1. Инициализировать угол - Angles
2. Инициализировать полигон – Polygon
   1. Передаю аргументы:
      1. from\_local - либо заготовленные данные в Полигон передаваться будут, либо сам пользователь загрузит (True, если заготовленные данные) и надо будет ещё номер (порядковый номер) этих данных прокинуть. На фронте типа появится доп. Окошко, в которое надо будет прокидывать количество допускаемых данных.
      2. data - Тут либо сразу данные (словарик/json) прокинутся, либо просто порядковый номер исходных данных из «бд»
3. Инициализировал объект, которым буду читать исходные данные - DB (внутри Полигона)

Может и записывать (сохранять тоже стоит через экземпляр DB)?

1. Работаю в полигоне
   1. Прочитываю все данные - get\_all\_data()
   2. Внутри Полигона разбиваю инфу на углы, на дир. Угол, на координаты исходные, и получу «вспомогательную сторону» (для информации, по какой формуле вычислять теоретическую сумму углов)
   3. Формирую список всех углов для дальнейшей обработки:
      1. Вычисляю теоретическую сумму горизонтальных углов - theoretical\_sum
      2. Вычисляю практическую сумму горизонтальных углов
      3. Вычисляю невязку
      4. Вычисляю теоретическую невязку (в зависимости от прибора, от его точности, двойная точность прибора, всякое такое)
      5. Вычисляю поправку, которую нужно внести в каждый угол

НАДО КАК-ТО НАУЧИТЬ класс Angle если я передаю в DMS, то один вариант инициализации, если DD, то другой вариант инициализации. Например, буду вычислять сумму углов или ещё что-нибудь, а мне надо вычислять будет потом DMS и т.п.

Либо передавать если передаю массив аргументов, то по ним надо пройтись и раскидать в нужные атрибуты, а если одно значение передаю, то это DD

# Надо бы

Написать класс для взаимодействия с АПИ

Там должно быть что-то типа как класс DB, методы на получение и на отдачу данных, их буду использовать внутри класса Полигон, также как с БД

Либо просто модифицировать модуль БД