

5. Модуль №5. Основы программирования на языке Python		10	5	5	
5.1.	Знакомство с языком Python	2	2	0	
5.2.	Переменные и выражения.	2	0	2	Практическая работа
5.3.	Условные выражения	2	1	1	Практическая работа
5.4.	Циклы.	2	1	1	Практическая работа
5.5.	Функции	2	1	1	Практическая работа

7. Модуль №7. Использование датчиков БАС и сбор данных		8	4	4	
7.1.	Сенсоры и датчики для сбора данных	2	2	0	
7.2.	Датчики: акселерометр,гироскоп, дальномер GPS.	2	1	1	
7.3.	Датчики при сборке вмастерской.	2	0	2	Практика сборки

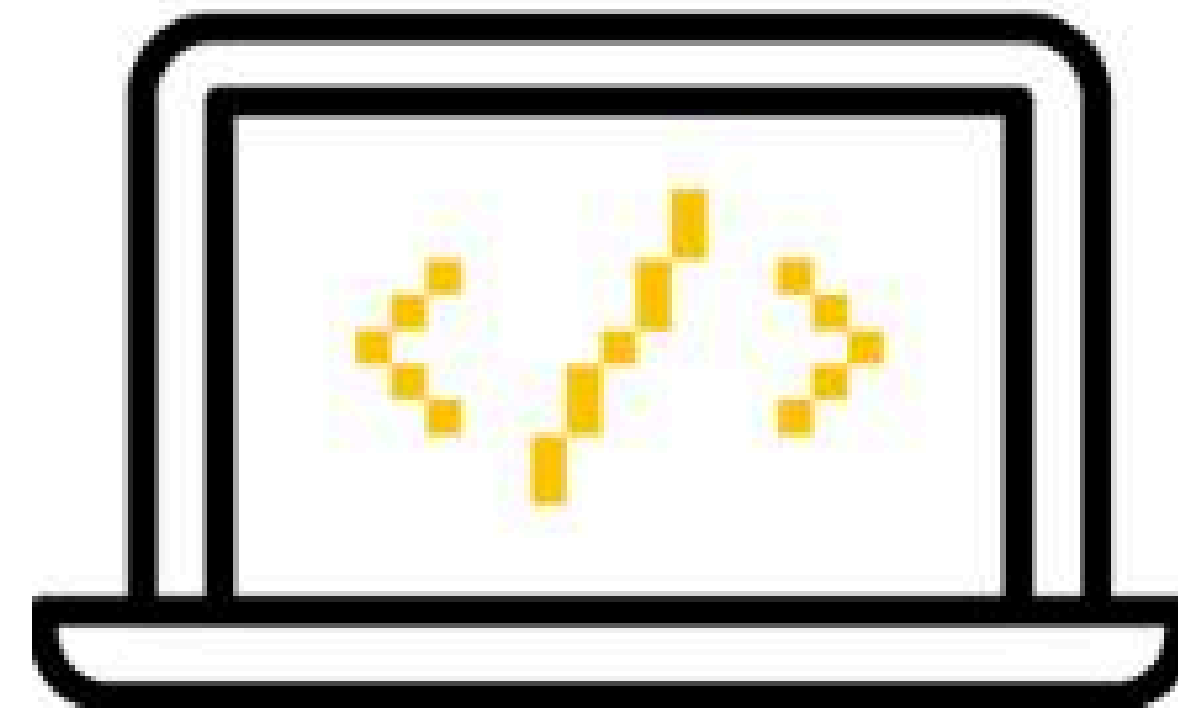
6. Модуль №6. Программирование для полетов внутри помещения		8	0	8	
6.1.	Основы программирования БАС на Python.	2	0	2	Практическая работа
6.2.	Работа со списком данных.	2	0	2	Практическая работа
6.3.	Разработка алгоритма автономного полета БАС.	2	0	2	Практическая работа
6.4.	Создать скрипт на языке программирования Python для самостоятельного управления квадрокоптером в помещении без использования сигнала GPS.	2	0	2	Практическое задание

