

The background of the slide is a light gray gradient. It is decorated with numerous realistic water droplets of various sizes. Some droplets are large and prominent, while others are small and scattered. They are primarily located in the top-left, bottom-right, and bottom-center areas, with a few smaller ones near the top center.

QA WITH PYTHON

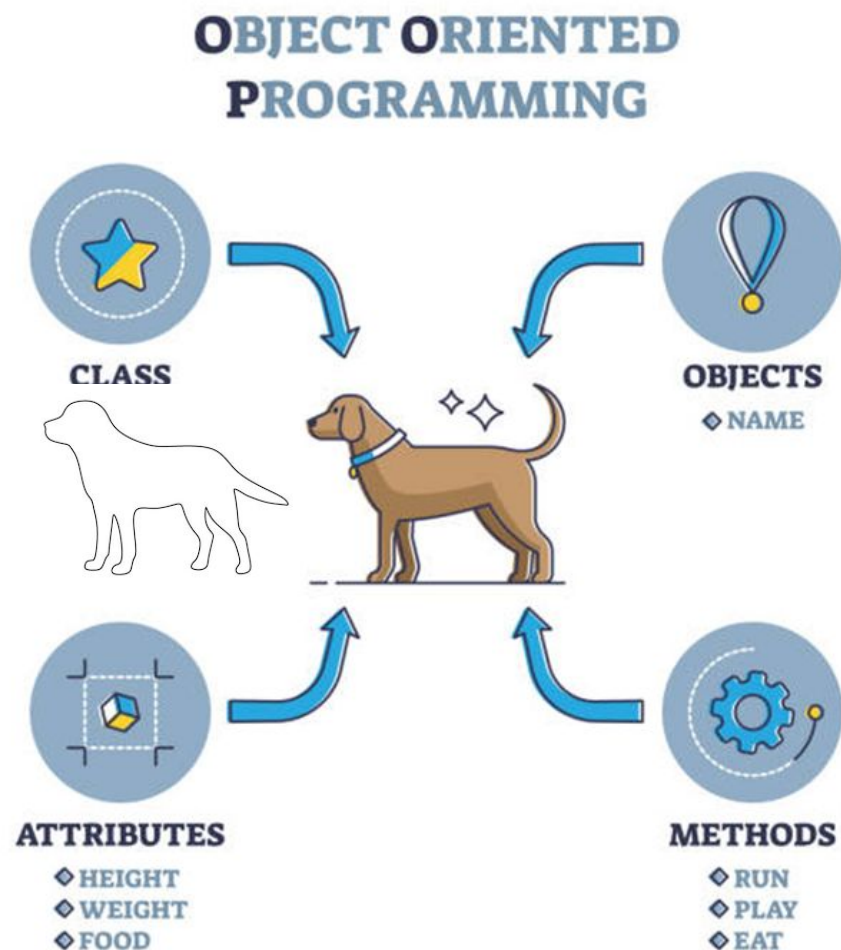
OOP.

GIT/GITHUB

ЧТО ТАКОЕ ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ (ООП)?

- ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ (ООП) — ЭТО ПОДХОД, ПРИ КОТОРОМ ПРОГРАММА РАССМАТРИВАЕТСЯ КАК НАБОР ОБЪЕКТОВ, ВЗАИМОДЕЙСТВУЮЩИХ ДРУГ С ДРУГОМ.
- 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.

АТТРИБУТЫ И МЕТОДЫ



КЛАСС - ЭТО ШАБЛОН КОДА, ПО КОТОРОМУ СОЗДАЮТСЯ ОБЪЕКТЫ, ИМЕЮЩИЕ ОБЩУЮ СТРУКТУРУ И ОБЛАДАЮЩИЕ ОДИНАКОВЫМ ПОВЕДЕНИЕМ. САМ ПО СЕБЕ КЛАСС НИЧЕГО НЕ ДЕЛАЕТ, НО С ЕГО ПОМОЩЬЮ МОЖНО СОЗДАТЬ ОБЪЕКТ И ИСПОЛЬЗОВАТЬ ЕГО В РАБОТЕ.

