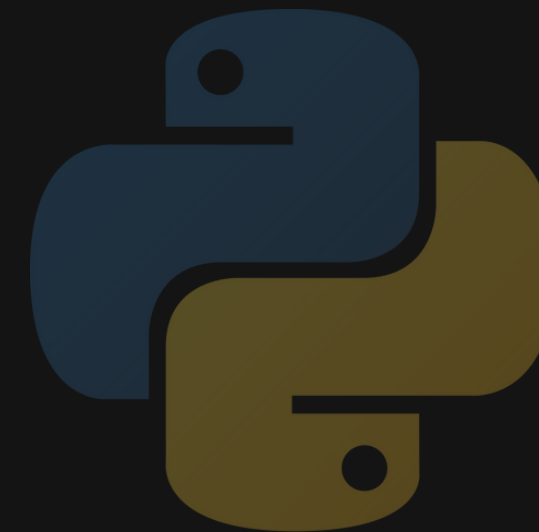


Добро пожаловать на 2й месяц!

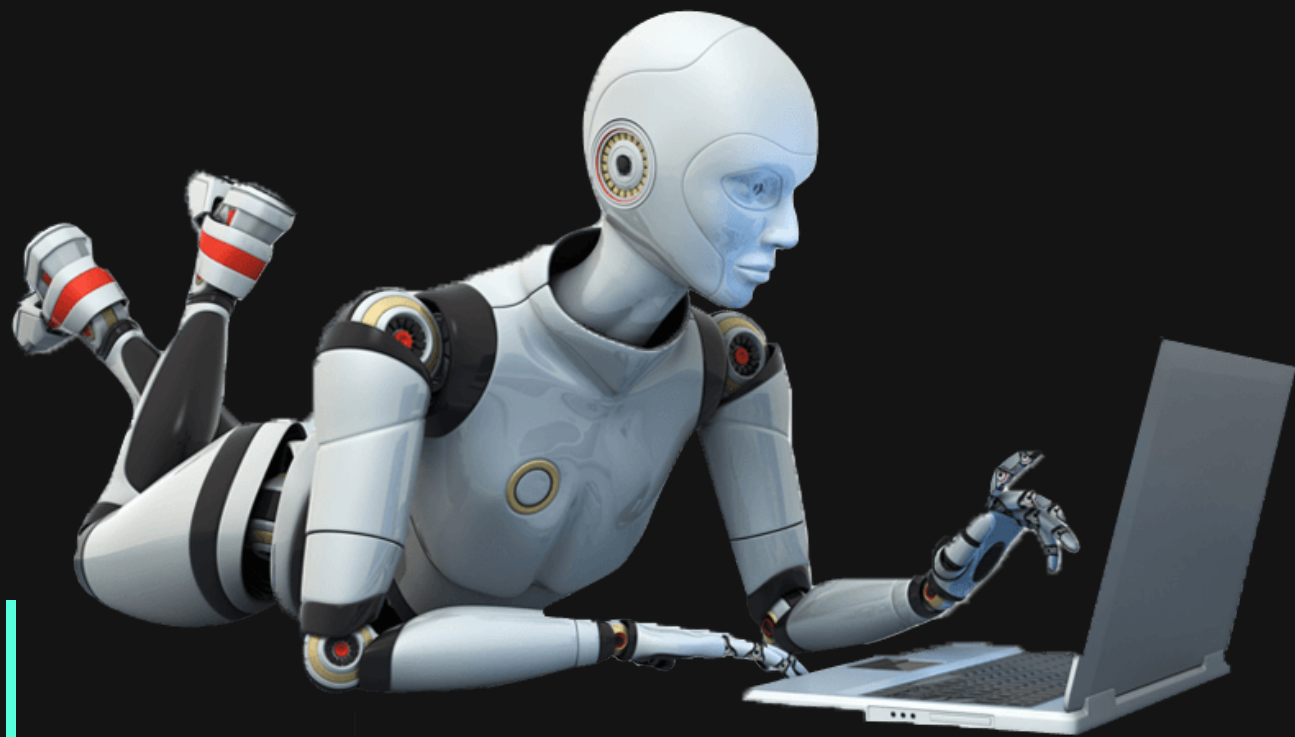


ЗНАКОМСТВО С ООП, ПРОДВИНУТЫЙ
PYTHON!



```
NAME = "ESENBEK"  
SECOND_NAME = "SAMATOVICH"
```

План месяца



*СМОТРИТ МЕМЧИКИ

УРОК 1

Суть ООП, Создание первых классов, Атрибуты и Методы классов, Наследование

УРОК 2

Основные принципы ООП, Инкапсуляция, Полиморфизм

УРОК 3

Множественное наследование, Магические методы в классах

УРОК 4

Практическое закрепление пройденного материала по ООП, Написание RPG игры в ООП стиле

