

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**«Нахождение и обработка контуров»**

**ОТЧЕТ**  
**по лабораторной работе №13**  
**дисциплины**  
**«Технологии распознавания образов»**

Выполнила:

Кувшин Ирина Анатольевна  
2 курс, группа ПИЖ-б-о-21-1,  
010.03.04 «Программная инженерия»,  
направленность (профиль) «Разработка  
и сопровождение программного  
обеспечения», очная форма обучения

---

(подпись)

Проверил:

---

(подпись)

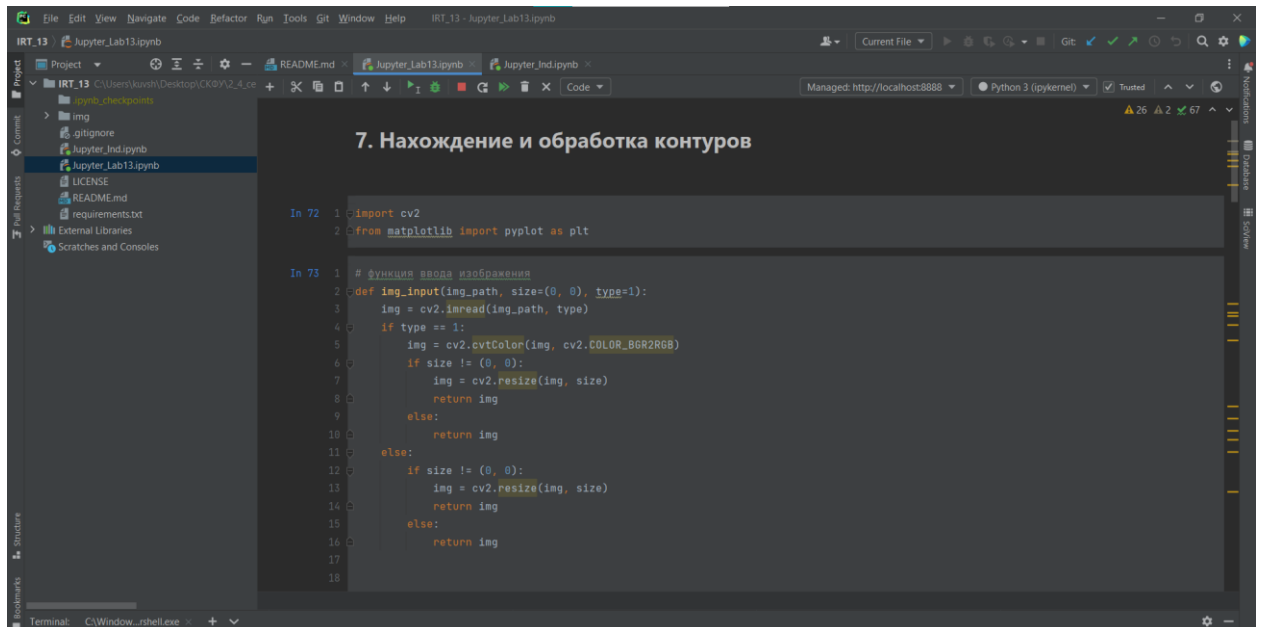
Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2023 г.

## Цель работы:

Обнаружение и выделение контуров на изображении, анализ контуров.  
Изучение функций `cv2.findContours()`, `cv2.drawContours()`.

