

# Объектно ориентированное программирование



**ITISMA**

**Краткий очерк: что, как и зачем**

# Что вы узнаете в этом видео



что такое ООП



как появилось ООП



как базово работает ООП

# Парадигмы программирования: это что?

— это стиль или способ построения программного обеспечения. Она определяет, как программист думает о структуре и поведении программ.

```
def main():  
    ...  
    <some_logic>  
  
def first_function():  
    ...  
    <some_other_logic>
```

Самая известная для новичков парадигма программирования — функциональная. В ней основными строительными блоками программ являются функции.

# Что такое Объектно-ориентированное-программирование



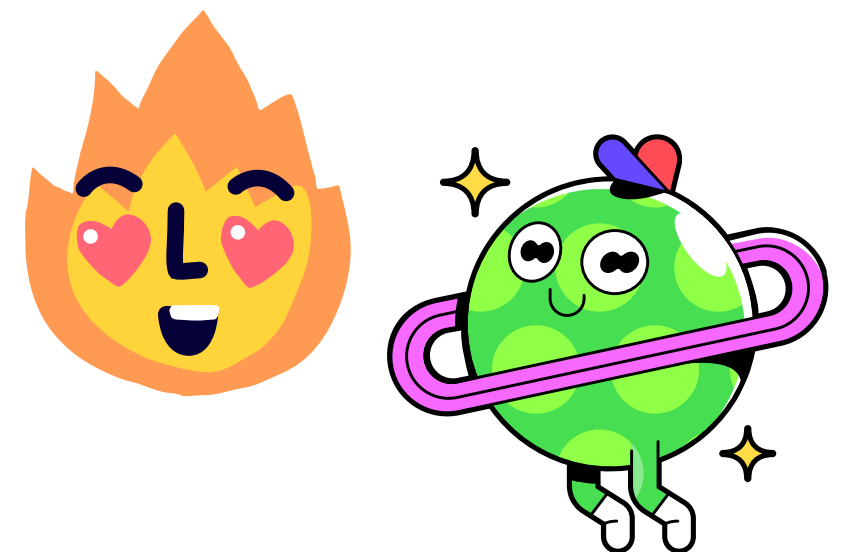
— это парадигма программирования, основанная на концепции 'объектов', которые могут содержать данные в виде полей, часто называемых атрибутами; и код в виде процедур, часто называемых методами. Особенностью объектов является то, что процедуры объекта могут получать доступ к полям данных объекта и часто модифицировать их. В ООП компьютерные программы проектируются с использованием объектов, которые взаимодействуют друг с другом.\*



# Что такое Объектно-ориентированное-программирование

## простыми словами

— это метод написания программ, в котором программы создаются из объектов. Объекты объединяют данные и действия над ними в одну сущность. Это похоже на создание моделей реальных вещей, где каждый объект может иметь свои характеристики и умеет что-то делать. ООП помогает организовать код так, чтобы его было легче понимать и изменять.





**Сталкивались ли вы с ООП? Когда и где?**



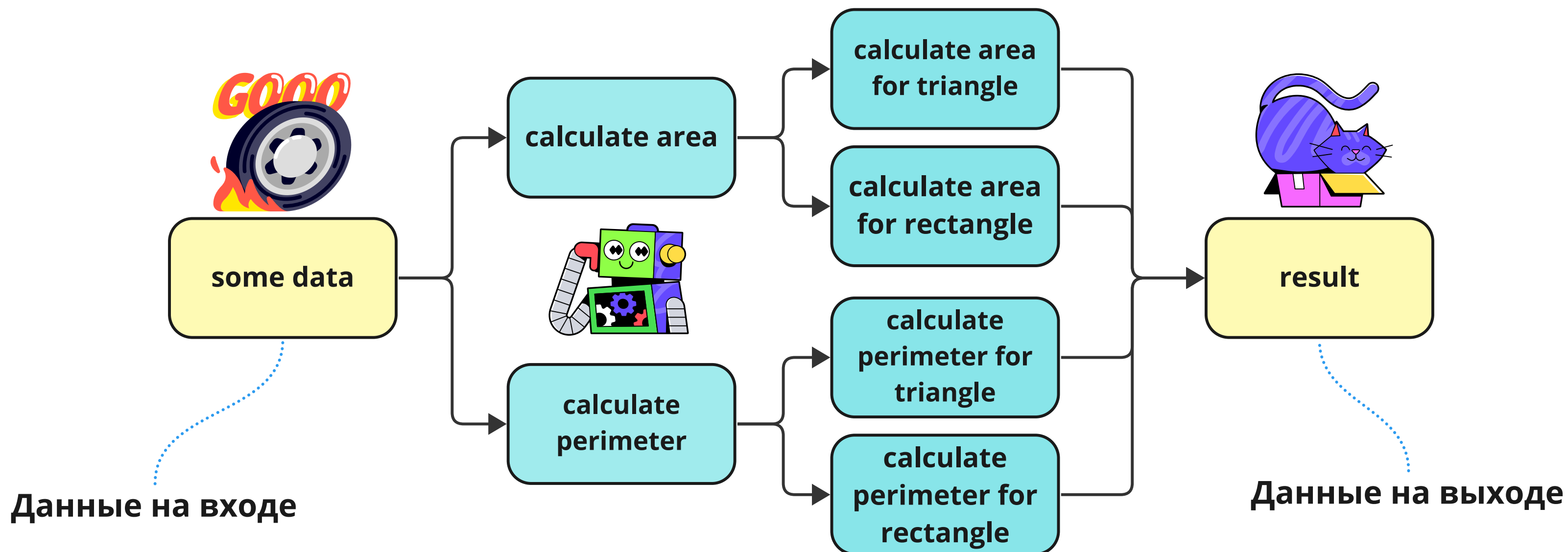
# Как развивалось Объектно-ориентированное программирование