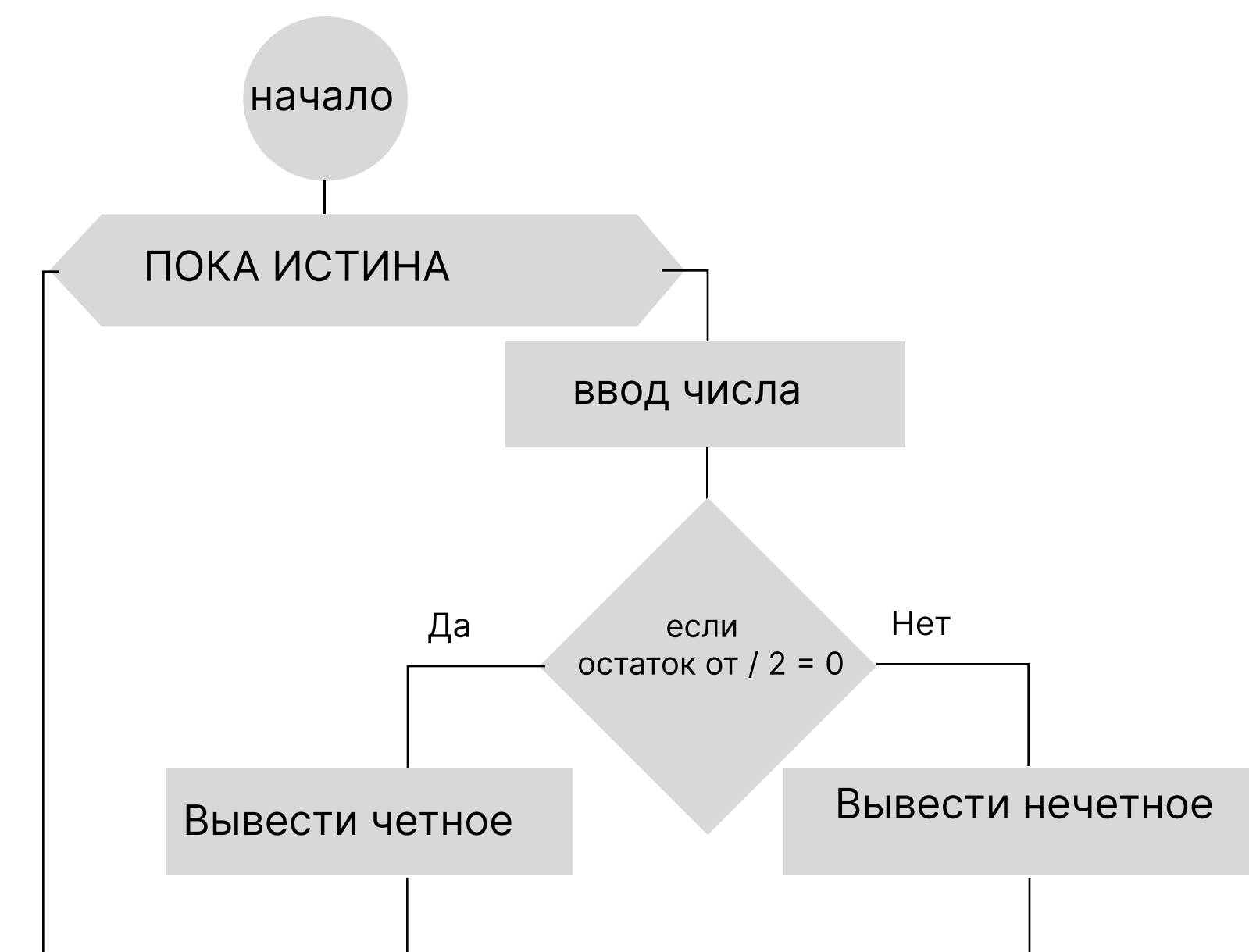
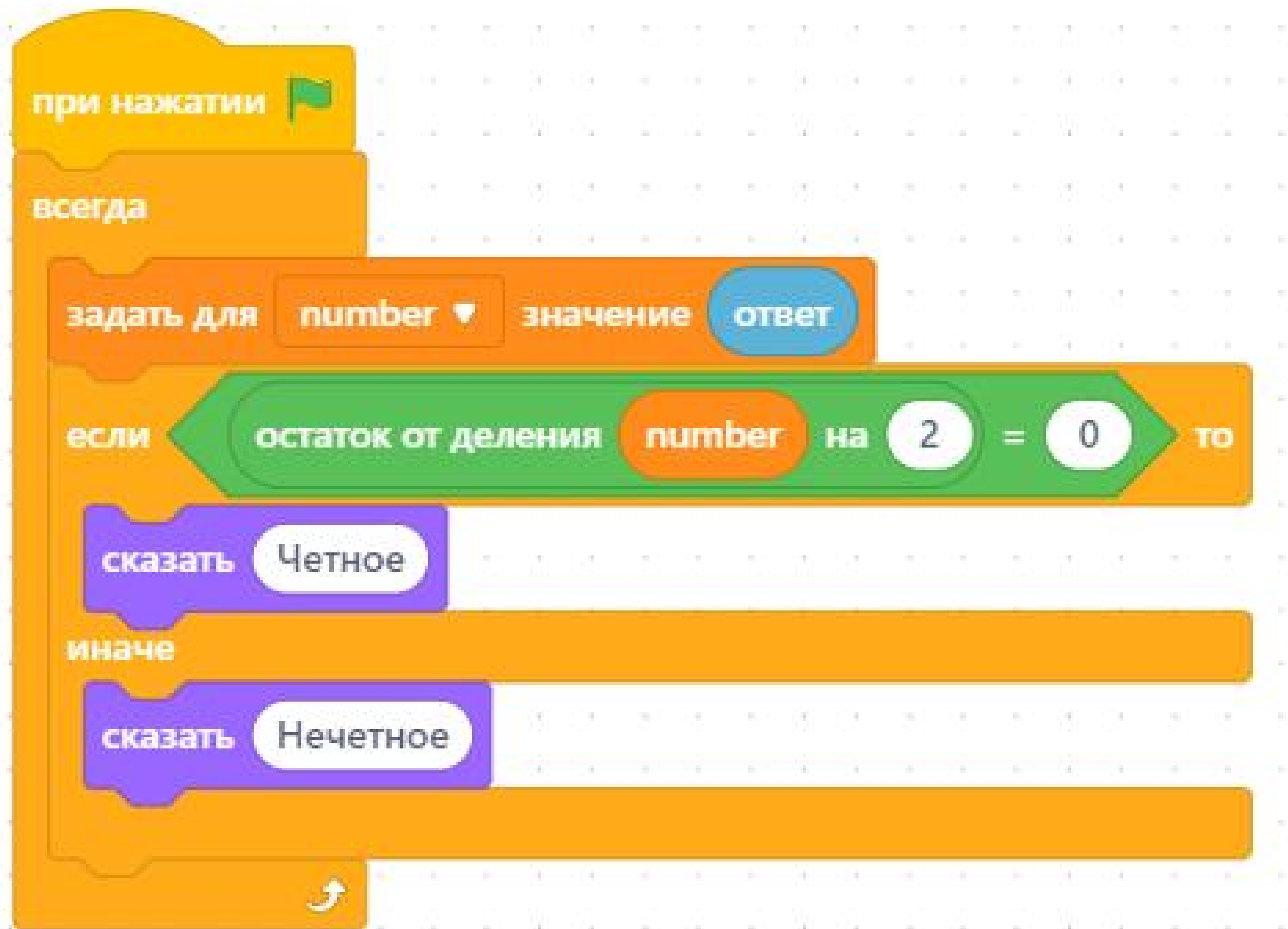
8. Модуль №8. Обработка и анализ данных полета БАС		4	4	0	
8.1.	Тема 1. Сбор, обработка и анализ данных фотограмметрической съемки.	2	2	0	
8.2.	Тема 2. Сбор, обработка и анализданных аэрофотосъёмки.	2	2	0	
9. Модуль №9. Применение БАС в различных отраслях		4	2	2	
9.1.	Технология применения БАСв геодезии и картографии.	2	2	0	
9.2.	Технологии применения БАС в других отраслях	2	0	2	Проектная работа.

