**Тест:**

1. Что такое функция и как ее объявить?

Ответ: Функции – многократно используемые фрагменты программы. Объявить словом def, затем имя функции и () для принятия параметров. Вызывается по имени.

1. Как создать функцию?

Ответ:

def name\_function(parameters)

блок/тело самой функции

1. Как показать, что созданная функция возвращает результат?

Ответ: Прописать, что должна возвращать функция через return

1. В чем разница между параметрами и аргументами функции?

Ответ: Параметры – прописываем в функцию и с ними в блоке/теле функции работаем. Аргументы – вводим, передаем в функции при её вызове.

1. Как рекурсивные функции работают в Python?

Ответ: Вызывают сами себя в теле функции.

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

Ответ: функции которые позволяют расширить функционал существующей функции без её изменения.

def capitalize\_name\_decorator(function):

def wrapper():

name = function()

capitalize\_name = name.capitalize()

return capitalize\_name

return wrapper

@capitalize\_name\_decorator

def input\_name():

name = input('Enter your name: ')

return name

enter\_name = input\_name()

print(enter\_name)

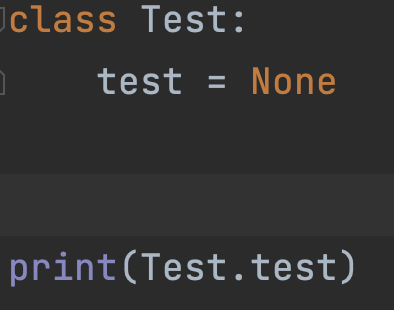
1. Как расшифровывается ООП?

Ответ: Объектно-ориентированное программирование.

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

Ответ: Инкапсуляция, Наследование, Полиморфизм, Абстракция

1. Что будет результатом программы и почему?



Ответ: None. Был вызван атрибут класса, а он равен None.

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 – этот

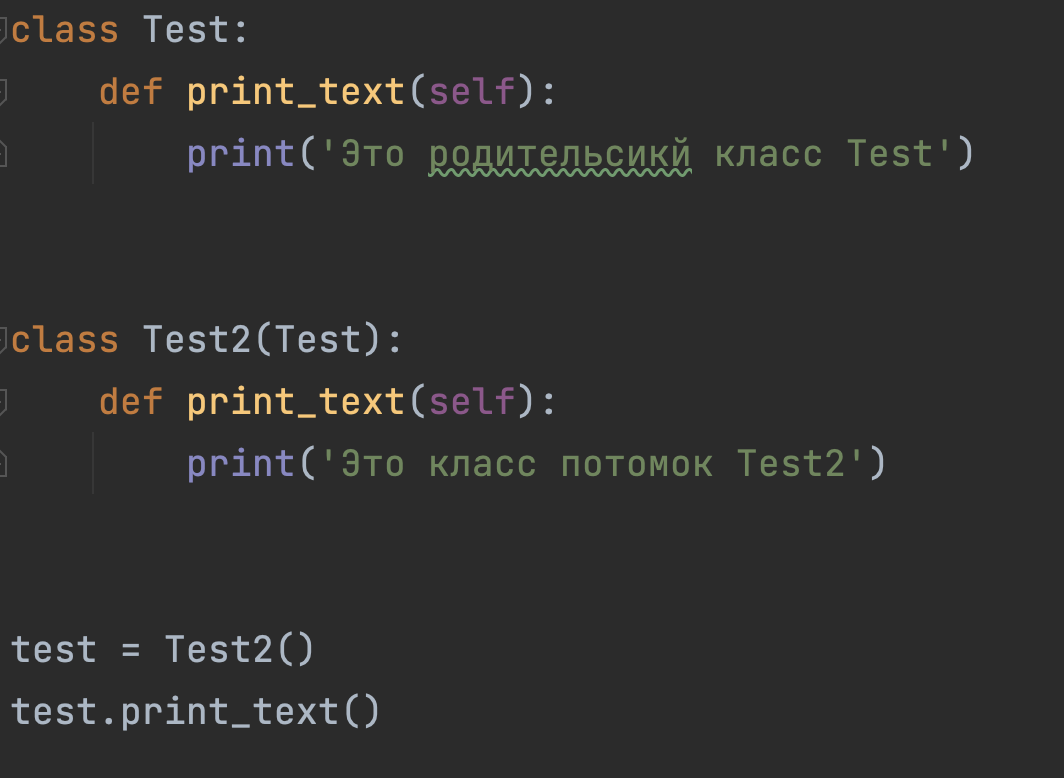
1. Как создать конструктор класса?