## Ход выполнения:

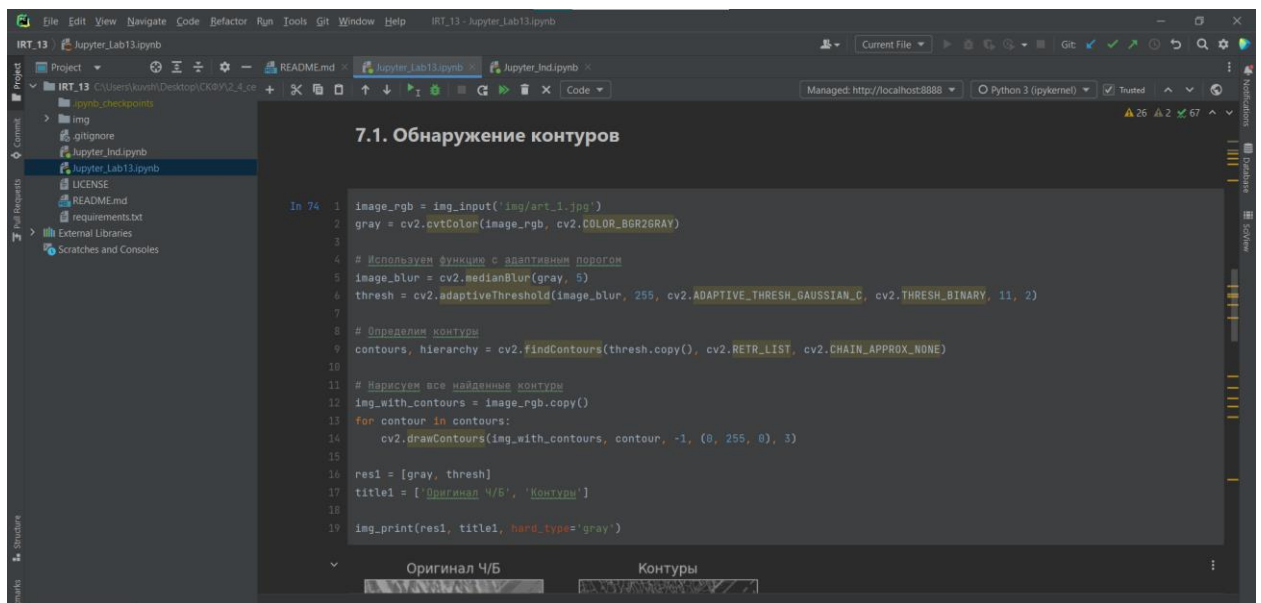


The screenshot shows the JupyterLab interface with a file explorer on the left and a code editor in the center. The file explorer shows a project named 'IRT\_13' with subfolders 'img', 'jupyter\_lab13.ipynb', 'LICENSE', 'README.md', 'requirements.txt', and 'External Libraries'. The code editor displays the following code:

```
7. Нахождение и обработка контуров

In 72: 1 import cv2
        2 from matplotlib import pyplot as plt

In 73: 1 # функция ввода изображения
        2 def img_input(img_path, size=(0, 0), type=1):
        3     img = cv2.imread(img_path, type)
        4     if type == 1:
        5         img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        6         if size != (0, 0):
        7             img = cv2.resize(img, size)
        8         return img
        9     else:
        10        return img
        11
        12     else:
        13         if size != (0, 0):
        14             img = cv2.resize(img, size)
        15             return img
        16         else:
        17             return img
        18
```



The screenshot shows the JupyterLab interface with the same file explorer and code editor. The code editor displays the following code:

```
7.1. Обнаружение контуров

In 74: 1 image_rgb = img_input('img/art_1.jpg')
        2 gray = cv2.cvtColor(image_rgb, cv2.COLOR_BGR2GRAY)
        3
        4 # Используем функцию с адаптивным порогом
        5 image_blur = cv2.medianBlur(gray, 5)
        6 thresh = cv2.adaptiveThreshold(image_blur, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 11, 2)
        7
        8 # Определим контуры
        9 contours, hierarchy = cv2.findContours(thresh.copy(), cv2.RETR_LIST, cv2.CHAIN_APPROX_NONE)
        10
        11 # Нарисуем все найденные контуры
        12 img_with_contours = image_rgb.copy()
        13 for contour in contours:
        14     cv2.drawContours(img_with_contours, contour, -1, (0, 255, 0), 3)
        15
        16 res1 = [gray, thresh]
        17 title = ['Оригинал Ч/Б', 'Контур']
        18
        19 img_print(res1, title, hard_type='gray')
```

Below the code editor, there are two small images side-by-side. The first image is labeled 'Оригинал Ч/Б' (Original Grayscale) and the second image is labeled 'Контур' (Contour). The 'Контур' image shows the detected contours of the original grayscale image.

