Тест:

1. Что такое функция и как ее объявить?
2. Как создать функцию?

Ответ: В *python* функция определяется ключевым словом *def* и далее идет название функции и в круглых скобках параметры. Например def func(n):

1. Как показать что созданная функция возвращает результат?

Ответ: в функции возвращает результат оператор *return*.

1. В чем разница между параметрами и аргументами функции?

Ответ: Параметры функции задаются при создании функции, а аргументы это величины укалывающиеся при вызове этой функции.

1. Как рекурсивные функции работают в Python?

Ответ: Рекурсивная функция – это функция которая обращается сама к себе в собственном коде.

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

Ответ: Декоратор – это функция, которая позволяет обернуть другую функцию для расширения её функциональности без изменения её кода.

def ekz\_work(func):

def wrapper():

print("Выполняю тест")

func()

print("Выполняю задачи")

return wrapper

@ekz\_work

def rest():

print("Отдыхаю!")

rest()

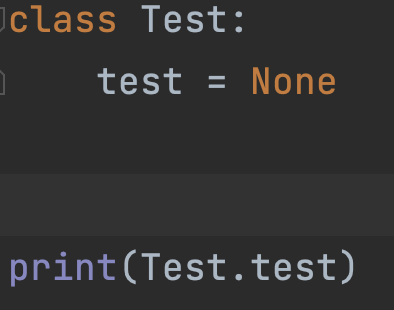
1. Как расшифровывается ООП?

Ответ: Объектно-ориентированное программирование.

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

Ответ: наследование, инкапсуляция и полиморфизм.

1. Что будет результатом программы и почему?



*Ответ: Результатом выполнения этой программы будет "None". Это происходит потому, что переменная "test" внутри класса Test равна None, и поскольку она статическая, она доступна без создания экземпляра класса*

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

Ответ: Полиморфизм

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

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

Ответ: private поля создаются с помощью двух подчеркиваний *\_\_field = 0.*

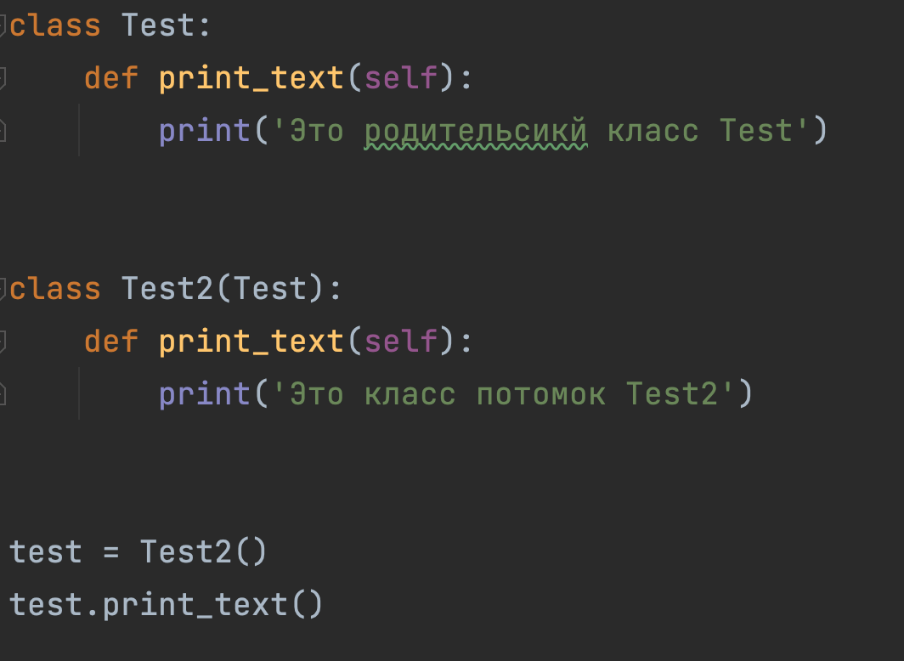
1. Как создать конструктор класса?

Ответ: Конструктор класса создается при помощи метода *\_\_init\_\_*

1. Как много конструкторов в классе может иметь Python?

Ответ: В *Python* в классе может быть только один конструктор *\_\_init\_\_.* Но при помощи декоратора можно создать альтернативный конструктор.

1. Что будет результатом данной программы



Ответ: Результатом выполнения этой программы будет вывод: *"Это класс потомок Test2".* *Test2* потомок класса *Test* но поскольку объект *test* принадлежит классу *Test2* то функция *print\_text* будет использована из класса *Test2.*

1. Какой параметр обязательно принимает в себя метод экземпляра?

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

Ответ: сам экземпляр класса.

1. Как использовать инкапсуляцию для защиты данных в классах?

Ответ: при помощи префиксов «\_» – защищенная переменная и «\_\_» – приватная переменная.

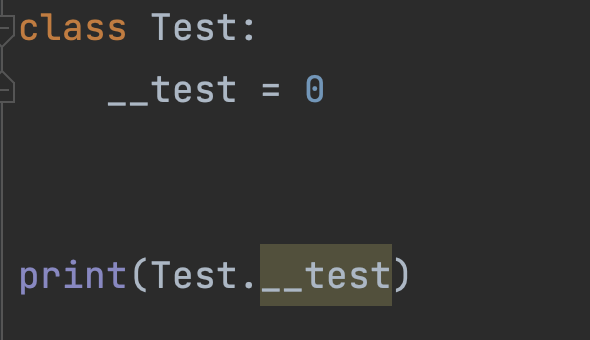
1. Какая разница между абстрактными классами и интерфейсами в Python?