ГРУППА КОМПАНИЙ

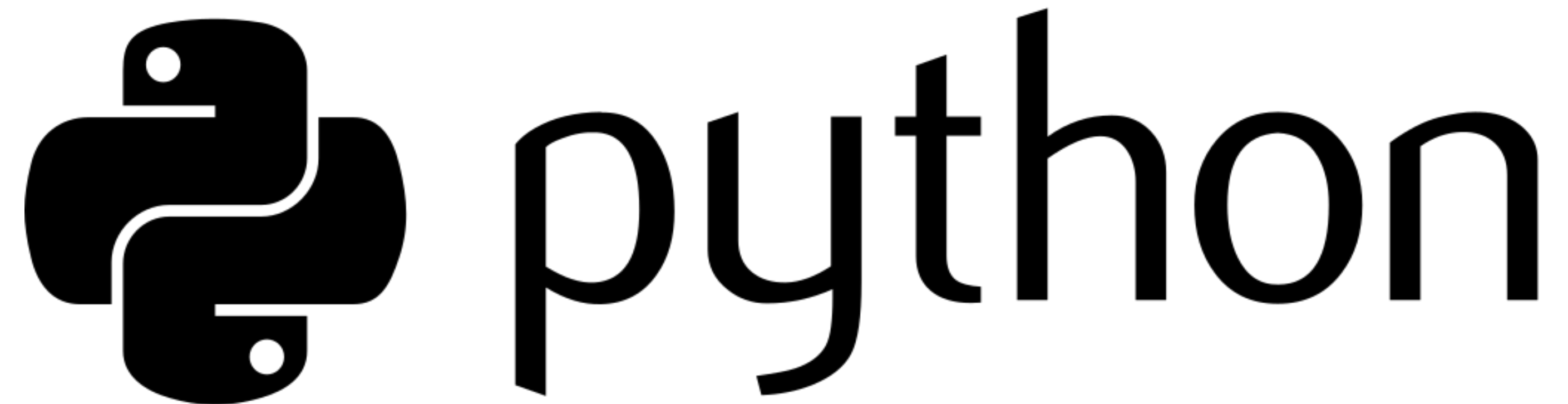
GEOSCAN

Программирование на Python





https://github.com/bas36/first_code



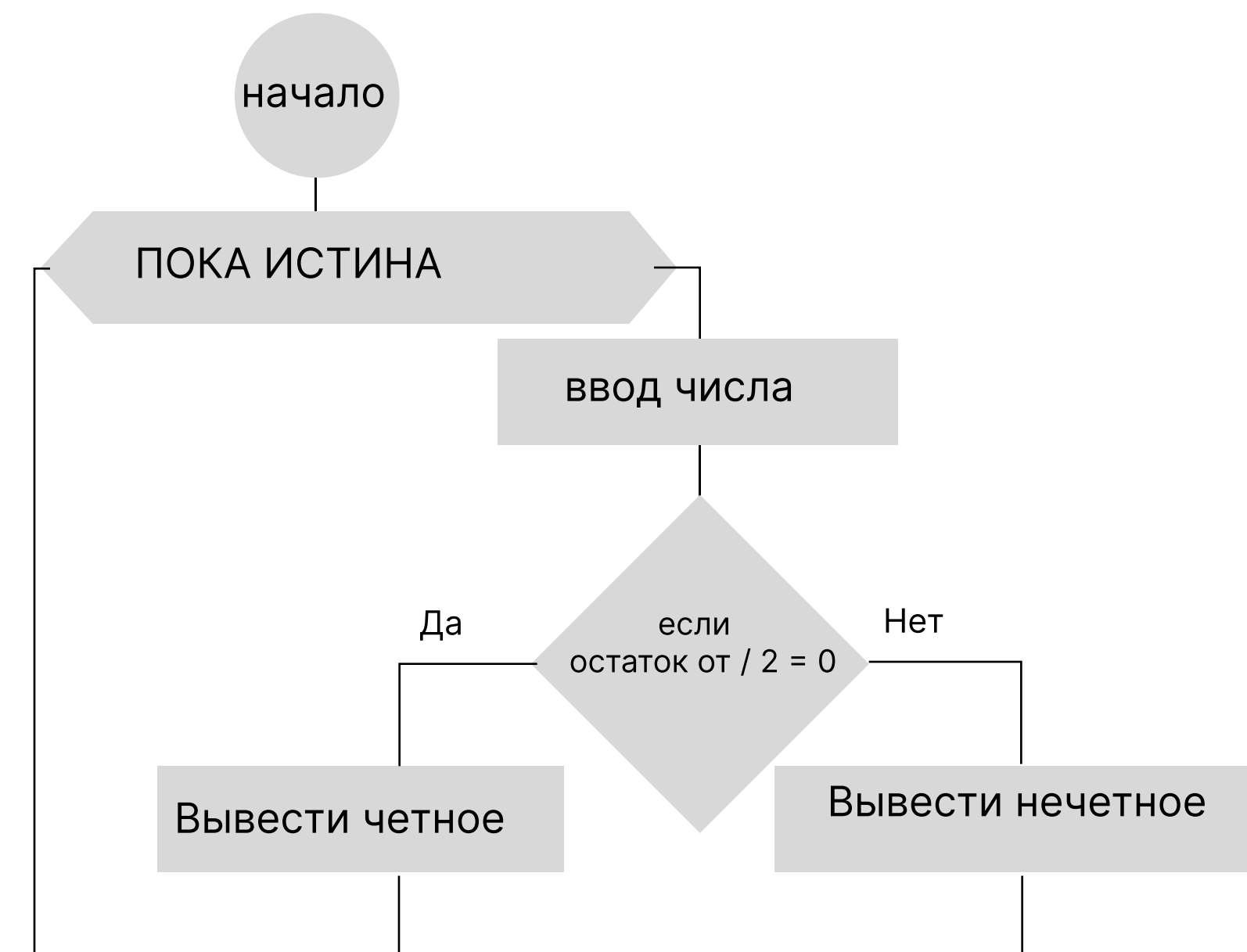
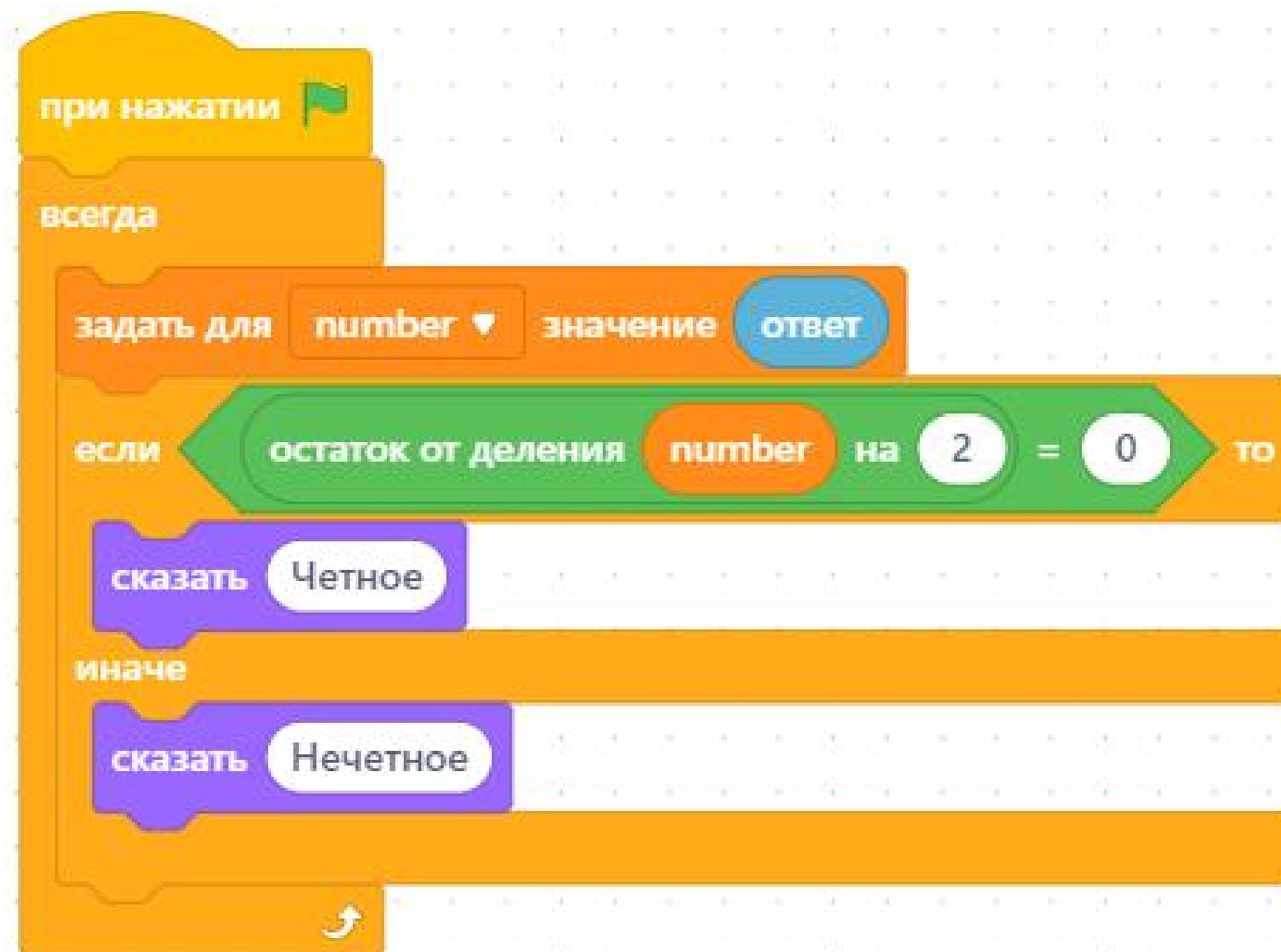
<https://chat.deepseek.com/>

clck.ru/3G2Mpq

```

while True:
    number = int(input("Введите число: "))
    if number % 2 == 0:
        print("Четное")
    else:
        print("Нечетное")

```



🔗 Геоскан `pioneer_sdk`

Geoscan PioneerSDK — это набор инструментов для взаимодействия с дронами Geoscan Pioneer. Он предоставляет краткий и удобный API для таких функций дронов, как автономный полёт, получение кадров с камеры дрона и многое другое. Пожалуйста, ознакомьтесь с `examples/` каталогом, чтобы найти отправные точки и вдохновение.

```
from pioneer_sdk import Pioneer  
pioneer_mini = Pioneer(logger=False)  
while True:  
    number = int(input("Введите число: "))  
    if number % 2 == 0:  
        print("Четное")  
        pioneer_mini.led_control(r=0, g=255, b=0)  
    else:  
        print("Нечетное")  
        pioneer_mini.led_control(r=255, g=0, b=0)
```


Что понадобится:

Все пункты хорошо освещены на сайте [Геоскан](#)

1. Установка python
2. Установка pycharm
3. Установка opencv
3. Установка pioner_sdk
4. Проверка прошивки квадрокоптера
5. Обновление параметров автопилота
6. Прошивка контроллера esp32

ГЕОСКАН ПИОНЕР
ДОКУМЕНТАЦИЯ_v 3.2

Проект "Пионер" Страница продукта

Быстрый поиск

РУКОВОДСТВА ПОЛЬЗОВАТЕЛЯ:

- Пионер Базовый >
- Пионер Мини >
- Пионер Макс >
- Пионер FPV >
- Пульт радиоуправления >

МОДУЛИ И НАВИГАЦИЯ:

- Дополнительные модули >
- Система ИК навигации
- Система УЗ навигации
- Система оптической навигации

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ:


ПИОНЕР
МИНИ



СНЯТ С ПРОДАЖ


Настройка Pioneer 🔗

- 1. Для работы с Pioneer Mini в первую очередь необходимо проверить актуальность его прошивки автопилота, ESP32 и параметров автопилота. Полный список необходимого ПО с инструкциями описан в начале страницы.
- 2. Следующим шагом включаем Pioneer Mini и подключаемся к нему по Wi-Fi. Имя сети каждого коптера уникально, но пароль у всех одинаковый: «12345678».


