Тест:

1. Что такое функция и как ее объявить?

Ответ: Функция – это составная инструкция которая может принимать данные ввода. Выполнять указания и возвращать данные вывода. Для объявления функции используем ключевое слово def пример (“def имя\_функции(параметры)”).

1. Как создать функцию?

Ответ: def fun()

1. Как показать что созданная функция возвращает результат?

Ответ: Для возвращения результата используем return.

1. В чем разница между параметрами и аргументами функции?

Ответ: Параметры – это переменные которые используются при создании функции.

Аргументы – это данные передаваемые функции при ее вызове.

1. Как рекурсивные функции работают в Python?

Ответ: Рекурсивная функция – это когда функция вызывает сама себя.

1. Что такое декораторы функций и как они используются? ( привести свой пример декоратора)

Ответ: Декоратор – это функция которая позволяет расширить (обернуть) другую функцию другим функционалом.

def decorator(func):  
 def wrapper():  
 print("Where is the ditonator?")  
 func()  
 print("I'm your father")  
 return wrapper  
@decorator  
def func1():  
 print('Дядя Игорь, ты чего?')  
func1()  
@decorator  
def func2():  
 print('Ты опять псилоцибиновые грибы употрибил?!')  
func2()

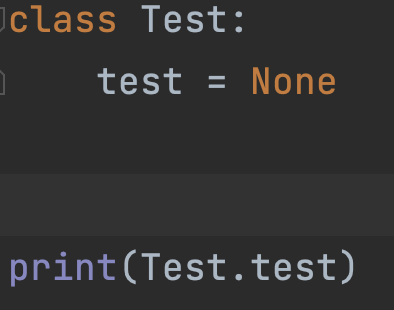
1. Как расшифровывается ООП?

Ответ: ООП – Объектно-ориентированное программирование

1. Что относится к основным принципам(парадигмам) ООП?

Ответ: Полиморфизм, Наследование, Абстракция, Инкапсуляция

1. Что будет результатом программы и почему?



Ответ: Выводом будет None

1. Какой принцип ООП описывает следующее предложение? Этот принцип является способностью использовать общий интерфейс для нескольких форм (типов данных).

Ответ: Речь идет про Полиморфизм

1. Какой из перечисленных вариантов является верным объявлением private поля?

* private field = 0
* field = 0
* \_field = 0
* \_\_field = 0

Ответ: \_\_field = 0

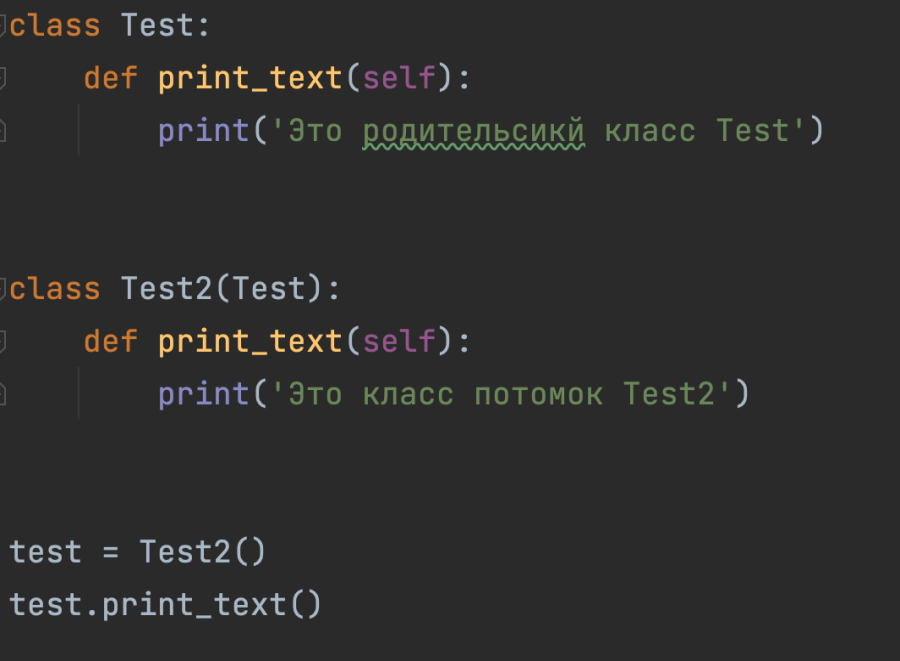
1. Как создать конструктор класса?