УРОК 5

Модули в python, GitHub, Виртуальная среда

УРОК 6

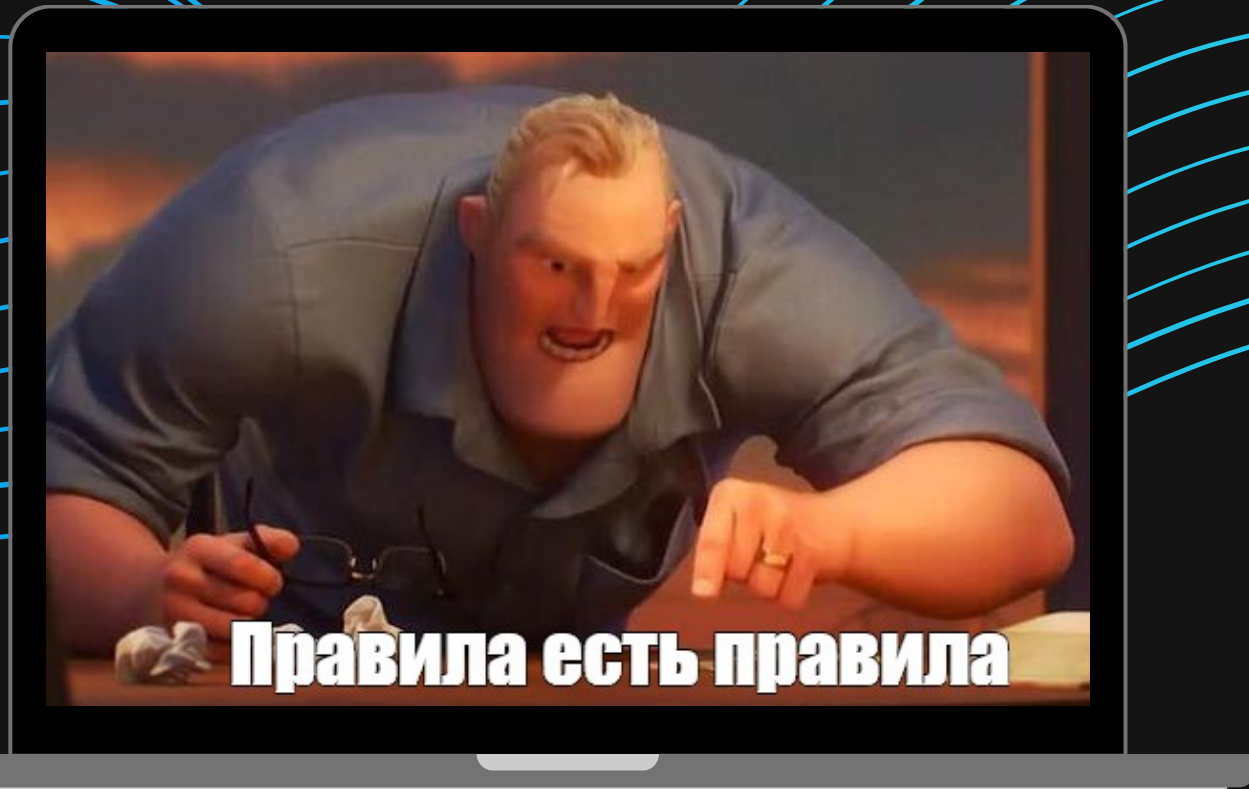
Регулярные выражения

УРОК 7

Алгоритмы и структуры данных, Big O нотация

УРОК 8

Базы данных и СУБД, Работа с БД в Python



Правила

DEADLINE

Дедлайн - до начала следующего урока.