 PioneerMiniac67b20d4320
Защищено

☐ Подключаться автоматически

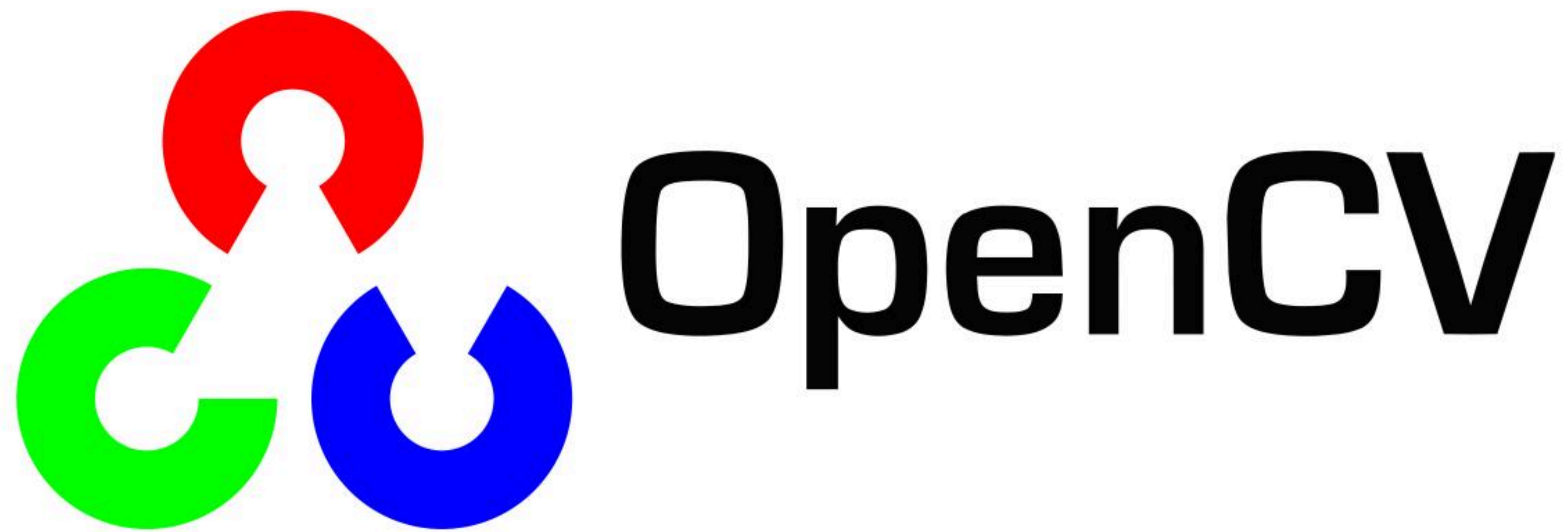
Подключиться

 PioneerMiniac67b20d4320
Защищено

Введите ключ безопасности сети

12345678 

ДалееОтмена



Веб камера ноутбука

```
import cv2

cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    cv2.imshow('Camera', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Камера квадрокоптера

```
from pioneer_sdk import Camera
import cv2

camera = Camera()
while True:
    frame = camera.get_cv_frame()
    cv2.imshow("video", frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cv2.destroyAllWindows()
```

С чем мы работаем?

```
import cv2
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    cv2.imshow('Camera', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
print(type(frame))
cap.release()
cv2.destroyAllWindows()

<class 'numpy.ndarray'>
```

Что такое numpy.ndarray?

Это многомерный массив, предоставляемый библиотекой NumPy, используемый для хранения числовых данных.

Он отличается от обычных списков Python тем, что поддерживает сложные математические операции, более эффективен по памяти и быстрее при вычислениях.

С чем мы работаем?

0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4
0	18	146	250	255	247	255	255	255	249	255	240	255	129	0	5
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0