ОБЪЕКТ - ЭТО ЭКЗЕМПЛЯР КЛАССА

У КАЖДОГО ОБЪЕКТА ЕСТЬ СВОЙСТВА (**АТТРИБУТЫ**) И ПОВЕДЕНИЕ (**МЕТОДЫ**)

КОНСТРУКТОР

- В ОБЪЕКТНО-ОРИЕНТИРОВАННОМ ПРОГРАММИРОВАНИИ КОНСТРУКТОРОМ КЛАССА НАЗЫВАЮТ МЕТОД, КОТОРЫЙ АВТОМАТИЧЕСКИ ВЫЗЫВАЕТСЯ ПРИ СОЗДАНИИ ОБЪЕКТОВ. ЕГО ТАКЖЕ МОЖНО НАЗВАТЬ КОНСТРУКТОРОМ ОБЪЕКТОВ КЛАССА.
- В PYTHON РОЛЬ КОНСТРУКТОРА ИГРАЕТ МЕТОД `__init__()`
- ПЕРВЫЙ ЕГО ПАРАМЕТР – `SELF` – ССЫЛКА НА САМ ТОЛЬКО ЧТО СОЗДАННЫЙ ОБЪЕКТ.

```
CLASS PERSON:
```

```
    DEF __INIT__(SELF, NAME, SURNAME):
```

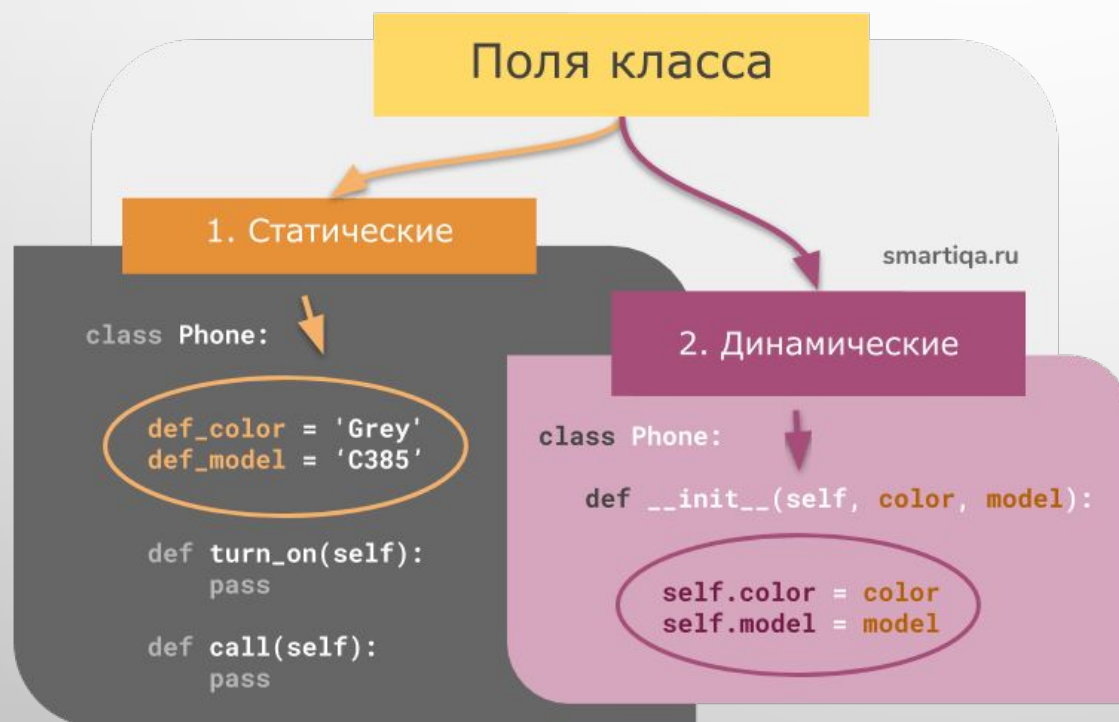
```
        SELF.NAME = NAME
```

```
        SELF.SURNAME = SURNAME
```

```
PERSON_1 = PERSON("ALEX", "BAKER") (ЗДЕСЬ ПРИ ВЫЗОВЕ КЛАССА В КРУГЛЫХ СКОБКАХ  
ПЕРЕДАЮТСЯ ЗНАЧЕНИЯ, КОТОРЫЕ БУДУТ ПРИСВОЕНЫ ПАРАМЕТРАМ МЕТОДА __init__() )
```

