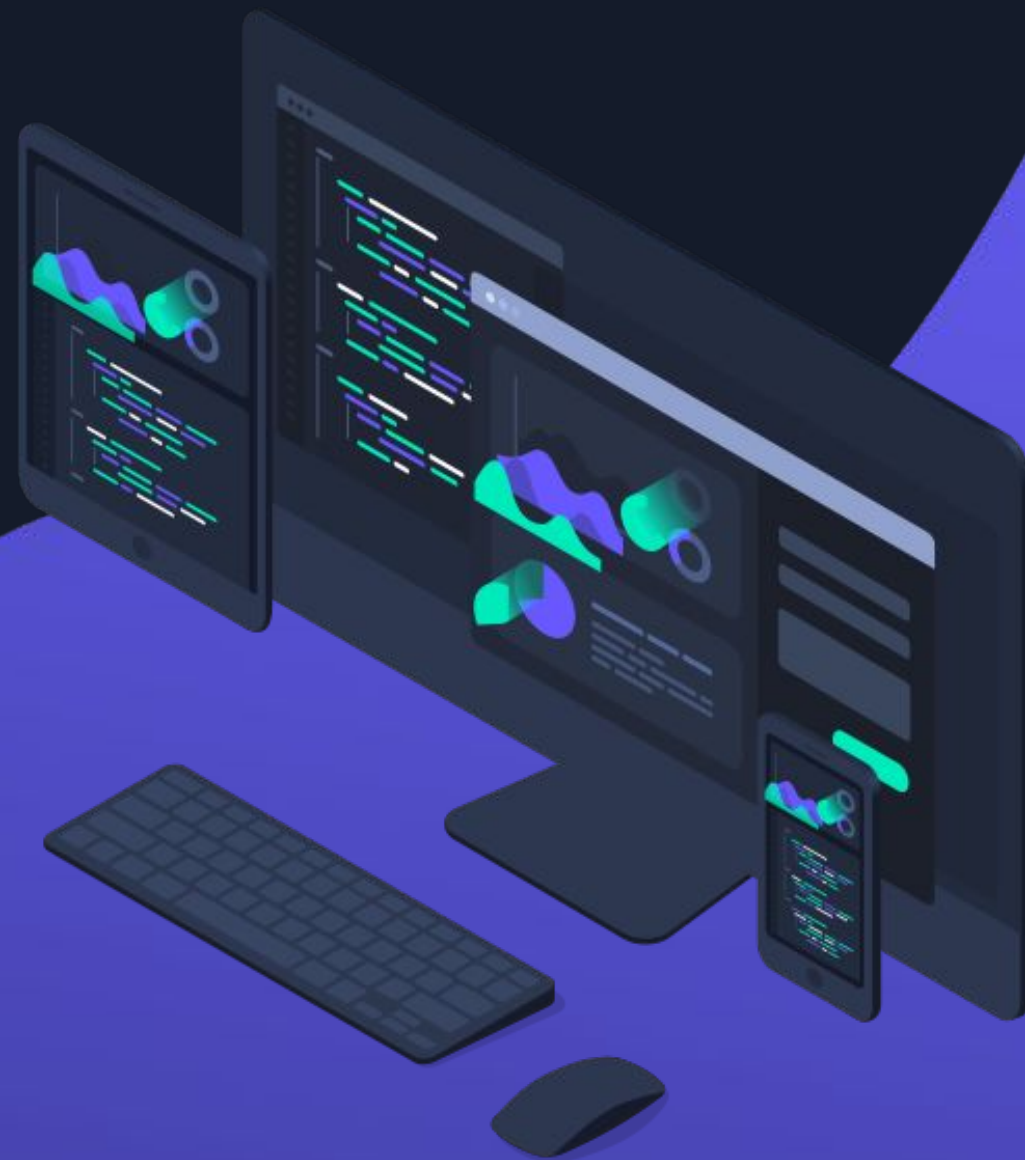


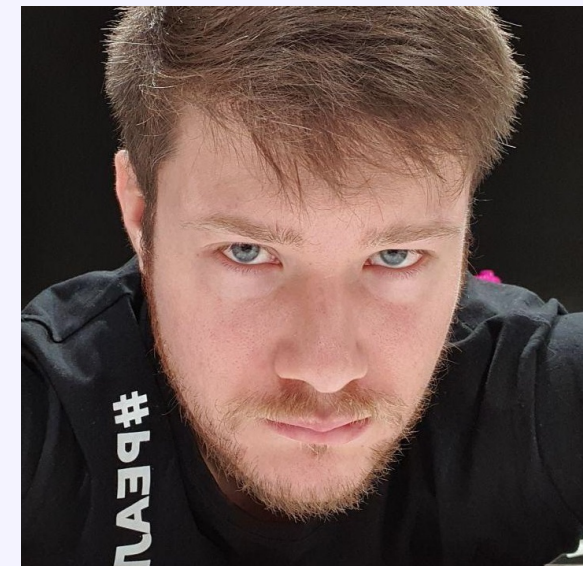
IT TEAM ONLINE





Николай Свиридов

- **Python Backend Developer**



КОНТАКТНЫЕ ДАННЫЕ

<https://www.linkedin.com/in/luchanos/>

t.me/luchanos

