Код на Python для создания обертки над API управления дроном:

import requests

class DroneAPI:

def \_\_init\_\_(self, api\_key):

self.api\_key = api\_key

self.base\_url = "https://droneapi.com/v1/"

def take\_off(self):

url = self.base\_url + "takeoff"

params = {"api\_key": self.api\_key}

response = requests.post(url, params=params)

return response.status\_code == 200

def land(self):

url = self.base\_url + "land"

params = {"api\_key": self.api\_key}

response = requests.post(url, params=params)

return response.status\_code == 200

def move\_left(self, distance):

url = self.base\_url + "move"

params = {"api\_key": self.api\_key, "direction": "left", "distance": distance}

response = requests.post(url, params=params)

return response.status\_code == 200

# и т.д. для других команд управления дроном

my\_drone = DroneAPI(api\_key="ABCD1234")

my\_drone.take\_off()

my\_drone.move\_left(50)

my\_drone.land()

Этот класс DroneAPI инкапсулирует детали взаимодействия с API дрона и предоставляет простой интерфейс для вызова методов take\_off(), land(), move\_left() и т.д.

Основные преимущества:

* Абстракция сложности API - не нужно знать url и параметры каждого запроса
* Централизованное управление аутентификацией через api\_key
* Проверка кодов ответа на ошибки
* Простой и понятный интерфейс для управления дроном в коде

Таким образом, использование обертки позволяет сделать код управления дроном более чистым и удобным.

* Импортируем библиотеку requests - она нужна для отправки HTTP запросов.
* Определяем класс DroneAPI - это наша обертка над API управления дроном.
* В **init** принимаем api\_key и сохраняем его как атрибут экземпляра класса. Также сохраняем базовый URL API.
* Метод take\_off отправляет запрос на взлет дрона. Формирует нужный URL, добавляет api\_key в параметры и отправляет POST запрос. Проверяет код ответа на 200 OK.
* Аналогично реализован метод land для посадки дрона.
* Метод move\_left позволяет двигать дрон влево на указанное расстояние. Опять же формирует запрос и отправляет его, проверяя ответ.
* Создаем экземпляр DroneAPI, передавая api\_key.
* Вызываем методы для взлета, перемещения и посадки дрона. Класс заботится обо всех деталях взаимодействия с API.
* Таким образом мы получаем простой и понятный интерфейс для управления дроном, не заморачиваясь с деталями API