- НЕОБХОДИМОСТЬ КОНСТРУКТОРОВ СВЯЗАНА С ТЕМ, ЧТО НЕ РЕДКО ОБЪЕКТЫ

ПОЛЯ/АТТРИБУТЫ КЛАССА

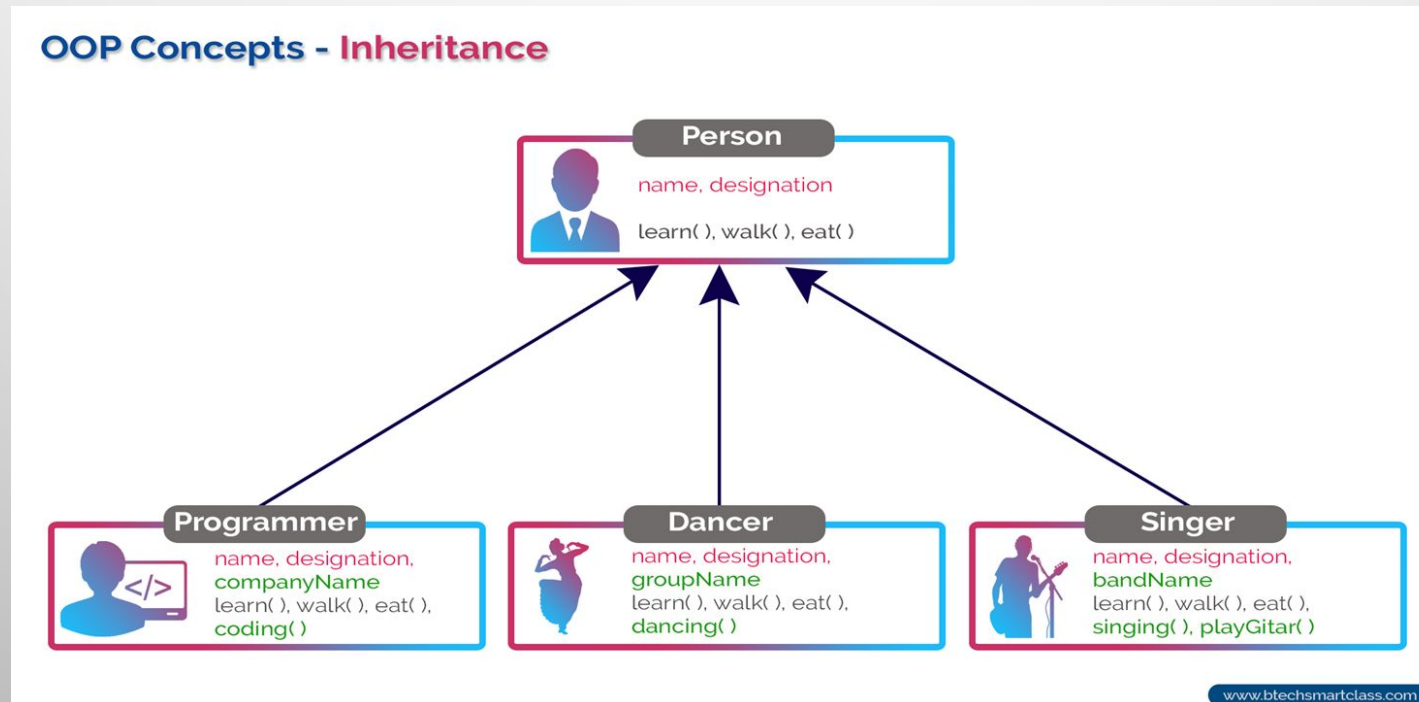


СТАТИЧЕСКИЕ ПОЛЯ (ОНИ ЖЕ ПЕРЕМЕННЫЕ ИЛИ АТТРИБУТЫ КЛАССА) - ЭТО ПЕРЕМЕННЫЕ, КОТОРЫЕ ОБЪЯВЛЯЮТСЯ ВНУТРИ ТЕЛА КЛАССА И СОЗДАЮТСЯ ТОГДА, КОГДА СОЗДАЕТСЯ КЛАСС