ОЦЕНКА ДОМАШКИ

```
if deadline == 'passed':  
    return 5  
if deadline == 'in time':  
    return 10
```

ТЕСТЫ

1. Промежуточный (10 вопросов), на уроке №4
2. Финальный (20 вопросов), на уроке №8

ОТПРАВКА ДЗ

ДЗ отправлять в формате:

- 1) Имя
- 2) Группа
- 3) Номер дз
- 4) Файл или Ссылка на репозиторий

МАКСИМАЛЬНЫЙ ФИДБЭК

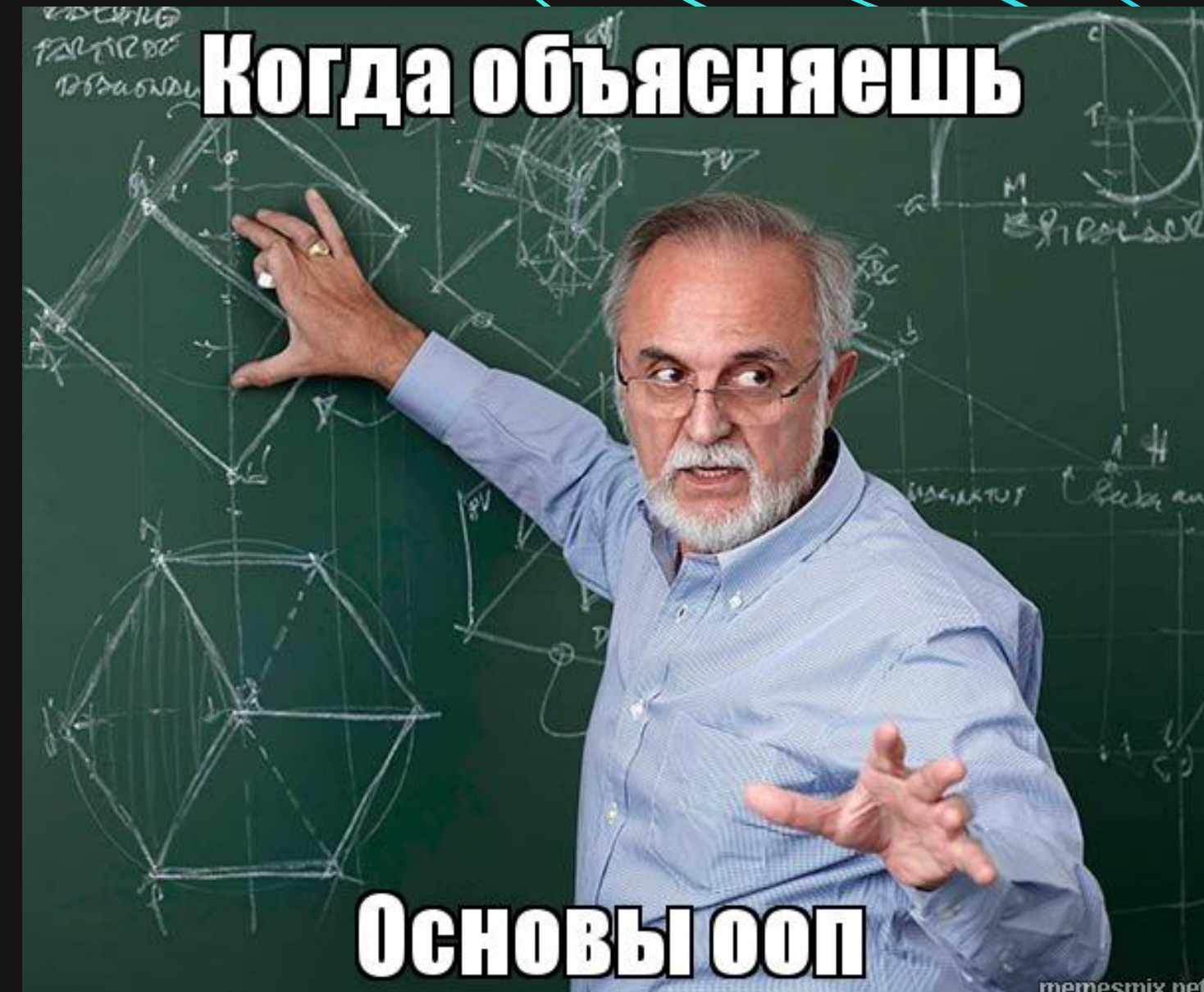
Чтобы понять - надо быть открытым и задавать любые вопросы

