

ПРОГРАММА

ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ

**“Python-разработка для БПЛА: паттерны проектирования, API-интерфейсы и библиотеки для оптимизации решений”**

**Задание 2.**

**Применение ООП в контексте разработки для БПЛА**

Преподаватель курса

**Максимов Егор Васильевич**

Выполнил

**Бубнович Антон Владимирович**

**№ потока** [**БПЛА\_256-1/2024**](https://www.odin.study/ru/Cohort/Info/42290)

**Москва 2024 г.**

**Задание 2. Применение ООП в контексте разработки для БПЛА**

**Инструкция:**

**Шаг 1.** Создать классы Drone (беспилотник), Camera (камера), GPS (система GPS) и FlightController (контроллер полета).

class Camera():

def \_\_init\_\_(self):

class GPS():

def \_\_init\_\_(self):

pass

class FlightController():

def \_\_init\_\_(self):

pass

class Drone():

def \_\_init\_\_(self):

pass

**Шаг 2.** Добавить в класс Drone поля для объектов Camera, GPS и FlightController. Таким образом беспилотник будет содержать эти компоненты.

class Drone(FlightController, GPS, Camera):

def \_\_init\_\_(self, id="ABC", model="copter"):

self.id = id

self.model = model

self.camera = Camera()

self.gps = GPS(coordinates=(0.0, 0.0))

self.fight\_controller = FlightController()

**Шаг 3.** Добавить в класс FlightController методы для управления полетом беспилотника - взлет, посадка, изменение высоты и координат. Вызывать эти методы для объекта класса Drone.

class FlightController:

def \_\_init\_\_(self):

self.coordinates = GPS()

self.height = 0

self.map\_flight = []

self.is\_flying = False

def start(self):

print(f"Двигатели запущены")

def takeoff(self):

print(f"Начинаем взлет")

def left(self):

print(f"Поворачиваем налево")

def right(self):

print(f"Поворачивам направо")

def adove(self):

print(f"Поднимаемся выше")

def below(self):

print(f"Снижаемся")