```
import cv2
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    cv2.imshow('Camera', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
print(frame.shape)
cap.release()
cv2.destroyAllWindows()
```

(480, 640, 3)



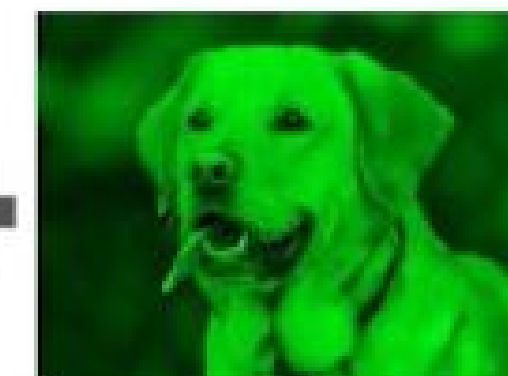
Colour Image

=



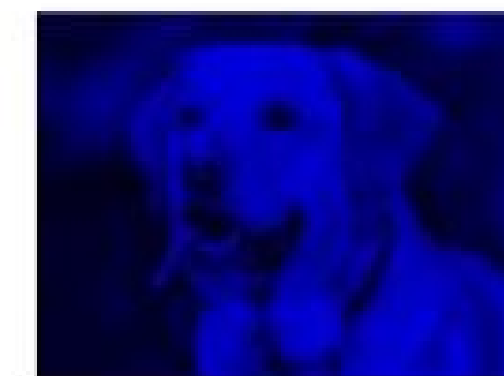
Red

+

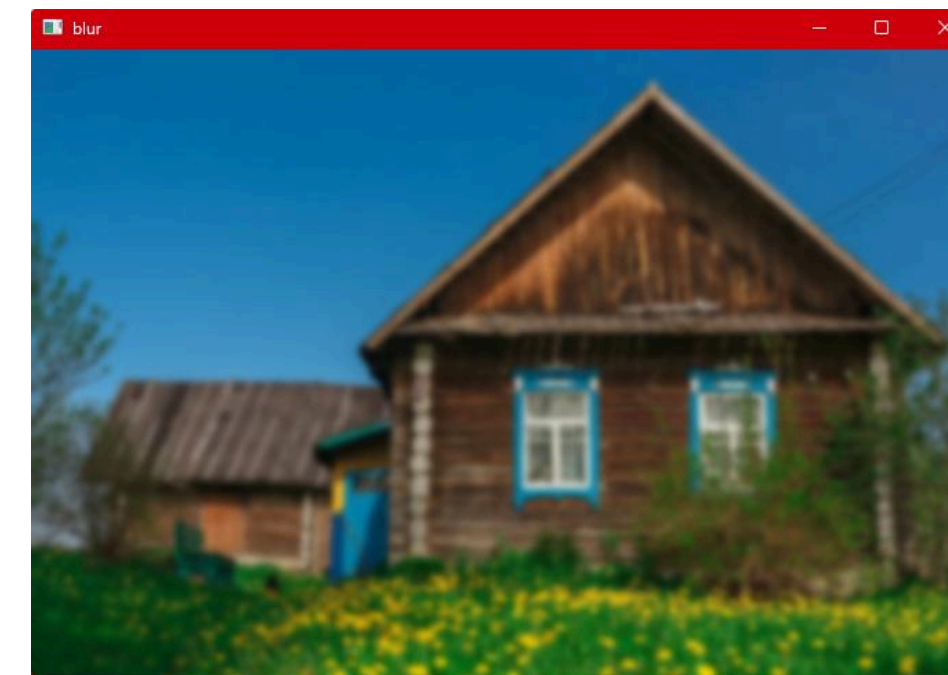
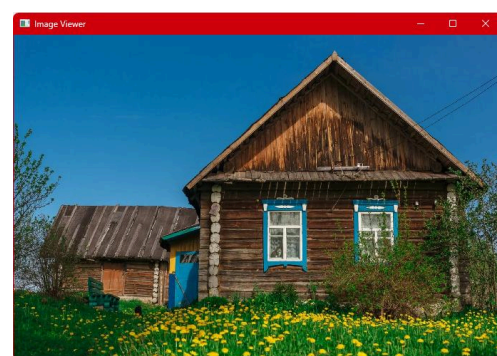


Green

+

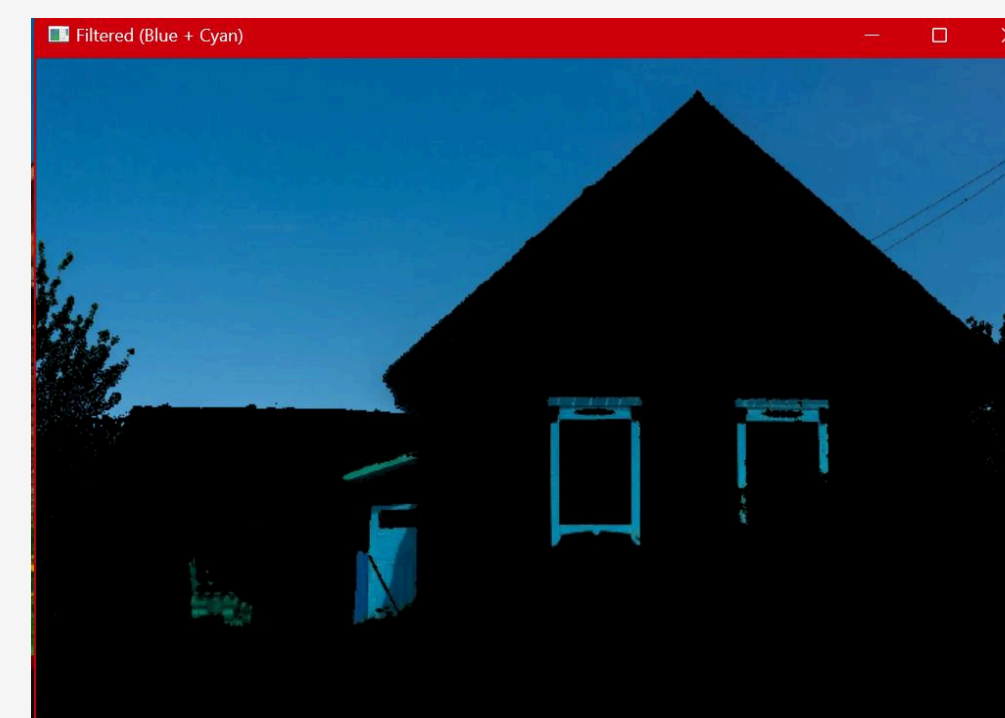


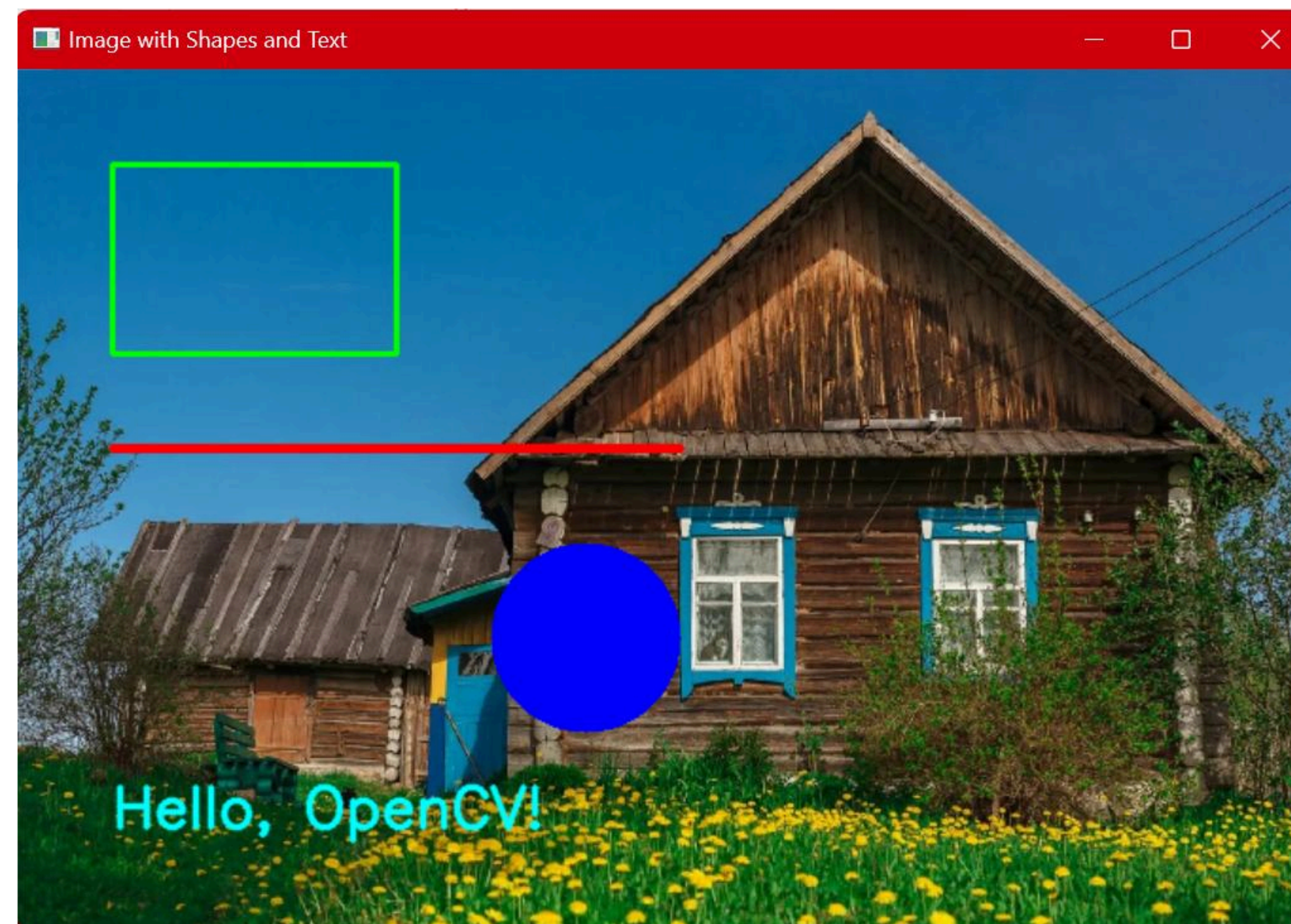
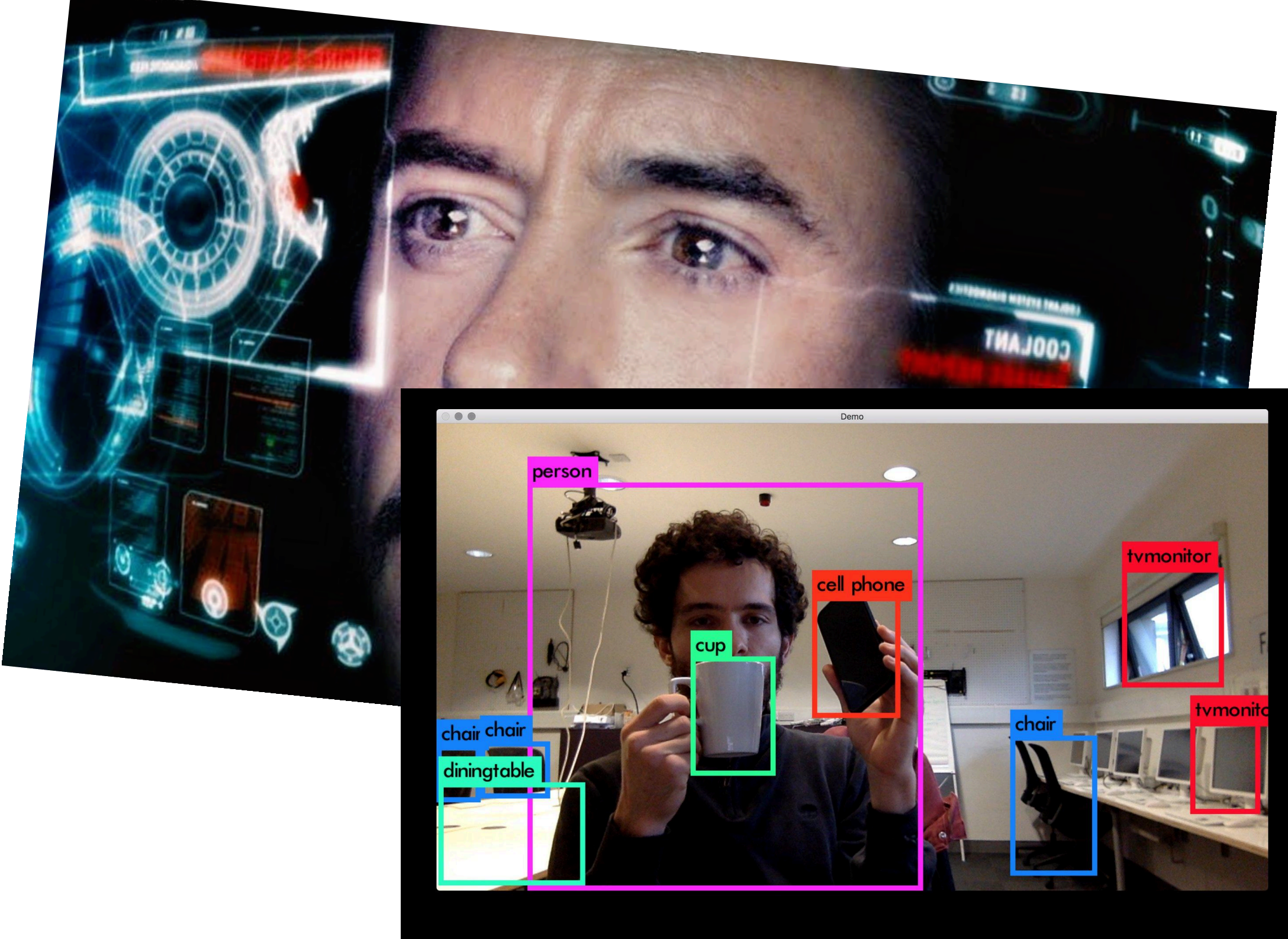
Blue



Преобразования кадров