Урок №1

- A. Суть ООП
- B. Создание первых классов
- C. Атрибуты и Методы классов
- D. Наследование

ДЕТАЛИ:

- Что такое класс и что такое объект
- Пример класса Car
- Атрибуты
- Конструктор (`__init__(self)`)
- Значения по умолчанию в конструкторе
- Создание объектов (аргументы с названием атрибута и без)
- Адрес объекта - `self`
- Методы
- Общие атрибуты класса
- Основные принципы ООП - Наследование
- Родительский и дочерний класс
- Связка по конструкторам
- Переопределение атрибутов уровня класса
- Пример `Transport->Rocket`, `Car->Truck`



ООП

Объектно-ориентированное программирование (ООП) — это подход, при котором программа рассматривается как набор объектов, взаимодействующих друг с другом. У каждого есть свойства и поведение. ООП ускоряет написание кода и делает его более читаемым.

Идеология объектно-ориентированного программирования (ООП) разрабатывалась, чтобы связать поведение определенного объекта с его классом. Людям проще воспринимать окружающий мир как объекты, которые поддаются определенной классификации (например, разделение на живую и неживую природу).

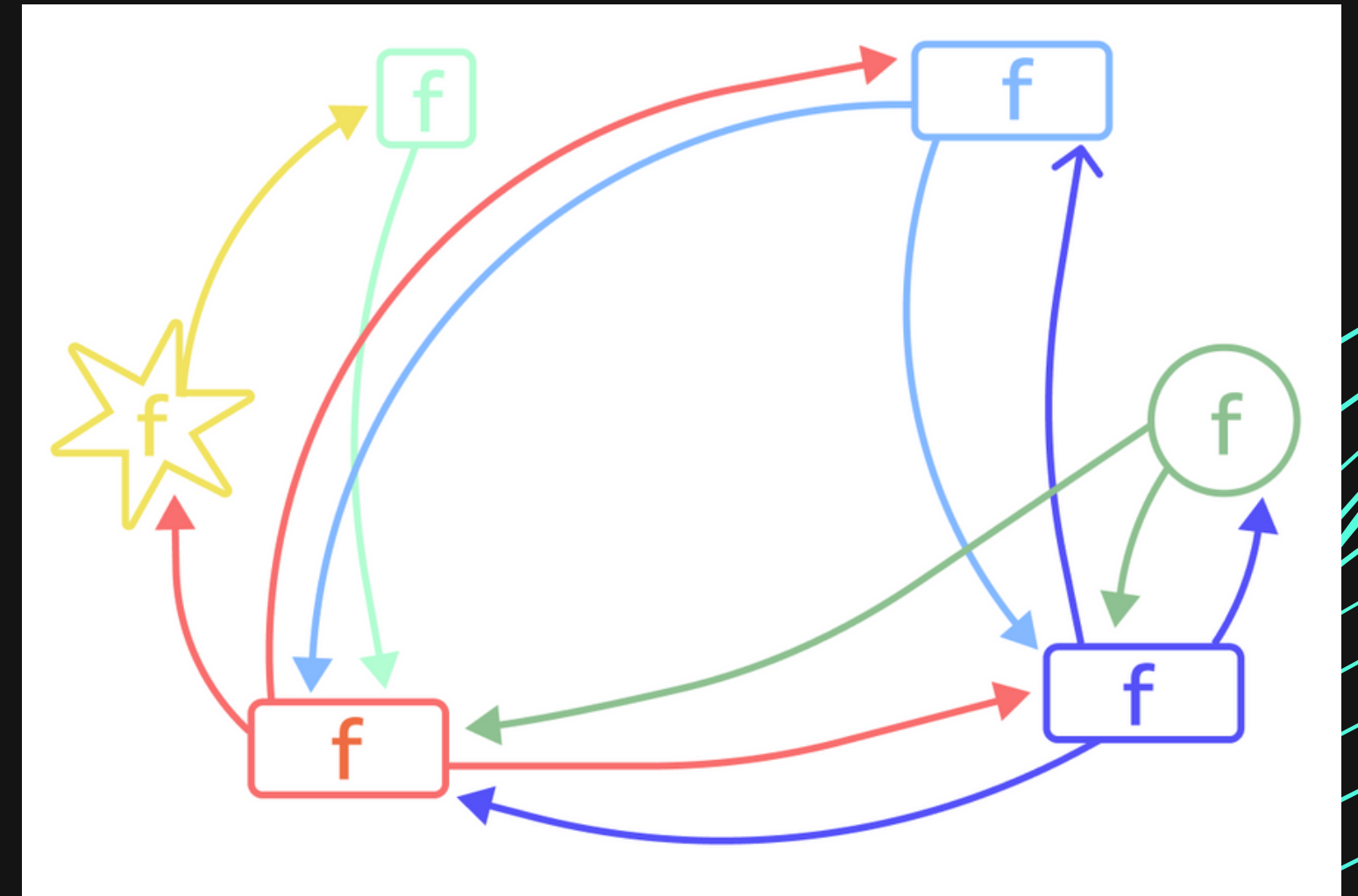


ИДЕМ ДАЛЬШЕ!

ТЕПЕРЬ ПОНЯТНО!

Зачем нужно ООП

До ООП в разработке использовался другой подход — процедурный. Программа представляется в нем как набор процедур и функций — подпрограмм, которые выполняют определенный блок кода с нужными входящими данными. Процедурное программирование хорошо подходит для легких программ без сложной структуры. Но если блоки кода большие, а функций сотни, придется редактировать каждую из них, продумывать новую логику. В результате может образоваться много плохо читаемого, перемешанного кода — «спагетти-кода» или «лапши».