[Vk.com/luchanos](https://vk.com/luchanos)

PEP-8

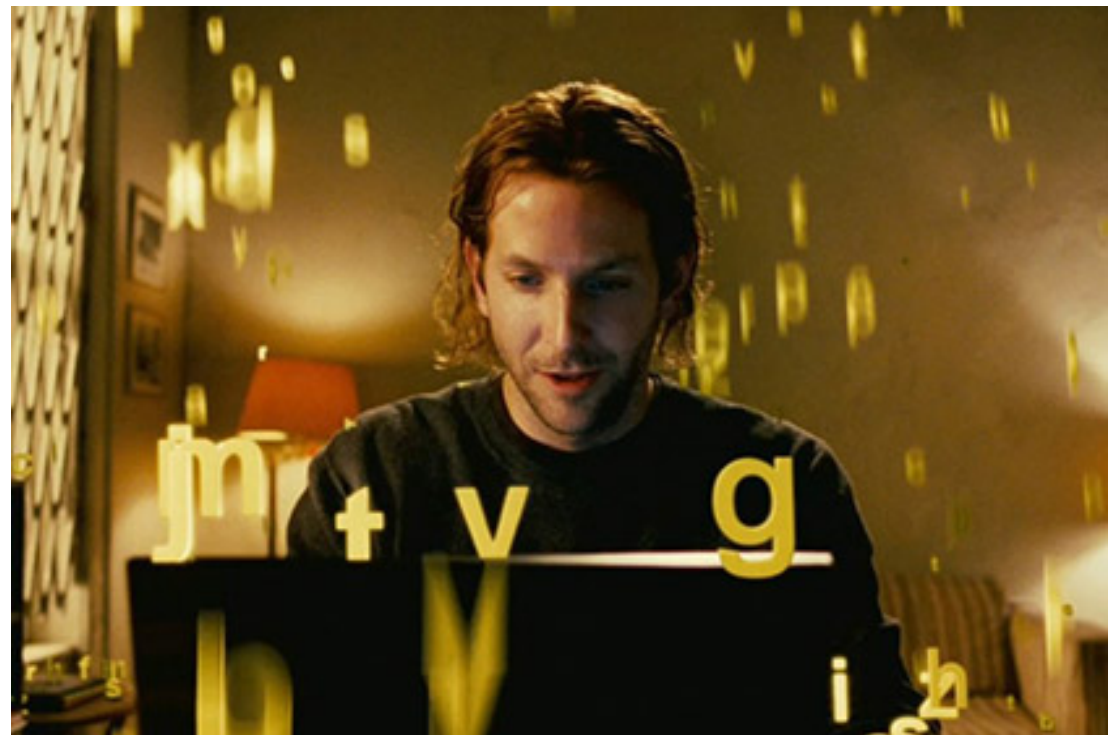
- этот документ описывает соглашения для написания кода на Python
- Библия, Пиратский кодекс чести Питонистов



Области видимости (LEGB)