Ответ: абстрактные классы могут содержать как реализованные методы, так и пустые методы, а интерфейсы в *Python* представляют собой абстрактные классы, содержащие только пустые методы без реализации.

1. Как показать в классе что метод является абстрактым?

Ответ: чтобы показать, что метод является абстрактным, нужно использовать декоратор *@abstractmethod* из модуля *abc*.

1. Что будет результатом программы?



Ответ: При попытке выполнить данный код возникнет ошибка, так как переменная *test* является "приватным" атрибутом класса *Test*.

1. Что делает финализатор класса?

Ответ: метод *\_\_del\_\_* представляет собой специальный метод в классе, который определяет действия, которые будут выполнены при удалении объекта.

1. Распишите работу магических методов: new, call

Ответ: *\_\_new\_\_* – статический метод для создания новых экземпляров класса до инициализации, *\_\_call\_\_* – магический метод, позволяющий экземпляру класса быть вызванным как функция.

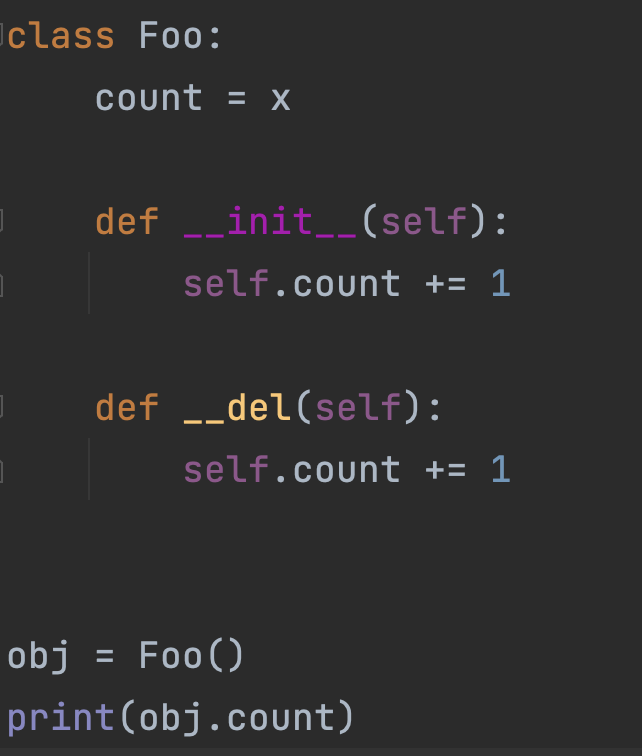
1. Если в классе определены два метода с одинаковыми именами и разными списками параметров, что будет результатом?

Ответ: последний определенный метод с заданным именем и этим же количеством аргументов будет полностью заменять предыдущие определения методов с тем же именем.

1. Значением поля класса по умолчанию может являться?

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

1. Укажите результат выполнения скрипта и почему:



Ответ: В результате выполнения скрипта будет ошибка потому что *x* не имеет значения и переменная *count* не определена. При выводе значения *obj.count* так же будет ошибка поскольку *count* не определена.

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

Ответ: При указании класса *А* нужно в круглых скобках указать родительский класс *В*. Например так: *class A(B):*

1. Что такое API и для чего оно используется?

Ответ: *API  (Application Programming Interface)* – это набор определенных правил и протоколов определяющих структуру запросов и ответов для обмена данными между различными программами или компонентами программного обеспечения

1. Какие преимущества предоставляют API?

Ответ: *API* предоставляет преимущества в виде возможности интеграции, масштабируемости, удобства, расширения возможностей и сокращения затрат на разработку.

1. Расскажите о различных типах HTTP-запросов.

Ответ: *GET* – получение информации о данных или списка объектов; *DELETE* – удаление данных; *POST* ­– добавление или замена данных; *PUT* – регулярное обновление данных.

1. Какие основные методы HTTP используются в RESTful API?

Ответ: *GET* – получение информации о данных или списка объектов; *DELETE* – удаление данных; *POST* ­– добавление или замена данных; *PUT* – регулярное обновление данных; *PATCH* – используется для частичного изменения ресурса.

1. Что такое запросы GET, POST, PUT и DELETE? Как они используются в API?

Ответ: *GET* – получение информации о данных или списка объектов; *DELETE* – удаление данных; *POST* ­– добавление или замена данных; *PUT* – регулярное обновление данных. В *API* они определяют, как выполнять операции с ресурсами, такими как чтение, создание, обновление и удаление.

1. Что такое заголовки HTTP? Какие заголовки часто используются в API?

Ответ: Заголовки *HTTP* содержат дополнительную информацию о передаваемых данных, такую как тип содержимого, аутентификация, кэширование и другие параметры. В *API* часто используются заголовки *Content-Type, Authorization, Accept, Cache-Control* и *X-Requested-With*.