Ответ: при помощи метода \_\_init\_\_

1. Как много конструкторов в классе может иметь Python?

Ответ: Неограниченное количество

1. Что будет результатом данной программы



Ответ: Выводом будет содержимое class Test2 (Это класс потомок Test2)

1. Какой параметр обязательно принимает в себя метод экземпляра?

* тип объекта
* сам экземпляр класса
* название класса
* таких аргументов нет

Ответ: Экземпляр класса

1. Как использовать инкапсуляцию для защиты данных в классах?

Ответ: частично приватным методом \_protected, или приватный метод \_\_private

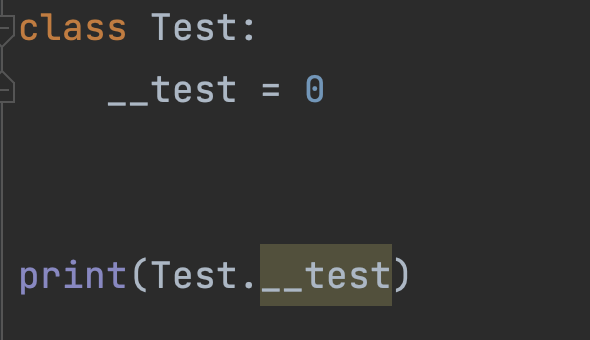
1. Какая разница между абстрактными классами и интерфейсами в Python?

Ответ: Абстрактный класс может иметь как абстрактные методы, так и неабстрактные. А вот интерфейс может иметь только абстрактные методы.

1. Как показать в классе что метод является абстрактым?

Ответ: с помощью abc.abstractmethod

1. Что будет результатом программы?

  
Ответ: Ошибка, по моему мнению из-за того что \_\_test - приватный

1. Что делает финализатор класса?

Ответ: Это магический метод \_\_del\_\_ он автоматически вызывается перед уничтожение экземпляра класса.

1. Распишите работу магических методов: new, call

Ответ: new – возвращает новый объект, который потом инициализируется с помощью \_\_init\_\_.

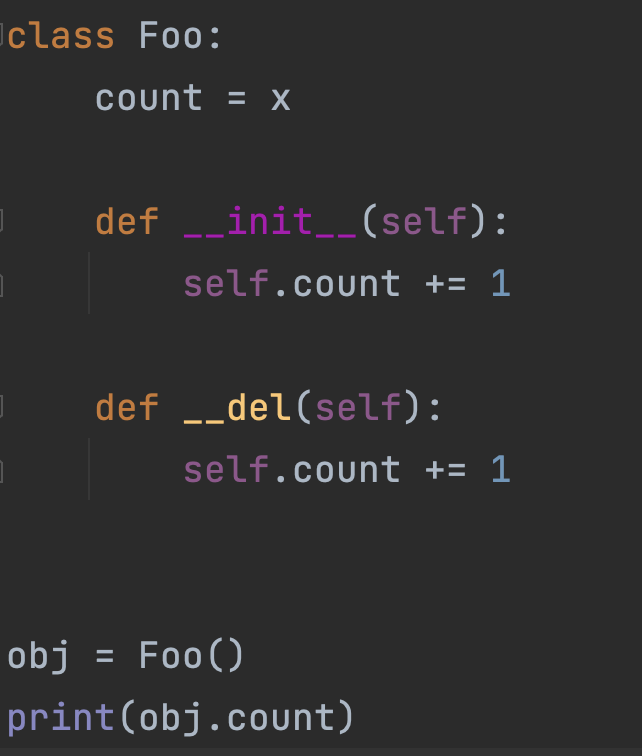
call – делает класс вызываемым

1. Если в классе определены два метода с одинаковыми именами и разными списками параметров, что будет результатом?
2. Значением поля класса по умолчанию может являться?

* значение переменной
* константа
* результат вызова функции
* возможность указания значений полей по умолчанию в Python не предусмотрена

Ответ: Константа

1. Укажите результат выполнения скрипта и почему:



Ответ: Ошибка х у нас не определен

1. Как указать в программе что класс A наследуется от B?

Ответ: class A(B)