Ответ: Методом \_\_init\_\_(self, тут параметры через запятую)

1. Как много конструкторов в классе может иметь Python?

Ответ: 1

1. Что будет результатом данной программы



Ответ: ‘Это класс потомок Test2’

1. Какой параметр обязательно принимает в себя метод экземпляра?

* тип объекта
* сам экземпляр класса – этот (self)
* название класса
* таких аргументов нет

1. Как использовать инкапсуляцию для защиты данных в классах?

Ответ: через два нижних подчеркивания, как на скрине пункта 19.

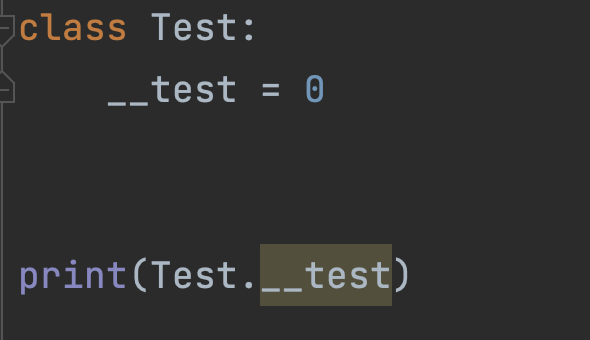
1. Какая разница между абстрактными классами и интерфейсами в Python?

Ответ: Абстрактные классы могут содержать методы с реализацией, а интерфейсы (в виде абстрактных классов) содержат только абстрактные методы без реализаций.

1. Как показать в классе что метод является абстрактым?

Ответ: использовать декоратор @abstractmethod из модуля abc

1. Что будет результатом программы?



Ответ: Ошибка

1. Что делает финализатор класса?

Ответ: вызывается перед удалением

1. Распишите работу магических методов: new, call

Ответ: new – создаёт и возвращает новый экземпляр класса, вызывается перед init и выделяет память для нового объекта. call – позволяет вызывать экземпляры класса, как функции (будет вызван метод call).

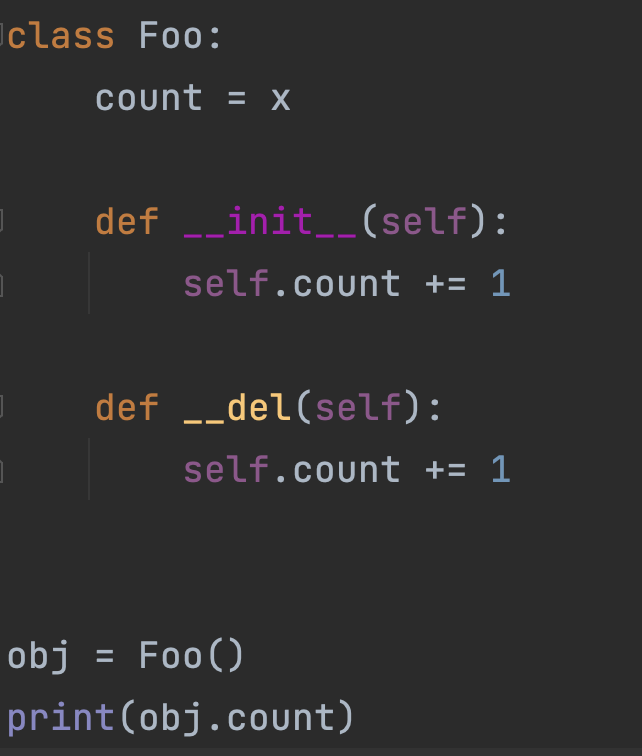
1. Если в классе определены два метода с одинаковыми именами и разными списками параметров, что будет результатом?

Ответ: последний

1. Значением поля класса по умолчанию может являться?

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

1. Укажите результат выполнения скрипта и почему:



Ответ: ошибка, потому что count = x, а x нигде не задан и не взят в кавычки.