```
# Поворот изображения на 90 градусов
rotated_frame = cv2.rotate(frame, cv2.ROTATE_90_CLOCKWISE)
# Переворот изображения (flip)
flipped_frame = cv2.flip(frame, 1) # 1 - горизонтальный переверт -1 горизонтально и вертикально 0 - вертикальный
# Изменение размера изображения (resize)
resized_frame = cv2.resize(flipped_frame, (320, 240)) # Новый размер: 320x240
# Преобразование в оттенки серого
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
# Размытие
blur = cv2.GaussianBlur(frame, (15, 15), 0)
# Инверсия цветов
bitwise = cv2.bitwise_not(frame)
```

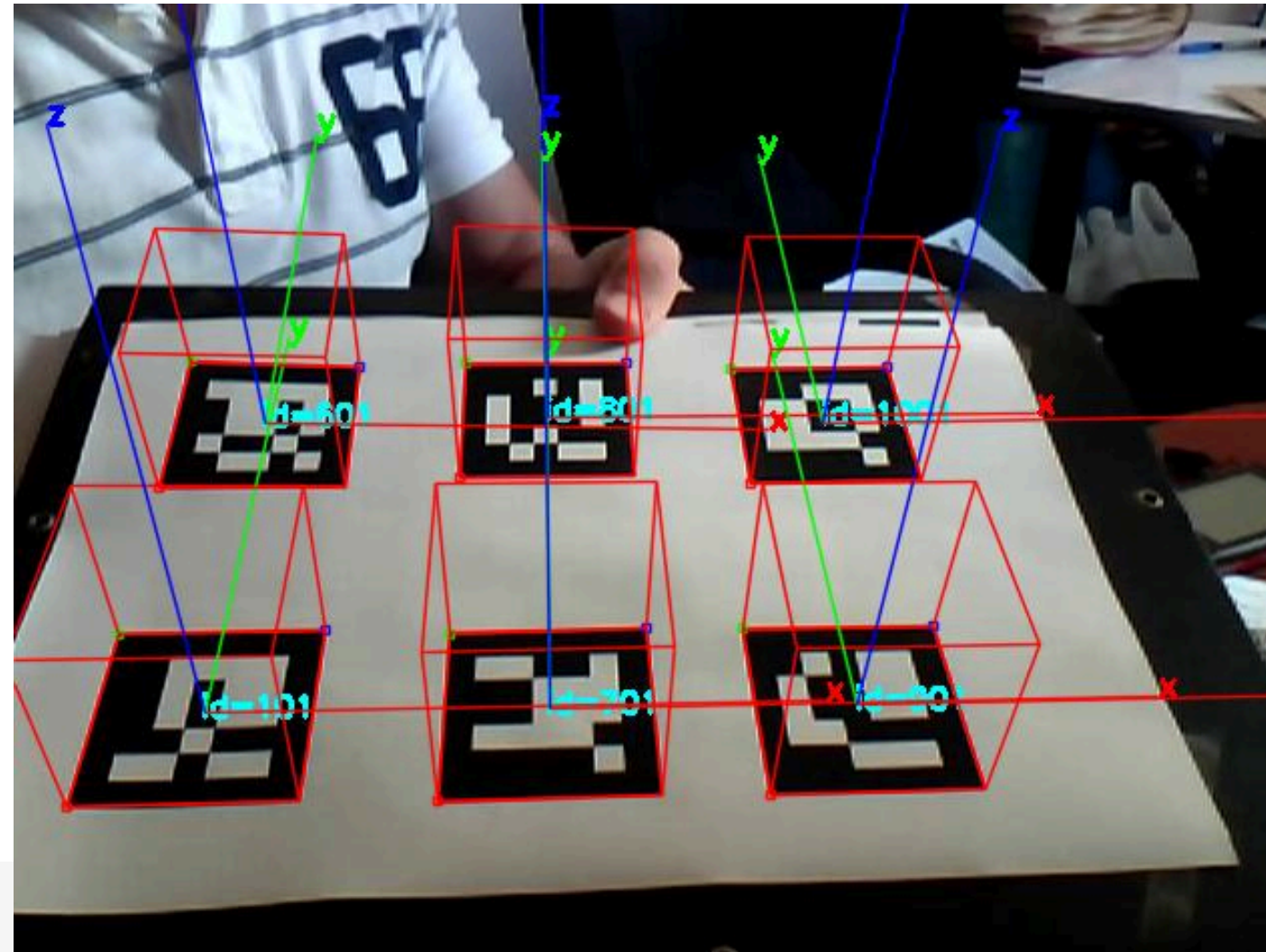




Рисование на изображении