ДИНАМИЧЕСКИЕ ПОЛЯ (ПЕРЕМЕННЫЕ ИЛИ АТТРИБУТЫ ЭКЗЕМПЛЯРА КЛАССА) – ЭТО ПЕРЕМЕННЫЕ, КОТОРЫЕ СОЗДАЮТСЯ НА УРОВНЕ ЭКЗЕМПЛЯРА КЛАССА.

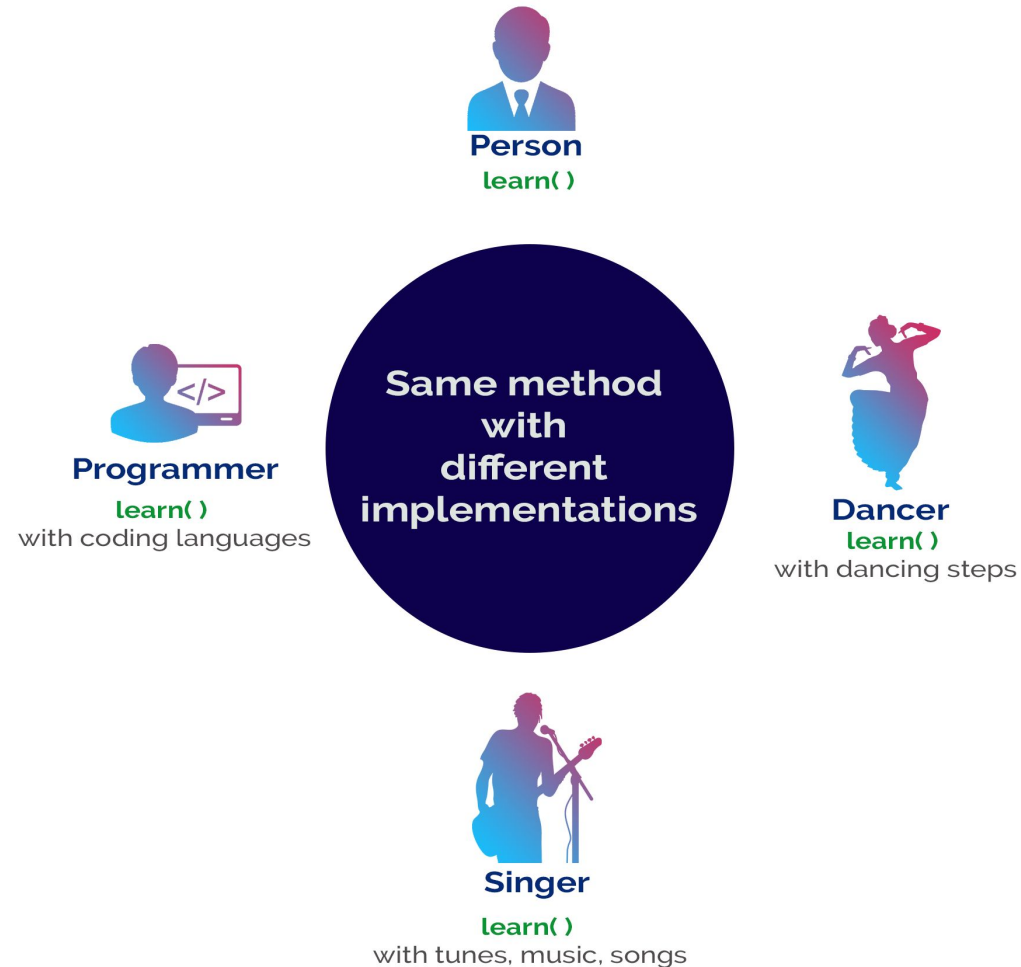
НАСЛЕДОВАНИЕ

- **НАСЛЕДОВАНИЕ** ПОЗВОЛЯЕТ ВЫДЕЛИТЬ ОБЩЕЕ ДЛЯ НЕСКОЛЬКИХ КЛАССОВ ПОВЕДЕНИЕ И ВЫНЕСТИ ЕГО В ОТДЕЛЬНУЮ СУЩНОСТЬ. ТО ЕСТЬ НАСЛЕДОВАНИЕ ЯВЛЯЕТСЯ СРЕДСТВОМ ПЕРЕИСПОЛЬЗОВАНИЯ КОДА (CODE REUSE) — ИСПОЛЬЗОВАНИЯ СУЩЕСТВУЮЩЕГО КОДА ДЛЯ РЕШЕНИЯ НОВЫХ ЗАДАЧ



