**Лабораторная работа N10 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 наследование?
7. Как реализуется наследование в Python?
8. Что собой представляет операция расширение класса?
9. Что собой представляет операция делегирование?
10. Механизм перегрузки оператора в классе позволяет изменить стандартное поведение операторов для объектов данного класса.
11. Принципы перегрузки операторов включают определение методов с определенными именами в классе, которые определяют, как операторы должны взаимодействовать с объектами этого класса. Примеры включают **\_\_add\_\_** для оператора сложения и **\_\_str\_\_** для оператора преобразования в строку.
12. Можно перегружать методы, связанные с операторами, такие как **\_\_add\_\_**, **\_\_sub\_\_**, **\_\_eq\_\_**, **\_\_str\_\_**, и другие.
13. Конструктор **\_\_init\_\_()** - это специальный метод, который инициализирует новый объект класса. Он вызывается при создании нового объекта и позволяет устанавливать начальные значения атрибутов.
14. Деструктор не является обязательным в Python, но может быть использован для освобождения ресурсов или выполнения других действий при уничтожении объекта. Он определяется с помощью метода **\_\_del\_\_()**.
15. Наследование - это механизм, который позволяет классу наследовать атрибуты и методы другого класса. Это позволяет создавать новый класс, используя уже существующий класс как основу.
16. В Python наследование реализуется путем указания базового класса в определении нового класса. Новый класс получает атрибуты и методы базового класса и может добавить собственные атрибуты и методы.
17. Операция расширения класса (extending a class) представляет собой создание нового класса, который наследует от существующего класса и добавляет новые атрибуты и методы.
18. Операция делегирования (delegation) в контексте классов означает передачу выполнения определенных методов из одного класса в другой класс, чтобы избежать дублирования кода и обеспечить повторное использование.