Введение в python

ООП

Парадигма ООП

- Класс – тип, описывающий устройство объектов
- Объект – экземпляр класса

У класса есть методы (аналог фукнций) и переменные

Примеры:

Класс – машина
Объект – рено логан на котором за вами приехал таксист

Класс – кошка
Объект – та кошка которая живет у вас дома

Старая версия основных принципов ооп

- ***Полиморфизм*** – возможность объектов с одинаковой спецификацией иметь различную реализацию
- ***Наследование*** – механизм позволяющий описать новый класс на основе уже существующего (родительского), при этом свойства и функциональность родительского класса заимствуются новым классом.
- ***Инкапсуляция*** – свойство языка программирования, позволяющее пользователю не задумываться о сложности реализации используемого программного компонента (что у него внутри?), а взаимодействовать с ним посредством предоставляемого интерфейса (публичных методов и членов)

SOLID принципы

S Принцип единственной ответственности (single responsibility principle)
Для каждого класса должно быть определено единственное назначение.

O Принцип открытости/закрытости (open–closed principle)
Программные сущности должны быть открыты для расширения, но закрыты для модификации»

L Принцип подстановки Лисков (Liskov substitution principle)
Объекты в программе должны быть заменяемыми на экземпляры их подтипов без изменения правильности выполнения программы

I Принцип разделения интерфейса (interface segregation principle)
Много интерфейсов, специально предназначенных для клиентов, лучше, чем один интерфейс общего назначения»

D Принцип инверсии зависимостей (dependency inversion principle)
Зависимость на Абстракциях. Нет зависимости на что-то конкретное

Статические и классовые методы

- Экземплярные методы – позволяют менять состояние конкретного экземпляра класса и класса в целом
- Классовые методы – могут менять состояния объектов класса в целом, что влияет на все экземпляры класса, но не могут менять состояние конкретного экземпляра
- Статические – по сути функция, которая из соображений удобства помещена в класс. Не позволяет менять экземпляр класса или класс целиком

Уровни доступа к переменным и методам

Существует 3 уровня доступа

- Public – методы и переменные с таким уровнем доступа доступны всем
- Protected – методы и переменные доступны из класса и из наследников этого класса
- Private – методы и переменные доступны только из данного класса

В питоне механизм фактически не реализован из коробки