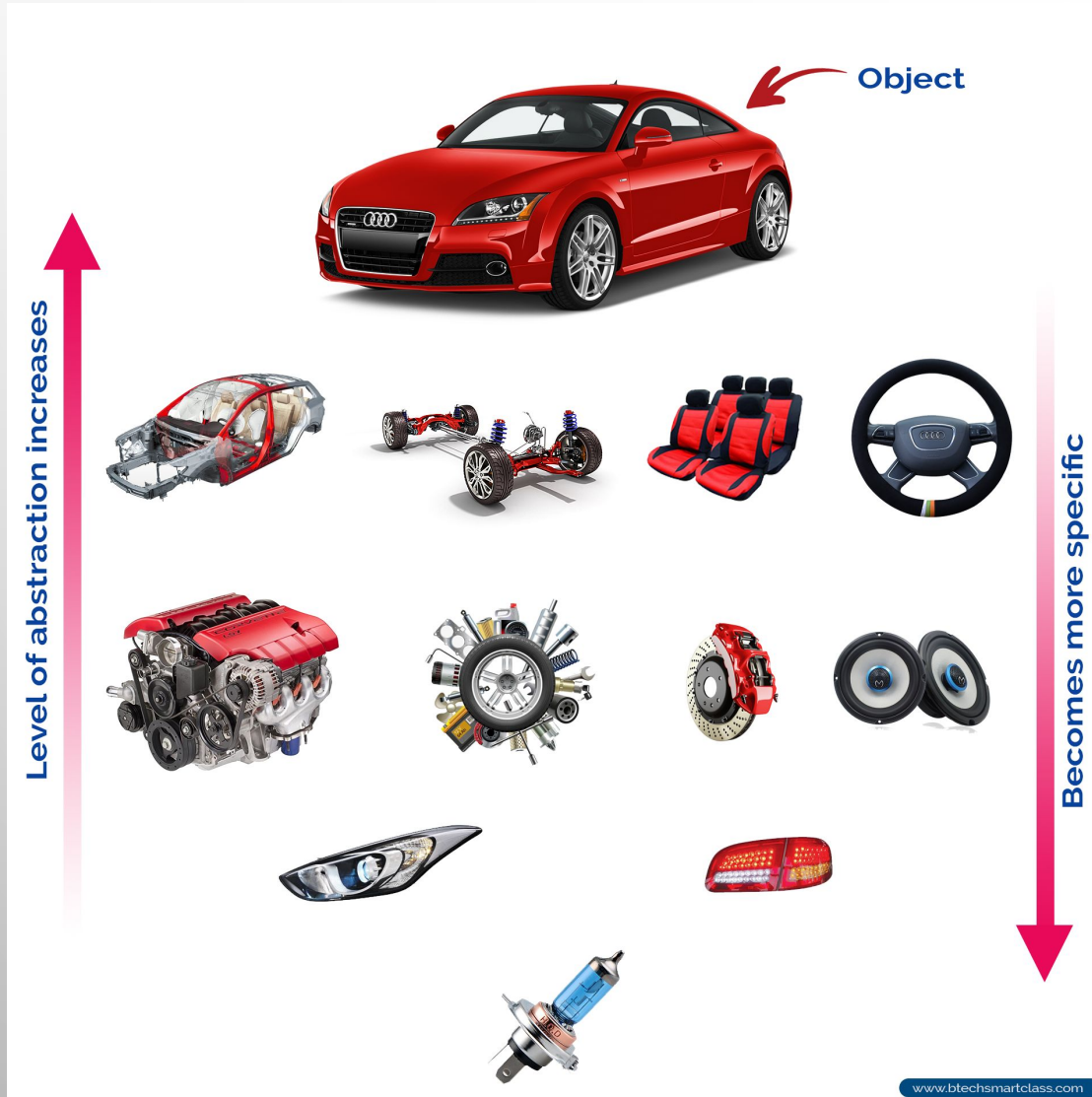
ПОЛИМОРФИЗМ

OOP Concepts - Polymorphism



ПОЛИМОРФИЗМ -
РАЗНОЕ ПОВЕДЕНИЕ
ОДНОГО И ТОГО ЖЕ
МЕТОДА В РАЗНЫХ
КЛАССАХ.

АБСТРАКЦИЯ



АБСТРАКЦИЯ — ЭТО
ВЫДЕЛЕНИЕ ОСНОВНЫХ,
НАИБОЛЕЕ ЗНАЧИМЫХ
ХАРАКТЕРИСТИК ОБЪЕКТА
И ИГНОРИРОВАНИЕ
ВТОРОСТЕПЕННЫХ.