В отличие от процедурного, объектно-ориентированное программирование позволяет вносить изменения один раз — в объект. Именно он — ключевой элемент программы. Все операции представляются как взаимодействие между объектами. При этом код более читаемый и понятный, программа проще масштабируется.



*МОМЕНТ ПРОЗРЕНИЯ

Объектно-ориентированное
программирование
используется, чтобы:



СТРУКТУРИРОВАТЬ ИНФОРМАЦИЮ И НЕ ДОПУСКАТЬ
ПУТАНИЦЫ;

ТОЧНО ОПРЕДЕЛЯТЬ ВЗАИМОДЕЙСТВИЕ ОДНИХ
ЭЛЕМЕНТОВ С ДРУГИМИ;

ПОВЫШАТЬ УПРАВЛЯЕМОСТЬ ПРОГРАММЫ;

БЫСТРЕЕ МАСШТАБИРОВАТЬ КОД ПОД РАЗЛИЧНЫЕ
ЗАДАЧИ;

ЭФФЕКТИВНЕЕ ПОДДЕРЖИВАТЬ ГОТОВЫЕ
ПРОГРАММЫ;

ЛУЧШЕ ПОНИМАТЬ НАПИСАННОЕ;

ВНЕДРЯТЬ ИЗМЕНЕНИЯ БЕЗ НЕОБХОДИМОСТИ
ПЕРЕПИСЫВАТЬ ВЕСЬ КОД.

Объекты и классы

Чтобы сделать код проще, программу разбивают на независимые блоки — объекты. В реальной жизни это может быть стол, чашка, человек, книга, здание и многое другое. В программировании объекты — это структуры данных: пользователь, кнопка, сообщение. У них, как и у реальных предметов, могут быть свойства: цвет, содержание или имя пользователя. А чтобы объединить между собой объекты с похожими свойствами, существуют классы.

Класс — это «шаблон» для объекта, который описывает его свойства. Несколько похожих между собой объектов, например профили разных пользователей, будут иметь одинаковую структуру, а значит, принадлежать к одному классу. Каждый объект — это экземпляр какого-нибудь класса.