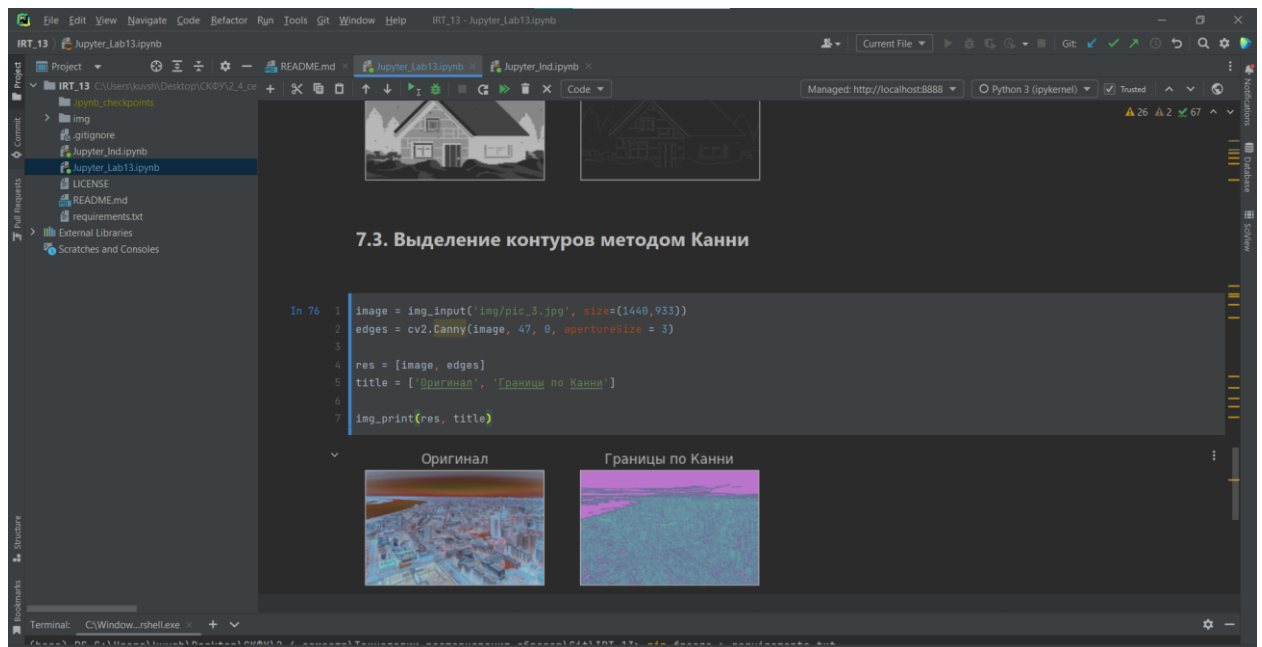
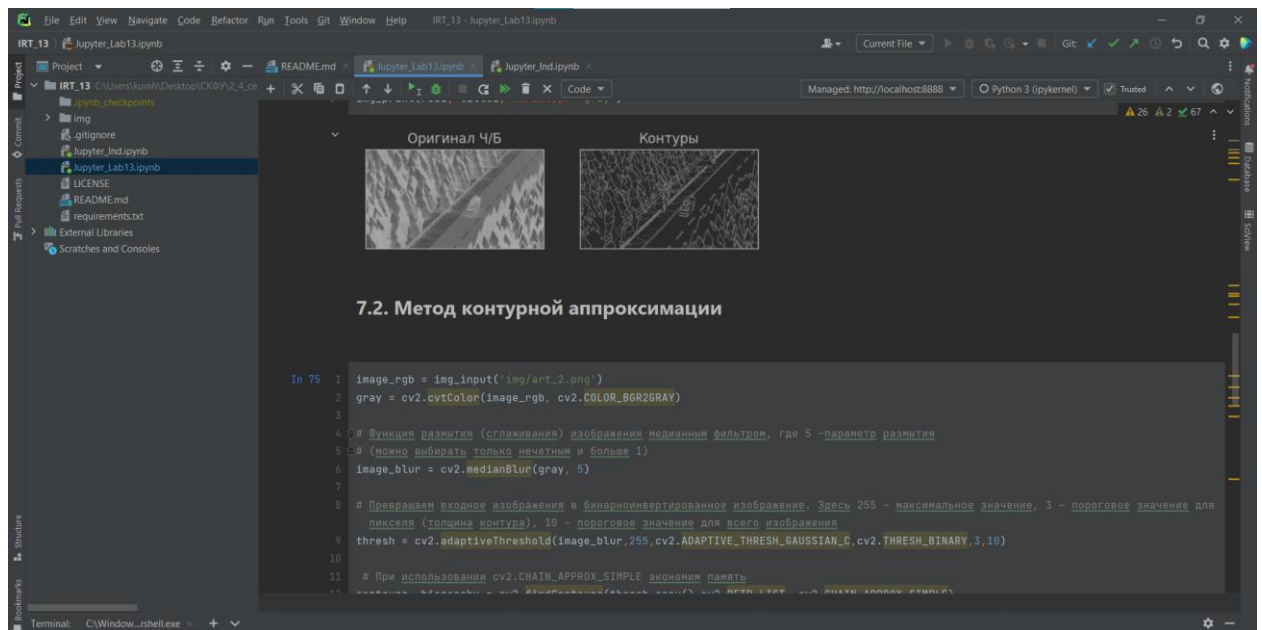