ИНКАПСУЛЯЦИЯ

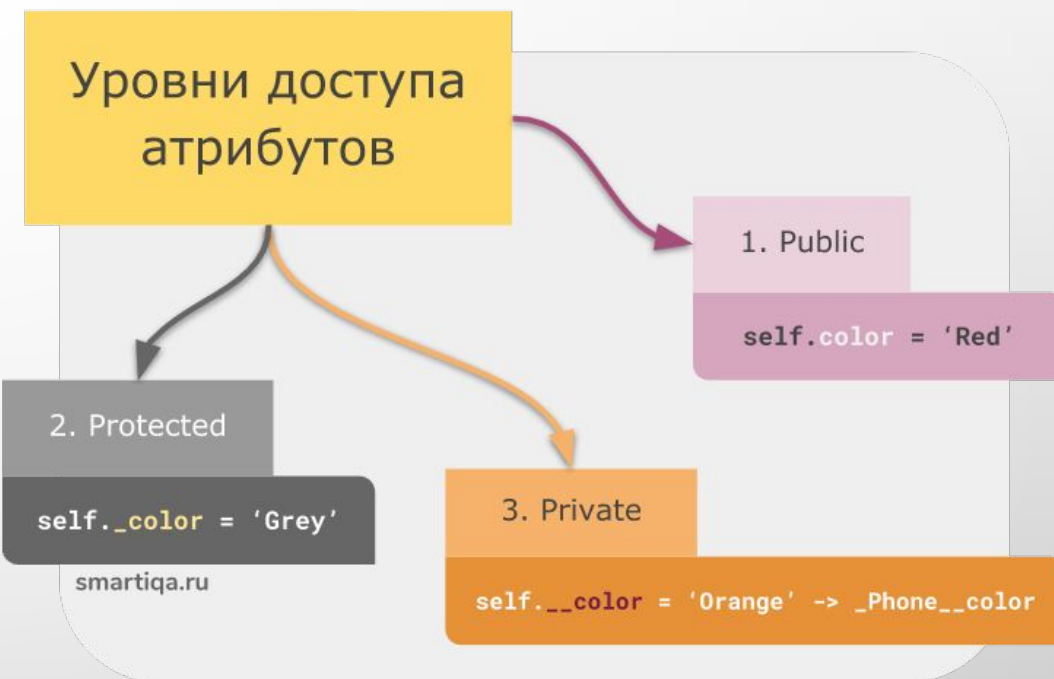
Encapsulation = Data + Code



Class = Variables + Methods

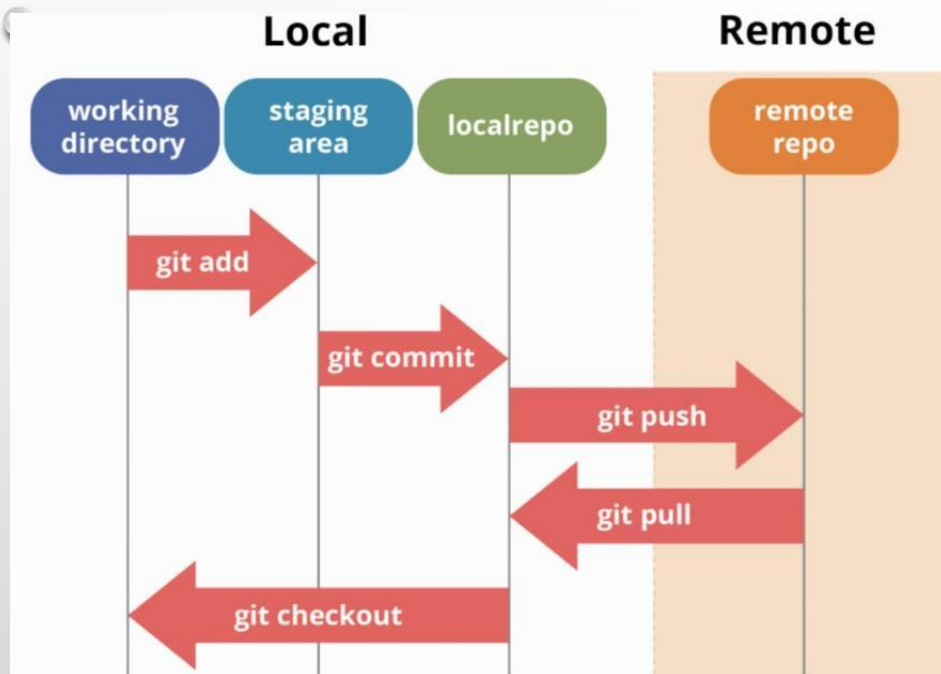
www.btechsmartclass.com

ГРУППИРОВКА ЛОГИЧЕСКИ
СВЯЗАННЫХ ДАННЫХ И МЕТОДОВ



