Урок 6. ООП. Вводный урок по работе с классами(шаблонами).

ФИО_____

Тест:

1. *Что верно про атрибут класса?*
   A) Общий для всех экземпляров B) Только у одного экземпляра
   C) Обязателен всегда D) Это метод

2. *Что выведет данная программа?*

```
class A:  3 usages
    v = 1

a1 = A()
a2 = A()
a1.v = 5
print(A.v, a1.v, a2.v)
```

3. *Когда вызывается __init__?*

   _____

4. *Как называется «экземпляр класса»?*
   A) Метод B) Поле C) Объект D) Пакет
5. *Что выведет данная программа?*

```
class C:  2 usages
    def __init__(self, x):
        self.x = x

    def inc(self, d=1):  1 usage
        self.x += d

c = C(10)
C.inc(c,  d: 2)
print(c.x)
```

_____

Теория:

Основы ООП:

Краеугольное понятие в ООП — **объект**. Это такой своеобразный контейнер, в котором сложены данные и прописаны действия, которые можно с этими данными совершать.

Что такое класс и объект

- **Класс** — «чертёж», по которому создаются объекты (экземпляры).
- **Объект (экземпляр)** — «вещь» с **данными** (атрибутами) и **поведением** (методами).

Например, у нас есть класс «Кошка», обладающий атрибутами «порода», «окрас», «возраст» и методами «мяукать», «мурчать», «умываться», «спать». Присваивая атрибутам определённые значения, можно создавать вполне конкретные объекты.

Допустим:

- Порода = абиссинская.
- Окрас = рыжий.
- Возраст = 4.

Таким образом мы можем создать сколь угодно много разных кошек.

```
class Cat():
  def __init__(self, breed, color, age):
     self.breed = breed
     self.color = color
     self.age = age

  def meow(self):
     print('Мяу!')

  def purr(self):
     print('Мрррр')

cat1 = Cat('Абиссинская', 'Рыжая', 4)
cat2 = Cat('Британская', 'Серая', 2)
```

Описание конструкции класса:

Метод __init__ – инициализатор класса. Он вызывается сразу после создания объекта, чтобы присваивать значения динамическим атрибутам. **self** – ссылка на текущий объект, она даёт доступ к атрибутам и методам, с которыми вы работаете.

Слово **self** общепринятое, но не обязательное, вместо него можно использовать любое другое. Однако это может запутать тех, кто будет читать ваш код. Названия классов принято писать с прописной буквы, а объектов – со строчной.
Итак, мы создали класс **Cat**, в котором объявили три атрибута: порода – **breed**, цвет – **color** и возраст – **age**. А ещё добавили два метода, чтобы наша кошка умела мяукать – **meow**() и мурчать – **purr**().
Экземпляры класса: **cat1** и **cat2.**
Атрибут класса – общий атрибут для всего класса и для каждого экземпляра класса, **атрибут экземпляра класса** – у каждого объекта свои (self.x).
Виды методов класса:

- **Обычный метод** получает self.
- **Класс-метод** получает сам класс (cls), помечается @classmethod.
- **Стат-метод** не получает ни self, ни cls, помечается @staticmethod.

Методы отличные от обычного помечаются декораторами:

```
class MathBox:
    version = "1.0"

    @classmethod
    def info(cls):
        return f"MathBox v{cls.version}"

    @staticmethod
    def add(a, b):
        return a + b

print(MathBox.info())  # MathBox v1.0
print(MathBox.add(2, 3))  # 5
```

Задачи:

Варианты атрибутов и экземпляров на ваше усмотрение (если не указано другое), но они должны присутствовать в каждой задаче, вариант задания аргументов через экземпляры тоже любое(позиционные, по умолчанию и т.д.):

1. ***Класс Rectangle***
   Поля: w, h. Методы: area(), perimeter(), scale(k) — умножает стороны на k (меняет текущий объект).
2. ***Класс Point***
   Поля: x, y. Методы: move(dx, dy), distance_to(other) — расстояние до другой точки (без библиотек: через корень **0.5).
3. ***Класс BankAccount***
   Поля: owner, balance (старт 0). Методы: deposit(amount), withdraw(amount) — если не хватает денег, ничего не делать (или вернуть False), get_balance().

4. ***Класс TodoList***
   Методы: add(text), remove(text), has(text) -> bool, list() — вернуть копию списка задач. Добавь count() — сколько задач.
5. ***Класс Timer***
   Методы: start(), stop(), elapsed() — вернуть суммарное время между всеми стартами и стопами. Подсказка: хранить «идёт ли сейчас» и «накопленное время». (Можно использовать time.perf_counter().)
6. ***Класс Student***
   Поля: name, grades (список). Методы: add_grade(x), avg() — средняя (если оценок нет — вернуть 0). best() — максимальная или None, если нет оценок.
7. ***Класс ShoppingCart***
   Методы: add_item(name, price), remove_item(name), total() — сумма цен, items() — копия списка кортежей (name, price).

8. Написать простую текстовую игру «Tanks».

Игра должна быть реализована в парадигме ООП.

Класс «Tank» должен иметь атрибуты: модель(строка) «model», броня танка (целочисленное значение) «armor», урон танка «damage» (целочисленное значение, определяется случайным образом в заданном диапазоне, который передается в метод инициализации класса «min_damage» и «max_damage») и уровень здоровья «health». Атрибуты задаются пользователем в методе инициализации класса. Также должны быть реализованы следующие методы класса:

1. Метод «print_info», который выводит на консоль информацию о танке в формате: "«модель танка» имеет лобовую броню «броня танка» мм. при «здоровье танка» ед. здоровья и урон в «урон танка» единиц".

2. Метод стрельбы «shot», принимает в качестве параметра принимает вражеский танк «enemy» - экземпляр класса «Tank». Метод выводит на консоль 2 состояния:

- "Экипаж танка «модель вражеского танка» уничтожен", если в результате выстрела здоровье вражеского танка опустилось до нуля и меньше.
- "«модель стреляющего танка»: Точно в цель, у противника «модель вражеского танка» осталось «здоровье вражеского танка» единиц здоровья", если в результате выстрела здоровье вражеского танка осталось больше нуля. Для реализации уменьшения здоровья танка в результате выстрела необходимо использовать вспомогательный метод «health_down».

3. Метод изменения уровня здоровья танка при попадании в него «health_down». Метод принимает на вход урон вражеского танка и корректирует здоровье танка в зависимости от следующей формулы: «здоровье танка» = «здоровье танка» - «урон вражеского танка» / «броня танка» Также метод должен выводить на консоль информацию в следующем формате: "«модель танка»: Командир, по экипажу «модель танка» попали, у нас осталось «здоровье танка» очков здоровья"

Во всех методах необходимо использовать форматированный вывод через f-строк. Используйте в этой задача **аннотации типов.**