Класс:
программист

Объект:
разработчик Иван

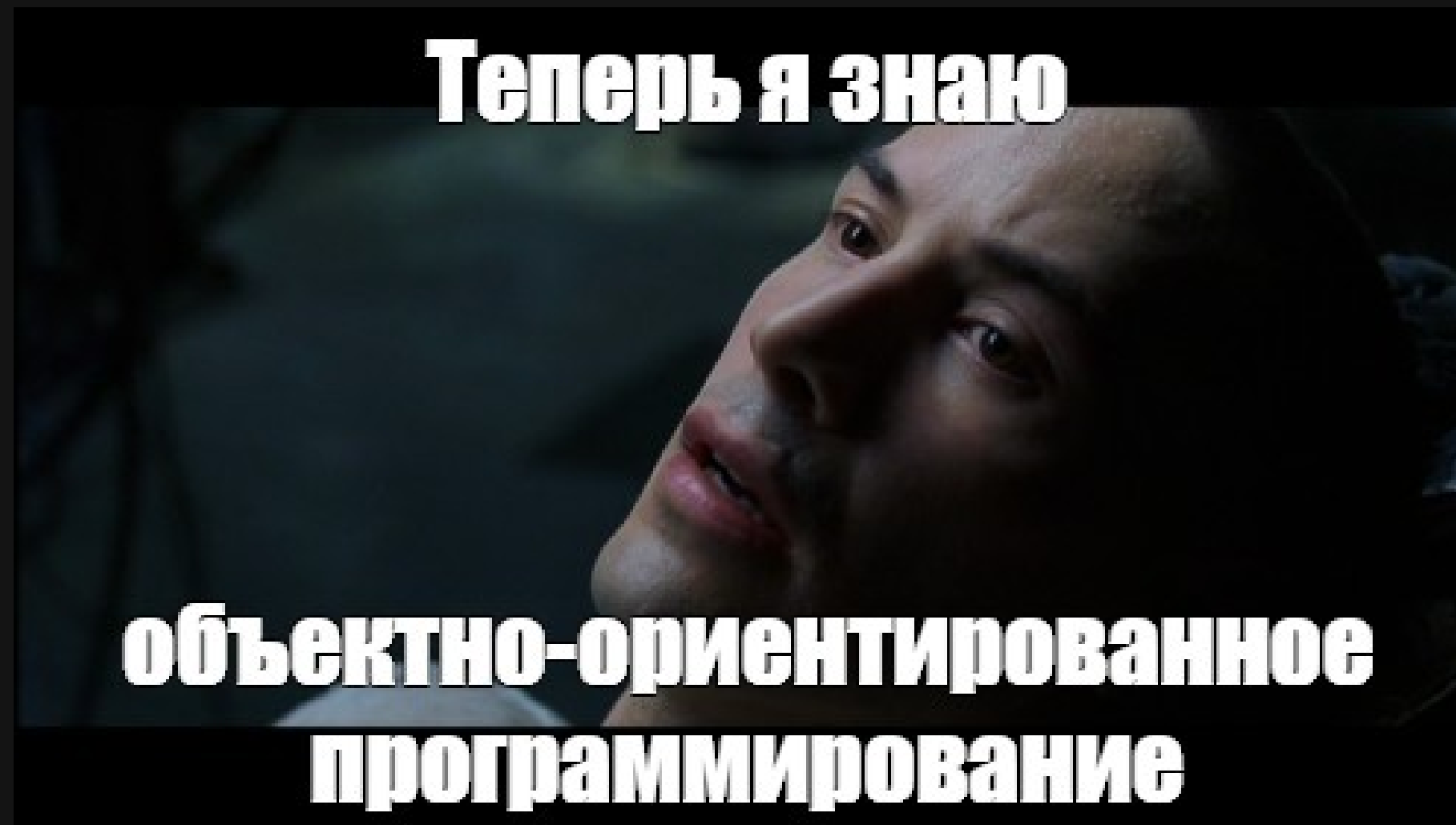
Атрибуты:
зарплата, обязанности

Методы:
написание кода

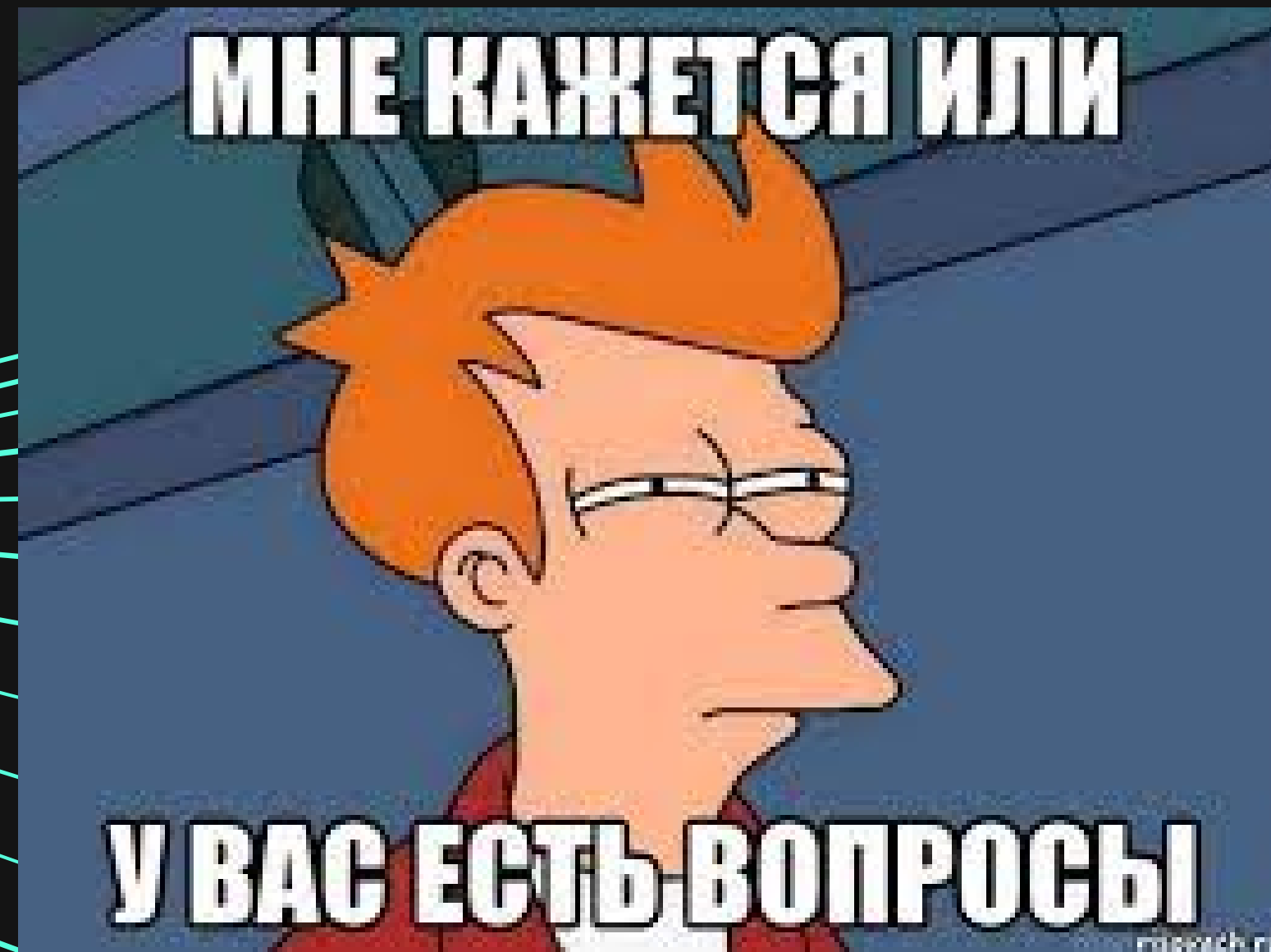
Атрибуты и методы

Объект — это набор переменных и функций, как в традиционном функциональном программировании. Переменные и функции и есть его свойства.

- Атрибуты — это переменные, конкретные характеристики объекта, такие как цвет поля или имя пользователя.
- Методы — это функции, которые описаны внутри объекта или класса. Они относятся к определенному объекту и позволяют взаимодействовать с ними или другими частями кода.



ВОПРОСЫ?



ДЗ №1

1. Создать класс Person с атрибутами fullname, age, is_married
2. Добавить в класс Person метод introduce_myself, который бы распечатывал всю информацию о человеке
3. Создать класс Student наследовать его от класса Person и дополнить его атрибутом marks, который был бы словарем, где ключ это название урока, а значение - оценка.
4. Добавить метод в класс Student, который бы подсчитывал среднюю оценку ученика по всем предметам
5. Создать класс Teacher и наследовать его от класса Person, дополнить атрибутом experience.
6. Добавить в класс Teacher атрибут уровня класса salary
7. Также добавить метод в класс Teacher, который бы считал зарплату по следующей формуле: к стандартной зарплате прибавляется бонус 5% за каждый год опыта свыше 3х лет.
8. Создать объект учителя и распечатать всю информацию о нем и высчитать зарплату
9. Написать функцию create_students, в которой создается 3 объекта ученика, эти ученики добавляются в список и список возвращается функцией как результат.
10. Вызвать функцию create_students и через цикл распечатать всю информацию о каждом ученике с его оценками по каждому предмету. Также рассчитать его среднюю оценку по всем предметам.