МЕХАНИЗМ КОНТРОЛЯ ДОСТУПА К
ДАНЫМ

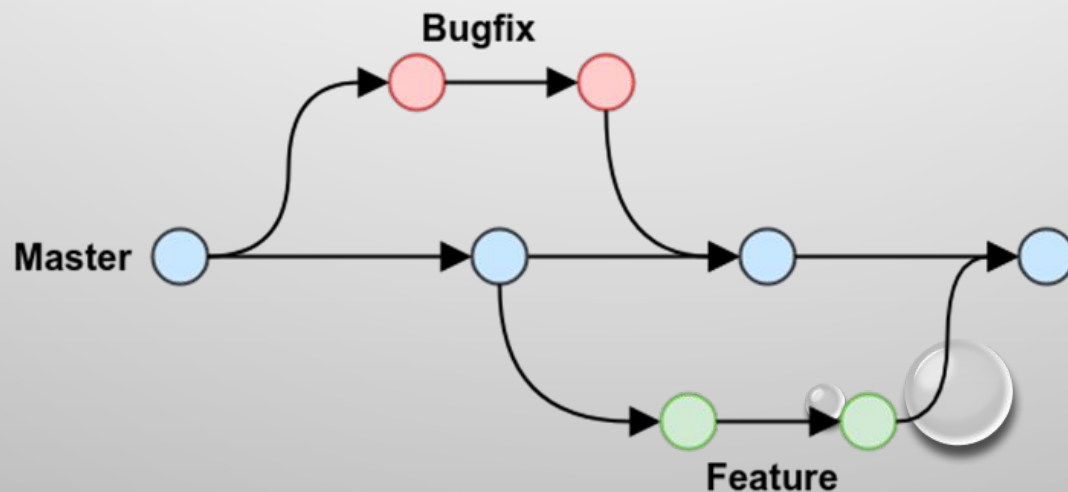
СИСТЕМА КОНТРОЛЯ ВЕРСИЙ



GIT – ЭТО КОНСОЛЬНАЯ УТИЛИТА, ДЛЯ ОТСЛЕЖИВАНИЯ И ВЕДЕНИЯ ИСТОРИИ ИЗМЕНЕНИЯ ФАЙЛОВ. ТАКЖЕ **GIT** НАЗЫВАЮТ СИСТЕМОЙ КОНТРОЛЯ ВЕРСИЙ (VCS)