# Как работает объектно-ориентированное программирование

## сравниваем с функциональной парадигмой

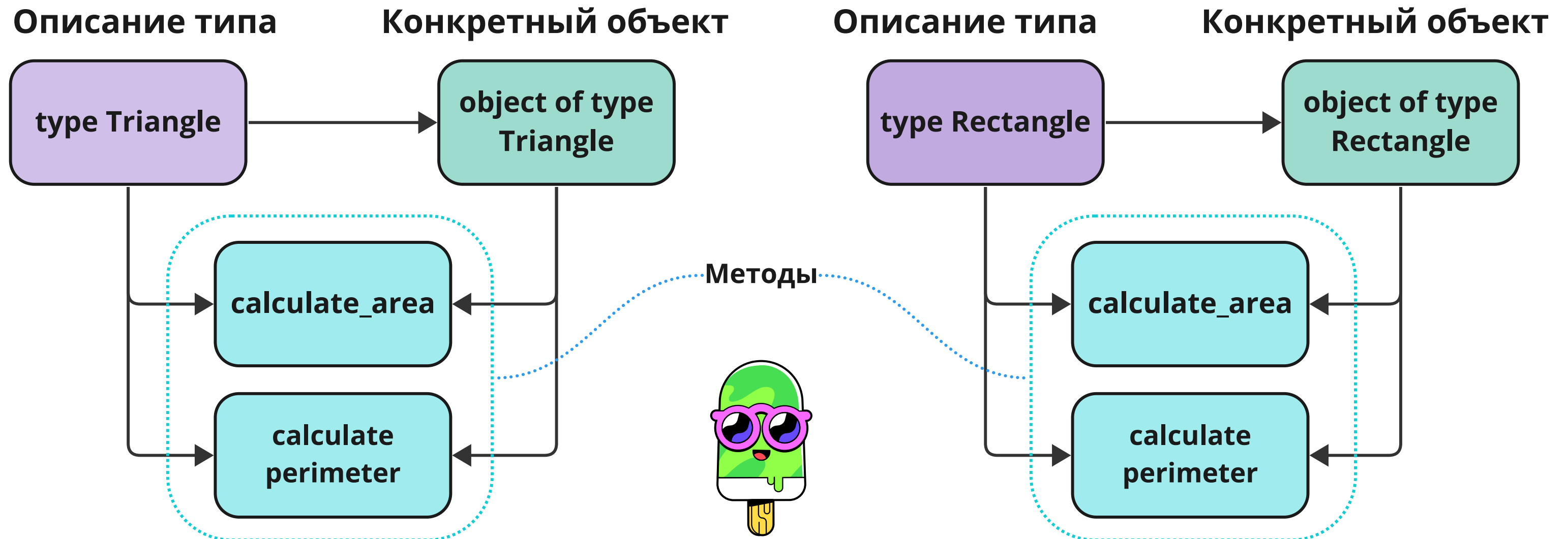
Функциональное программирование - мы описываем функции