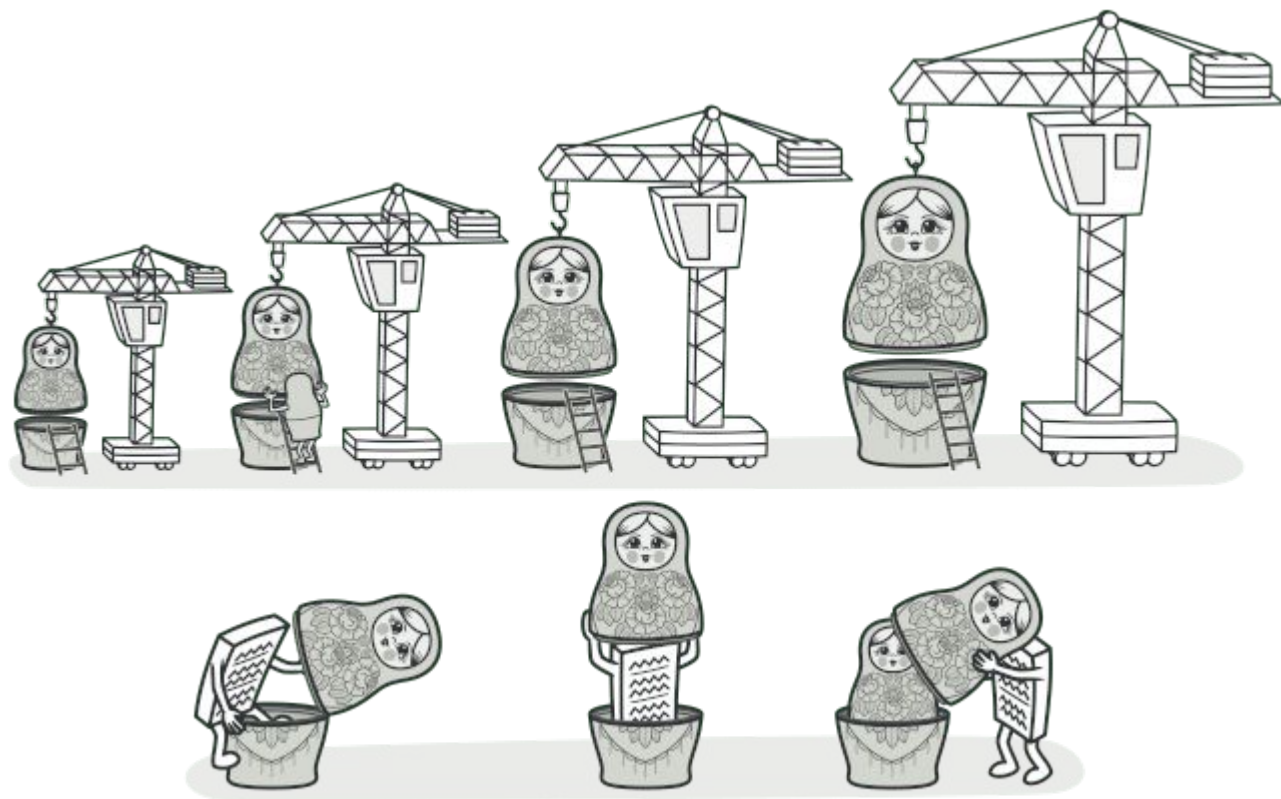
Определяют порядок видимости переменных в зависимости от того, откуда к ним обращаются.

- L - local - локальная
- E - enclosing - объемлющая
- G - global - глобальная
- B - built-in - встроенная