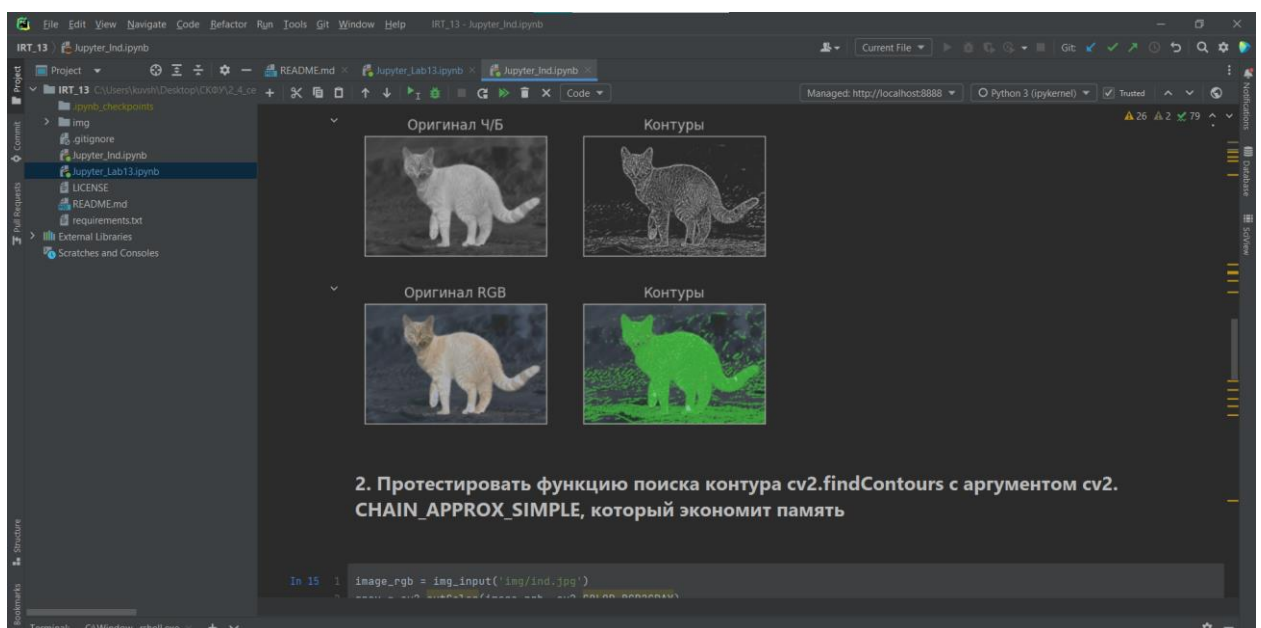
