**Лабораторная работа N9 ООП**

1. Опишите механизм перегрузки оператора в классе?
2. Какие принципы лежат в основе перегрузки операторов в классах?
3. Какие методы можно перегружать?
4. Что собой представляет конструктор \_\_init\_\_()?
5. Зачем нужен деструктор?
6. Что такое полиморфизм?
7. Как реализуется полиморфизм в Python?

**1.Механизм перегрузки оператора в классе позволяет переопределить поведение операторов для объектов класса.**

**2.Принципы перегрузки операторов включают создание методов с определенными именами, принимающих self и другой объект, и возвращающих результат.**

**3.Можно перегружать методы, такие как \_\_add\_\_, \_\_sub\_\_, \_\_mul\_\_, \_\_truediv\_\_, а также методы сравнения, такие как \_\_lt\_\_, \_\_le\_\_, \_\_eq\_\_, \_\_ne\_\_, \_\_gt\_\_, \_\_ge.**

**4.Конструктор \_\_init\_\_ - это метод, вызываемый при создании объекта класса, инициализирующий его атрибуты.**

**5.Деструктор в Python не так важен из-за сборки мусора, но может использоваться для освобождения ресурсов при удалении объекта.**

**6.Полиморфизм позволяет объектам разных классов иметь методы с одинаковыми именами, но с разным поведением.**

**7.В Python полиморфизм реализуется через наследование и перегрузку методов, позволяя объектам разных классов вести себя по-разному, используя одинаково названные методы.**