```
cv2.rectangle(frame, (50, 50), (200, 200), (0, 255, 0), 3) # Прямоугольник
cv2.circle(frame, (150, 150), 50, (255, 0, 0), 3) # Круг
cv2.line(frame, (0, 0), (320, 240), (0, 0, 255), 2) # Линия
cv2.putText(frame, text, (text_x, text_y), font, font_scale, font_color, thickness, line_type) # Текст
cv2.polylines(resized_frame, [points], isClosed=True, color=(0, 255, 0), thickness=3) # Многоугольник
```


ARUCO



```
import cv2
import numpy as np

dictionary = cv2.aruco.getPredefinedDictionary(cv2.aruco.DICT_4X4_250)
parameters = cv2.aruco.DetectorParameters()
detector = cv2.aruco.ArucoDetector(dictionary, parameters)

cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    corners, ids, _ = cv2.aruco.detectMarkers(gray, dictionary, parameters=parameters)
    cv2.aruco.drawDetectedMarkers(frame, corners, ids)
    cv2.imshow("ArUco Detection", frame)
    if cv2.waitKey(1) & 0xFF == ord("q"):
        break
cap.release()
cv2.destroyAllWindows()
```

ArUco markers generator!

Dictionary: 4x4 (50, 100, 250, 1000) ▾

Marker ID: 0

Marker size, mm: 100