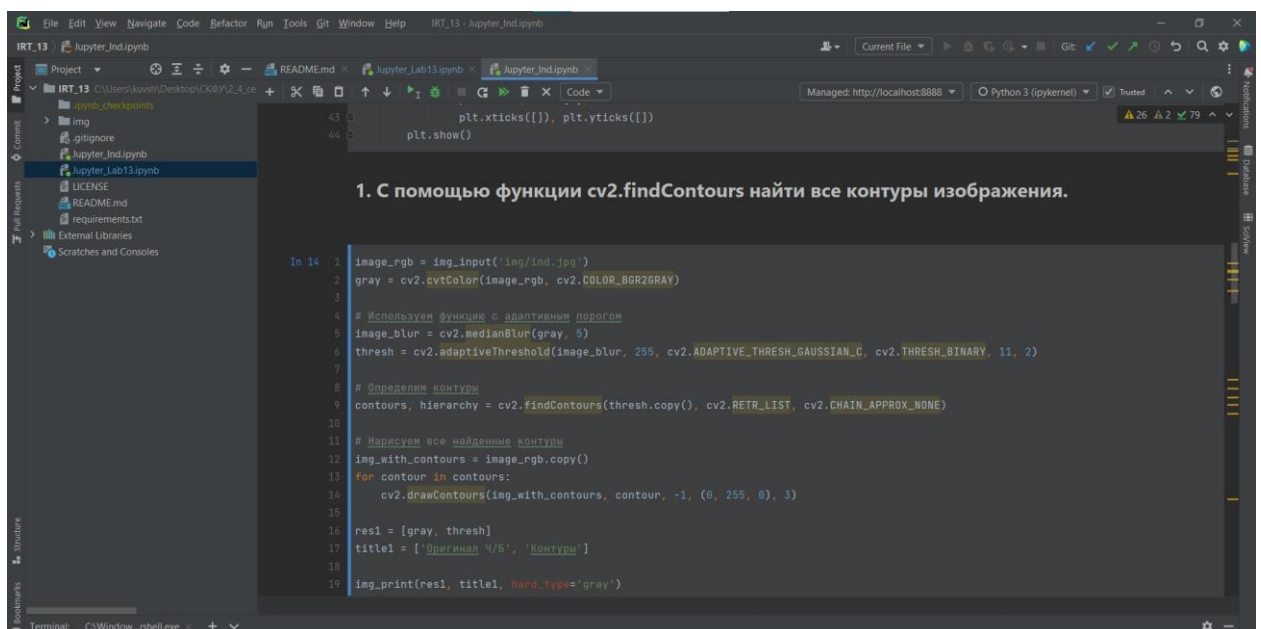
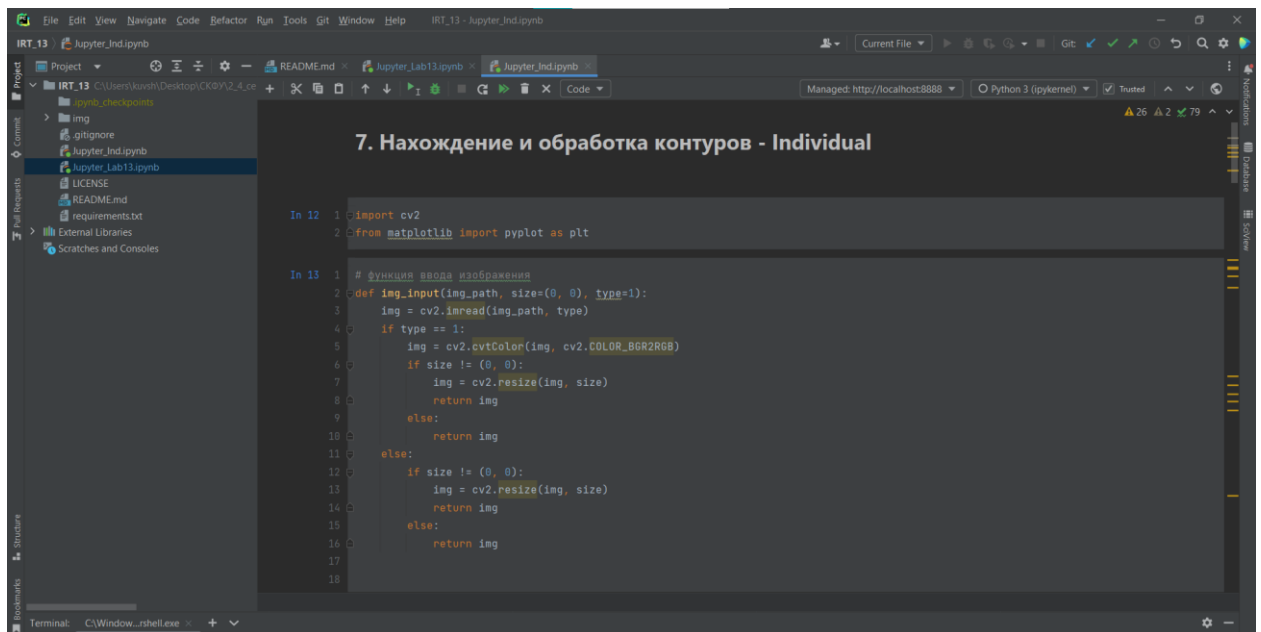