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

class A(B):

Практическая часть:

1. Требуется проверить, возможно ли из представленных отрезков условной длины сформировать треугольник. Для этого создайте класс TriangleChecker, принимающий только положительные числа. С помощью метода is\_triangle() возвращаются следующие значения (в зависимости от ситуации):

* Ура, можно построить треугольник!
* С отрицательными числами ничего не выйдет!
* Жаль, но из этого треугольник не сделать.

1. Задача

Есть Помидор со следующими характеристиками:

1. Индекс
2. Стадия зрелости(стадии: Отсутствует, Цветение, Зеленый, Красный)

Помидор может:

1. Расти (переходить на следующую стадию созревания)
2. Предоставлять информацию о своей зрелости

Есть Куст с помидорами, который:

1. Содержит список томатов, которые на ней растут

И может:

1. Расти вместе с томатами
2. Предоставлять информацию о зрелости всех томатов
3. Предоставлять урожай

И также есть Садовник, который имеет:

1. Имя
2. Растение, за которым он ухаживает

И может:

1. Ухаживать за растением
2. Собирать с него урожай

Задание:

##### **Класс Tomato**

1. Создайте класс Tomato
2. Создайте статический атрибут states, который будет содержать все стадии созревания помидора
3. Создайте метод \_\_init\_\_(), внутри которого будут определены два приватных атрибута: 1) \_index - передается параметром и 2) \_state - принимает первое значение из словаря states
4. Создайте метод grow(), который будет переводить томат на следующую стадию созревания
5. Создайте метод is\_ripe(), который будет проверять, что томат созрел (достиг последней стадии созревания)

##### **Класс TomatoBush**

1. Создайте класс TomatoBush
2. Определите метод \_\_init\_\_(), который будет принимать в качестве параметра количество томатов и на его основе будет создавать список объектов класса Tomato. Данный список будет храниться внутри атрибута tomatoes.
3. Создайте метод grow\_all(), который будет переводить все объекты из списка томатов на следующий этап созревания
4. Создайте метод all\_are\_ripe(), который будет возвращать True, если все томаты из списка стали спелыми
5. Создайте метод give\_away\_all(), который будет чистить список томатов после сбора урожая

##### **Класс Gardener**

1. Создайте класс Gardener
2. Создайте метод \_\_init\_\_(), внутри которого будут определены два атрибута: 1) name - передается параметром, является публичным и 2) \_plant - принимает объект класса TomatoBush, является приватным
3. Создайте метод work(), который заставляет садовника работать, что позволяет растению становиться более зрелым
4. Создайте метод harvest(), который проверяет, все ли плоды созрели. Если все - садовник собирает урожай. Если нет - метод печатает предупреждение.
5. Создайте статический метод knowledge\_base(), который выведет в консоль справку по садоводству.

##### **Тесты (main)**

1. Вызовите справку по садоводству
2. Создайте объекты классов TomatoBush и Gardener
3. Используя объект класса Gardener, поухаживайте за кустом с помидорами
4. Попробуйте собрать урожай
5. Если томаты еще не дозрели, продолжайте ухаживать за ними
6. Соберите урожай

3) Создайте систему управления банковскими счетами, которая позволяет создавать, управлять и выполнять операции с банковскими счетами различных клиентов.

1. Реализуйте класс Client, представляющий клиента банка. Класс должен иметь атрибуты name (имя клиента) и id (уникальный идентификатор клиента).
2. Реализуйте класс BankAccount, представляющий банковский счет. Класс должен иметь атрибуты account\_number (номер счета), balance (баланс счета) и client (объект типа Client, которому принадлежит счет). Класс также должен иметь методы deposit(amount) и withdraw(amount), которые позволяют пополнить или снять деньги со счета.
3. Реализуйте класс Bank, представляющий банк. Класс должен иметь атрибут accounts, который является словарем, где ключами являются номера счетов, а значениями - объекты типа BankAccount. Класс также должен иметь методы create\_account(client, initial\_balance) для создания нового счета и get\_account(account\_number) для получения счета по его номеру.
4. Добавьте в класс Bank методы для выполнения переводов между счетами (transfer(sender\_account, receiver\_account, amount)), а также для получения общего баланса клиента (get\_total\_balance(client)), который включает сумму денег на всех его счетах.
5. Реализуйте обработку ошибок, например, недостаточно средств на счете при снятии денег или отсутствие счета при переводе.