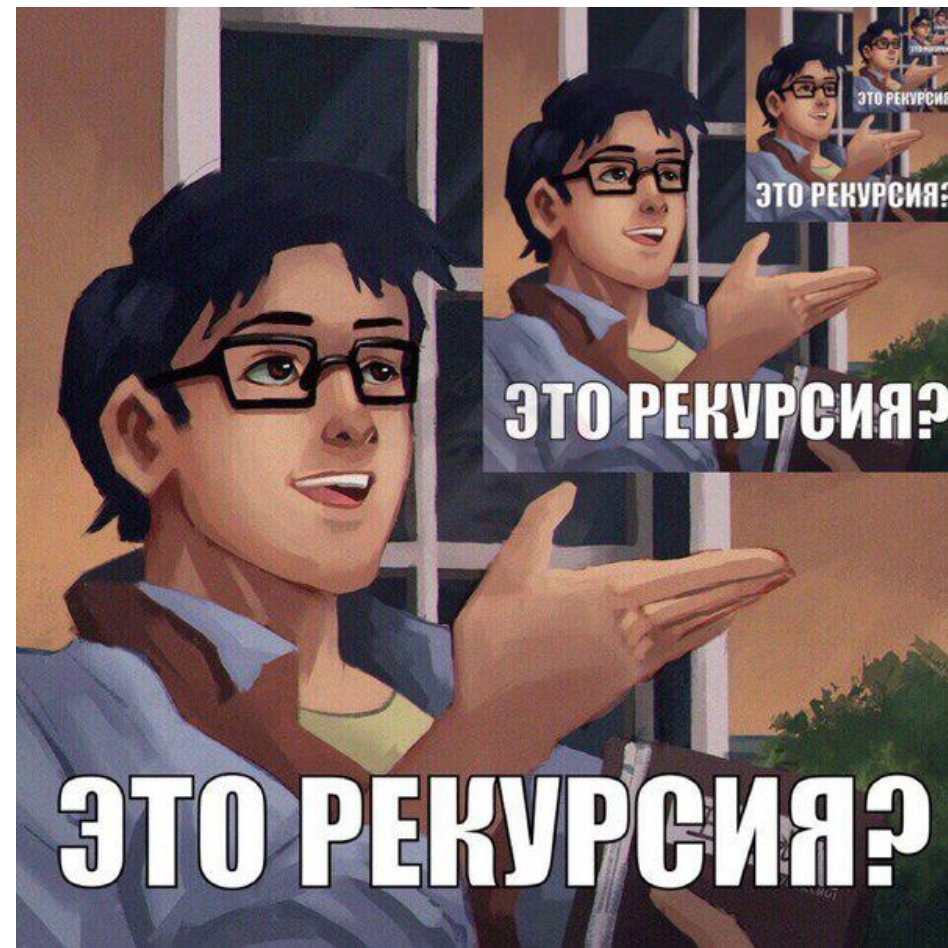
Декоратор

- паттерн проектирования, позволяющий изменить поведение функции без её непосредственного изменения
- в Python также имеется синтаксический сахар с подобным названием



Рекурсивные функции

- функция, которая осуществляет вызов самой себя
- любой цикл можно представить в виде рекурсии и наоборот



Класс

- Это объект, который описывает правило для создания другого объекта, а также свойства, которыми он будет обладать.
- По сути играет роль чертежа, по которому мы собираем экземпляр!



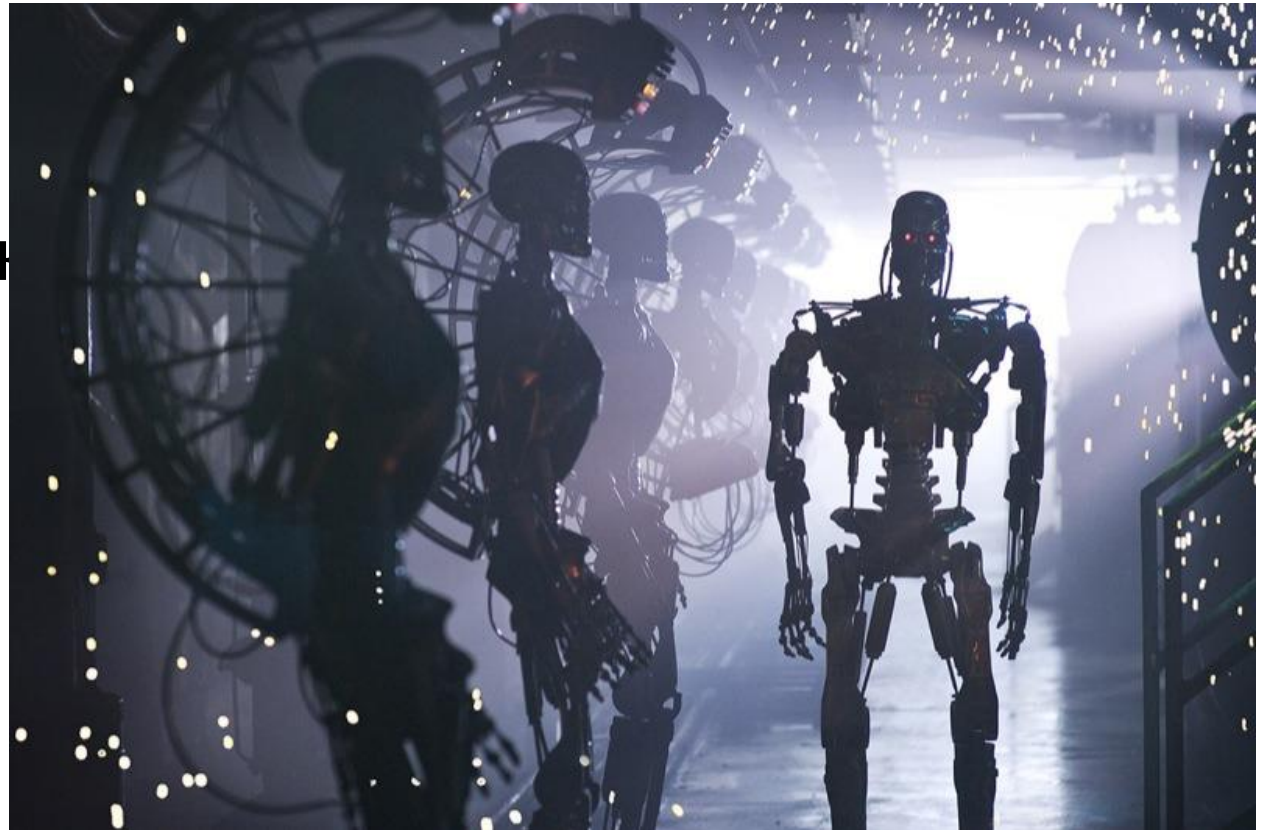
А у мистера Поттера есть его ключ?