Рисунок 13.1– Код программы

## Индивидуальное задание



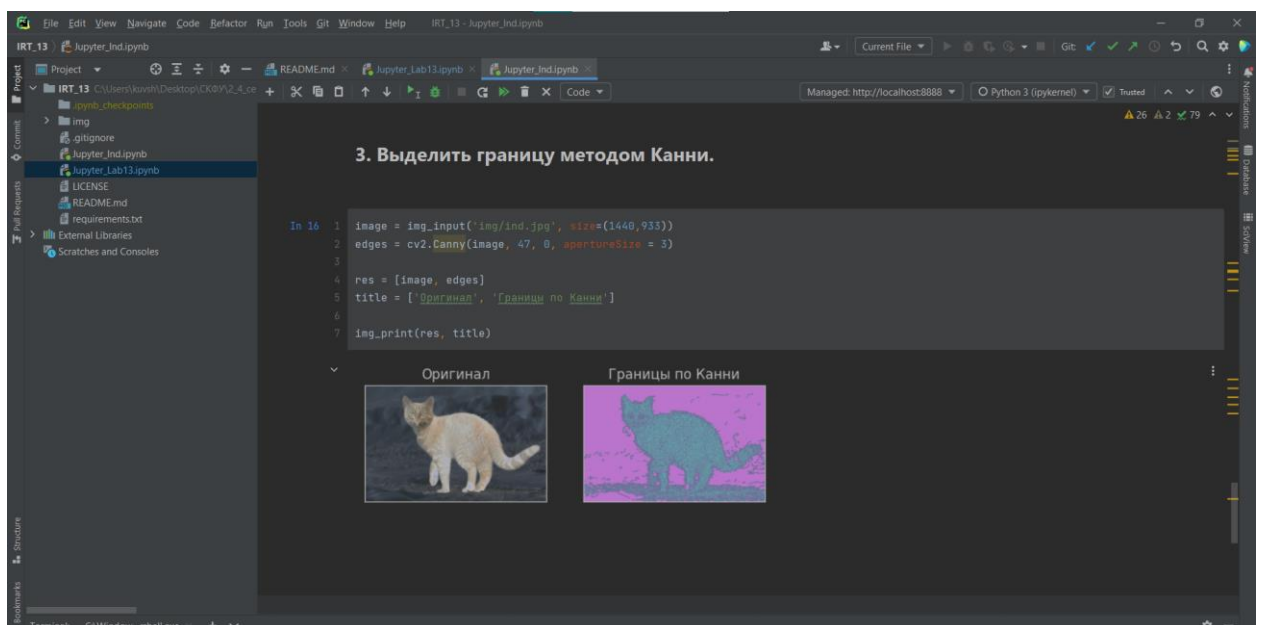
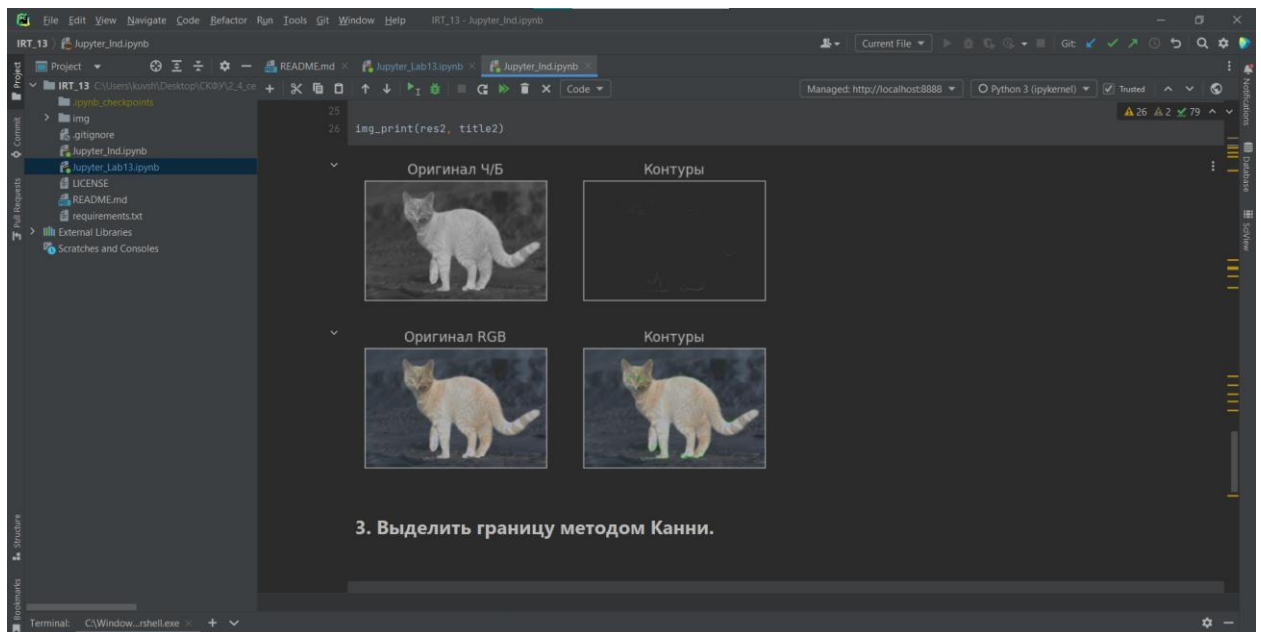