4) Напишите программу с классом Student, в котором есть три атрибута: name, groupNumber и age. По умолчанию name = Ivan, age = 18, groupNumber = 10A. Необходимо создать пять методов: getName, getAge, getGroupNumber, setNameAge, setGroupNumber. Метод getName нужен для получения данных об имени конкретного студента, метод getAge нужен для получения данных о возрасте конкретного студента, vетод setGroupNumberнужен для получения данных о номере группы конкретного студента. Метод SetNameAge позволяет изменить данные атрибутов установленных по умолчанию, метод setGroupNumber позволяет изменить номер группы установленный по умолчанию. В программе необходимо создать пять экземпляров класса Student, установить им разные имена, возраст и номер группы.

5) Класс «Волшебник» (Wizard)

Экземпляр класса при инициализации принимает аргументы:

– имя;

– рейтинг;

– на какой возраст выглядит.

Класс должен обеспечивать функциональность:

– change\_rating(value) – изменять рейтинг на значение value; не может стать больше 100 и

меньше 1, изменяется только до достижения экстремального значения; при увеличении

рейтинга уменьшается возраст на abs(value) // 10, но только до 18, дальше не уменьшается;

при уменьшении рейтинга возраст соответственно увеличивается;

– к экземпляру класса можно прибавить строку: (wd += string), значение рейтинга

увеличивается на ее длину, а возраст, соответственно, уменьшается на длину // 10, условия

изменения такие же;

– экземпляр класса можно вызвать с аргументом-числом; возвращает значение: (аргумент

- возраст) \* рейтинг;

\_\_str\_\_() – возвращает строку:

“Wizard <name> with <rating> rating looks <age> years old”

– экземпляры класса можно сравнивать: сначала по рейтингу, затем по возрасту, затем по

имени по алфавиту; для этого нужно реализовать методы сравнения: <, >, <=, >=, ==, !=.

6) Класс «Сотрудник компании» Worker  
Экземпляр класса при инициализации принимает аргументы: имя, должность и стаж работы сотрудника, метод print\_info() выводит информацию о сотруднике в формате: «Имя: Василий Должность: Системный администратор :Стаж: 3 года» При выводе стажа нужно учитывать, что «года» должно заменяться на «лет» или «год» в зависимости от числа.

worker1 = Worker("Алексей", "Программист", 17)

worker1.print\_info()

worker2 = Worker("Анна", "Маркетолог", 2)

worker2.print\_info()

worker3 = Worker("Дмитрий", "Аналитик", 1)

worker3.print\_info()

7) Класс ПЕРСОНА, экземпляр класса инициализируется аргументами фамилия, дата

рождения и содержит методы, позволяющие вывести информацию о персоне, а также определить ее возраст.

Дочерние классы: АБИТУРИЕНТ (фамилия, дата рождения, факультет),

СТУДЕНТ(фамилия, дата рождения, факультет, курс), ПРЕПОДАВАТЕЛЬ (фамилия, дата рождения, факультет, должность, стаж), содержат свои методы вывода информации.

Создайте список из n персон, выведите полную информацию из базы, а также

организуйте поиск персон, чей возраст попадает в заданный диапазон.

8\*\*) Шахматный король ходит по горизонтали, вертикали и диагонали,  
  
 но только на 1 клетку. Даны две различные клетки шахматной доски,  
  
определите, может ли король попасть с первой клетки на вторую одним ходом.  
  
  
 Пример  
 Cell 1 coordinates:  
 >>4, 4  
 Cell 2 coordinated:  
 >>5, 5  
 YES  
 Конь  
 Определите, может ли конь попасть с первой клетки на вторую одним ходом.  
 Ладья  
 Определите, может ли ладья попасть с первой клетки на вторую одним ходом.  
 Ферзь  
 Определите, может ли ферзь попасть с первой клетки на вторую одним ходом.  
 Слон  
 определите, может ли слон попасть с первой клетки на вторую одним ходом.  
Пример реализации  