Кое-что получше. У него есть рисунок ключика.

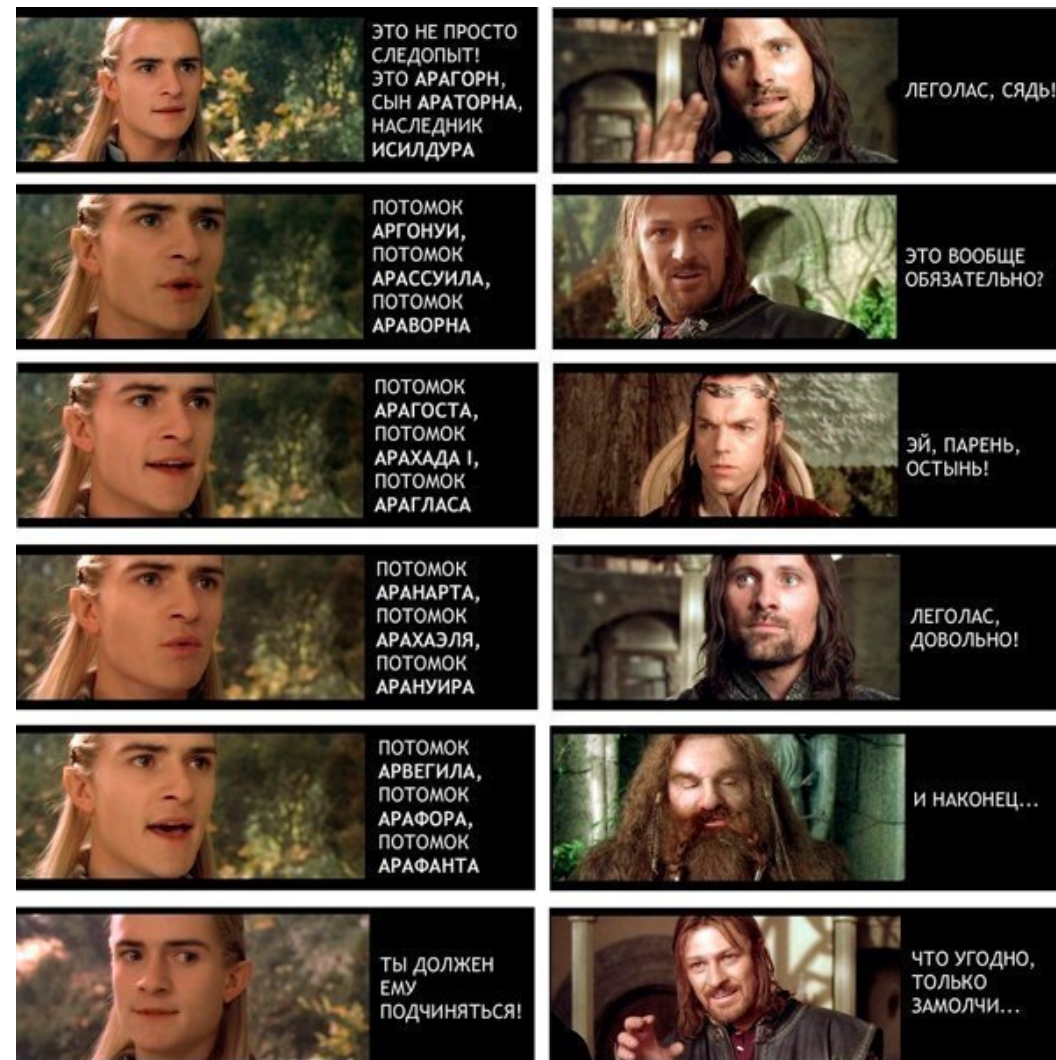
Экземпляр

- это объект, который был создан с помощью класса, как по чертежу
- в Python всё является объектами каких-либо классов
- самый базовый класс - object