Рисунок 13.2— Код программы

### Контрольные вопросы:

1. Что такое дискретный путь бинарного изображения между двумя пикселями?

Дискретный путь бинарного изображения между двумя пикселями — это последовательность пикселей, в которой при перемещении от начального пикселя к конечному пикселю каждый пиксель последовательности имеет два соседних пикселя, принадлежащих этой последовательности: предшествующий и последующий.

2. Что такое контур бинарного изображения?

Контур бинарного изображения – это любой замкнутый путь. В случае бинарного изображения контур – это граница связанной области с одинаковой интенсивностью.

3. Для чего используют контура?

Контура используют для анализа формы бинарного изображения, с их помощью проводят обнаружение и распознавание объектов. Контуры удобно выделять в бинарном изображении, поэтому перед поиском контуров применяют пороговую обработку.

4. Что представляет из себя каждый отдельный контур в функции `cv2.findContours()`?

Каждый отдельный контур – это массив Numpy с координатами (x, y) граничных точек объекта.

5. Что выводит параметр `hierarchy` в функции `cv2.findContours()`?

Параметр `hierarchy` выводит иерархию контуров.

6. На чем основывается метод Канни для выделения контуров?

Метод Канни обнаруживает границы связанной области изображения, выполняя поиск локальных максимумов градиента  $f(x, y)$ . Градиент вычисляется после применения фильтра Гаусса. Метод использует два порога для нахождения сильных и слабых краев. Слабые края включаются в выход, если они связаны с сильными.

7. Какие аргументы содержит функция Канни?

Функция Канни `cv2.Canny(img, T_lower, T_upper, apertureSize = 3)` содержит в скобках следующие аргументы: `img` – входное изображение,

T\_lower: нижний порог, T\_upper: верхний порог, apertureSize – размер апертуры оператора Собеля.

8. Что достаточно задать, если контур включает в себя прямые отрезки?

Если контур включает в себя прямые отрезки, то достаточно задать начальную и конечную координату отрезка.

9. Какую функцию необходимо использовать, чтобы нарисовать все контуры изображения?

Чтобы нарисовать все контуры изображения, нужно использовать функцию `img = cv2.drawContours(img, contours,-1, (0,255,0),3)`.

10. Какую функцию необходимо использовать, чтобы нарисовать конкретный контур изображения?

Чтобы нарисовать конкретный контур, скажем 4-й, добавим:

`cnt = contours[4],`

`img = cv2.drawContours(img, [cnt], 0, (0,255,0), 3).`