# Что такое Объектно-ориентированное-программирование простыми словами

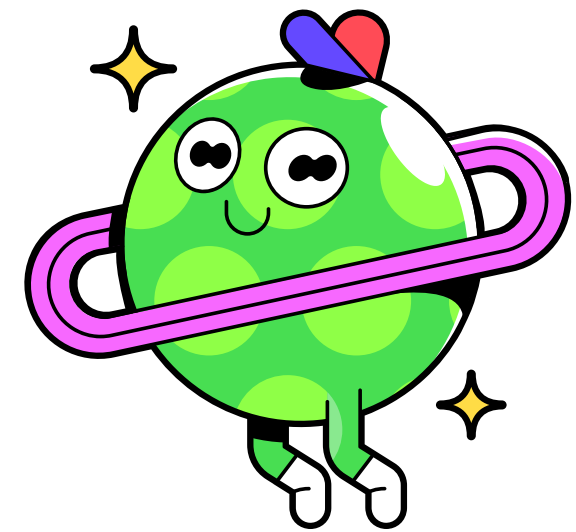
Объектно-ориентированное программирование - мы описываем типы и их методы



# ООП хорошо в меру и в умелых руках

## Достоинства ООП

- ☞ Модульность - бьем все на классы
- ☞ Повторное использование кода
- ☞ Расширяемость
- ☞ Абстракция



## Недостатки ООП

- ☞ Сложность и избыточность для простых задач
- ☞ Потребление ресурсов
- ☞ Крутая кривая обучения
- ☞ Проблемы с масштабируемостью



# В нашем Телеграм-канале много интересного

## @itism\_movement



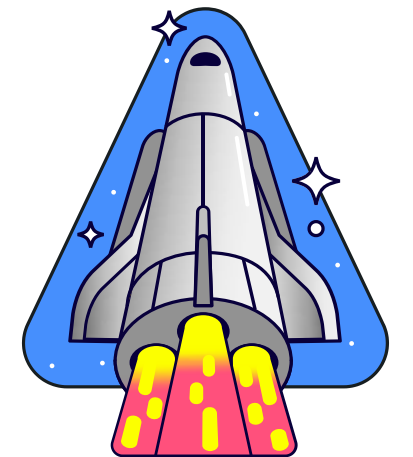
t.me







### ITism [Community Edition]

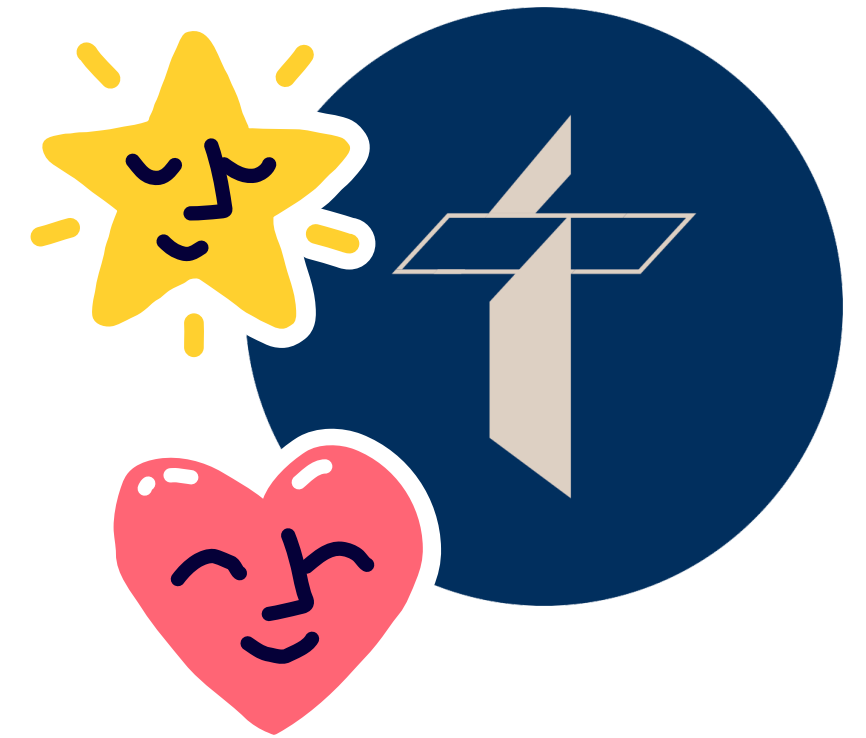
Инженерная культура доступная каждому

# А теперь немного кода



# Выводы

-  ООП предоставляет нам мощные инструменты для создания программного обеспечения
-  ООП улучшает модульность кода и способствует его повторному использованию
-  Нет одной универсальной парадигмы, которая подходит для всех задач. Везде есть свои сильные и слабые стороны
-  Умение выбирать правильные инструменты для конкретной задачи — ключ к успеху



Подпишись на **ITIS**

**Инженерную Культуру доступную каждому**