ЗАДАЧИ, КОТОРЫЕ РЕШАЕТ VCS:

- ЗАЩИЩАЕТ ИСХОДНЫЙ КОД ОТ ПОТЕРИ.
- ОБЕСПЕЧИВАЕТ КОМАНДНУЮ РАБОТУ.
- ПОМОГАЕТ ОТМЕНИТЬ ИЗМЕНЕНИЯ.
- РАСПРЕДЕЛЁННАЯ РАБОТА.



ОСНОВНЫЕ ПОНЯТИЯ GIT

- GIT — СИСТЕМА КОНТРОЛЯ ВЕРСИЙ (VERSION CONTROL SYSTEM). АКТИВНО ИСПОЛЬЗУЕТСЯ В ИТ-ПРОЕКТАХ ДЛЯ ОТСЛЕЖИВАНИЯ И ВЕДЕНИЯ ИСТОРИИ ИЗМЕНЕНИЯ ФАЙЛОВ.
- MASTER — ГЛАВНАЯ ИЛИ ОСНОВНАЯ ВЕТКА РЕПОЗИТОРИЯ.
- ВЕТКА (BRANCH) — ЭТО ПАРАЛЛЕЛЬНАЯ ВЕРСИЯ РЕПОЗИТОРИЯ. ОНА ВКЛЮЧЕНА В ЭТОТ РЕПОЗИТОРИЙ, НО НЕ ВЛИЯЕТ НА ГЛАВНУЮ ВЕРСИЮ, ТЕМ САМЫМ ПОЗВОЛЯЯ СВОБОДНО РАБОТАТЬ В ПАРАЛЛЕЛЬНОЙ.
- СОММИТ — ЭТО ОПЕРАЦИЯ, КОТОРАЯ ОТПРАВЛЯЕТ ВСЕ ПОДГОТОВЛЕННЫЕ ИЗМЕНЕНИЯ (ОНИ МОГУТ ВКЛЮЧАТЬ ЛЮБОЕ КОЛИЧЕСТВО ФАЙЛОВ) КАК ЕДИНОЕ ЦЕЛОЕ В РЕПОЗИТОРИЙ

`GIT COMMIT -m <YOUR COMMIT COMMENT>`

- HEAD – ЭТО УКАЗАТЕЛЬ, КОТОРЫЙ ВСЕГДА УКАЗЫВАЕТ НА ПОСЛЕДНИЙ КОММИТ В ВЕТКЕ. ВСЯКИЙ РАЗ, КОГДА ВЫ ДЕЛАЕТЕ КОММИТ, HEAD ОБНОВЛЯЕТСЯ ПОСЛЕДНИМ КОММИТОМ.

ПОЛЕЗНЫЕ ССЫЛКИ

- [HTTPS://WWW.W3SCHOOLS.COM/PYTHON/PYTHON_CLASSES.ASP](https://www.w3schools.com/python/python_classes.asp) - КЛАССЫ И ОБЪЕКТЫ
- [HTTPS://WWW.W3SCHOOLS.COM/PYTHON/PYTHON_INHERITANCE.ASP](https://www.w3schools.com/python/python_inheritance.asp) - НАСЛЕДОВАНИЕ
- [HTTPS://GIT-SCM.COM/DOC](https://git-scm.com/doc) - ДОКУМЕНТАЦИЯ GIT
- [HTTPS://GIT-SCM.COM/BOOK/RU/V2/%D0%9F%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5-C%3A-%D0%9A%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D1%8B-GIT-%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%BD%D1%8B%D0%B5-%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D1%8B](https://git-scm.com/book/ru/v2/%D0%9F%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5-C%3A-%D0%9A%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D1%8B-GIT-%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%BD%D1%8B%D0%B5-%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D1%8B) – ОСНОВНЫЕ КОМАНДЫ GIT