ПРАКТИКА 12

1. Классы
2. Объекты
3. Методы
4. Поля
5. Параметр self
6. Магические методы
7. Отношения между классами

ЗАДАЧИ:
ЗАДАЧА 0

Создайте класс Soda (для определения типа газированной воды), принимающий 1 аргумент при инициализации (отвечающий за добавку к выбираемому лимонаду). В этом классе реализуйте метод show_my_drink(), выводящий на печать «Газировка и {ДОБАВКА}» в случае наличия добавки, а иначе отобразится следующая фраза: «Обычная газировка».

Написать класс Point. Написать метод/функцию для нахождения расстояния между двумя точками. Создайте класс: отрезок. Переопределите оператор сложения двух точек в линию(для переопределения оператора сложения использовать магический метод __add__(self, other), а для проверки типов использовать функцию isinstance).

Статический метод вызывается при обращении к классу, а не его экземпляру. Пример:

In [1]:
```
class Example:
    @staticmethod
    def example_static():
        print('Я статический метод')
Example.example_static()
```

```
Я статический метод
```

Реализовать свой аналог библиотеки math, используя ООП и статические методы (достаточно 1 метода).

ЗАДАЧИ:
ЗАДАЧА 3

2.1

Напишите программу по следующему описанию. Есть класс "Воин". От него создаются два экземпляра-юнита. В случайном порядке они бьют друг друга. После каждого удара надо выводить сообщение, какой юнит атаковал, и сколько у противника осталось здоровья. Как только у кого-то заканчивается ресурс здоровья, программа завершается сообщением о том, кто одержал победу.

2.2

Добавьте другого персонажа. Например, мага. Дайте каждому классу персонажа свою спецспособность. Например, у война суперспособность будет наносить критический удар, а у мага уворот.

2.3

Добавьте броню и виды оружия.

2.4

Сбалансируйте игру

Придумать и реализовать пример множественного наследования.

КОНТРОЛЬНЫЕ ВОПРОСЫ