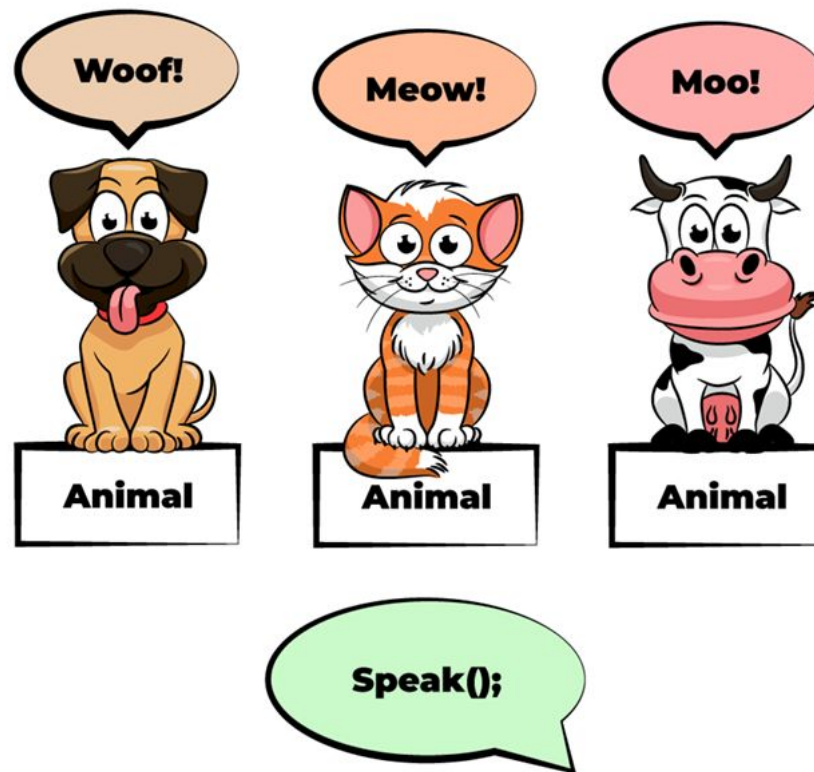
Наследование

- оно же “подражание”
- механизм в составе концепции ООП, способствующий повторному использованию компонентов ПО.



Полиморфизм

- использование единственной сущности для представления различных типов в различных сценариях использования



Инкапсуляция

- механизм связывания данных с методами, которые с этими данными работают, а также сокрытие внутренних деталей реализации от пользователя



Магические методы

- по сути являются интерфейсами для реализации некоторых полезных возможностей объектов (например арифметические операции), которые созданы на основе классов с этими методами



Домашнее задание

1. Реализуйте базовый класс Car.

У класса должны быть следующие атрибуты: speed, color, name, is_police (булево).

А также методы: go, stop, turn(direction), которые должны сообщать, что машина поехала, остановилась, повернула (куда);

опишите несколько дочерних классов: TownCar, SportCar, WorkCar, PoliceCar;

добавьте в базовый класс метод show_speed, который должен показывать текущую скорость автомобиля; для классов TownCar и WorkCar переопределите метод show_speed. При значении скорости свыше 60 (TownCar) и

40 (WorkCar) должно выводиться сообщение о превышении скорости.

Реализовать метод для user-friendly вывода информации об автомобиле.

Домашнее задание

2. Давайте представим, что мы занимаемся проектированием CRM для сервисного центра по обслуживанию и ремонту техники.

Реализуйте класс Заявка. Каждая заявка должна иметь следующие поля: уникальный идентификатор (присваивается в момент)

создания заявки автоматически, дата и время создания заявки (автоматически), имя пользователя, серийный номер оборудования, статус (активная заявка или закрытая например, статусов может быть больше). Id заявки сделать приватным полем.

У заявки должны быть следующие методы:

- метод, возвращающий, сколько заявка находится в активном статусе (если она в нём)
- метод, изменяющий статус заявки
- метод, возвращающий id заявки

3. Реализовать класс матрицы произвольного типа. При создании экземпляра передаётся вложенный список. Для объектов класса реализовать метод сложения и вычитания матриц, а также умножения, деления матрицы на число и user-friendly вывода матрицы на экран.

I T E A
O N L I N E

