

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

Институт цифрового развития
Кафедра инфокоммуникаций

«Основы цифровой обработки изображений в OpenCv»

ОТЧЕТ
по лабораторной работе №7
дисциплины
«Технологии распознавания образов»

Выполнила:
Кувшин Ирина Анатольевна
2 курс, группа ПИЖ-б-о-21-1,
09.03.04 «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:

Изучение типов изображений, способов их формирования. Изучение основных функций OpenCv, применяемых для цифровой обработки изображений.

1. Считывание изображения и вывод его на экран, запись изображения в файл.
2. Вывод свойств изображения и сформированной матрицы на экран.
3. Доступ к изображению для изменения значений цвета пикселей.
4. Создание бинарного изображения и его негатива.
5. Применение библиотеки matplotlib для вывода нескольких изображений в общем окне.
6. Выделение и взятие в рамку определенного региона изображения.
7. Уменьшение размера изображения и вывод матрицы на экран.
8. Знакомство с процессом дискретизации и квантования изображения.
9. Приобретение практических навыков использования этих функций.

Ход выполнения:

1.2. Считывание изображения и вывод его на экран

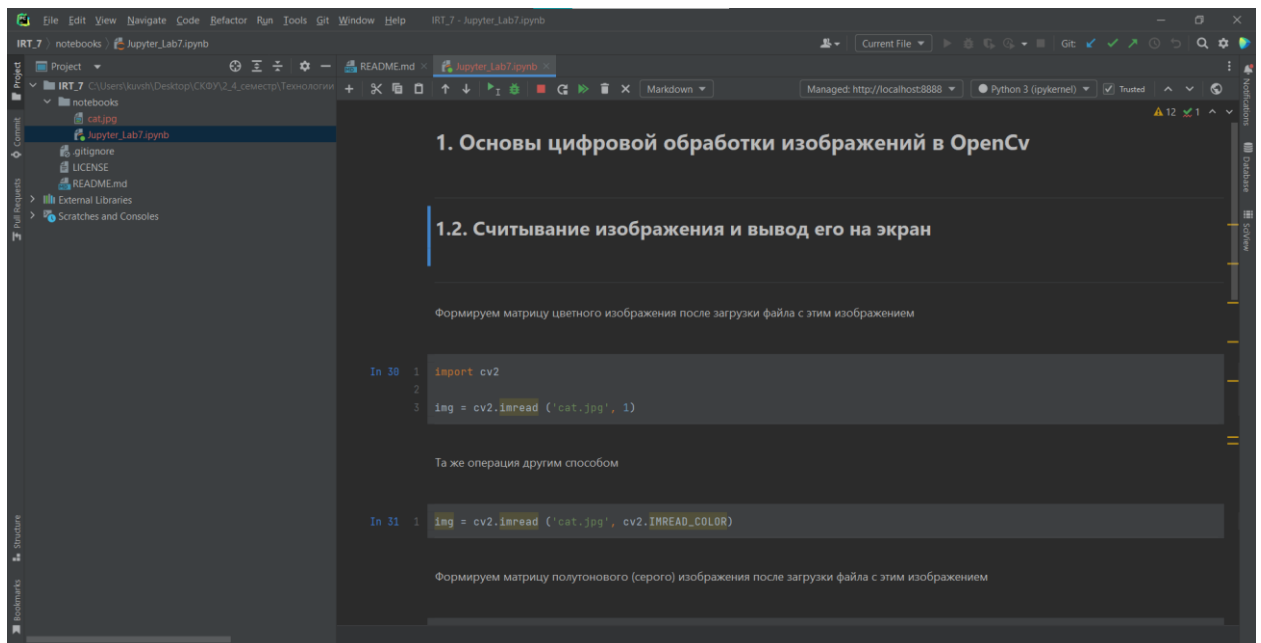


Рисунок 7.1 – Код программы

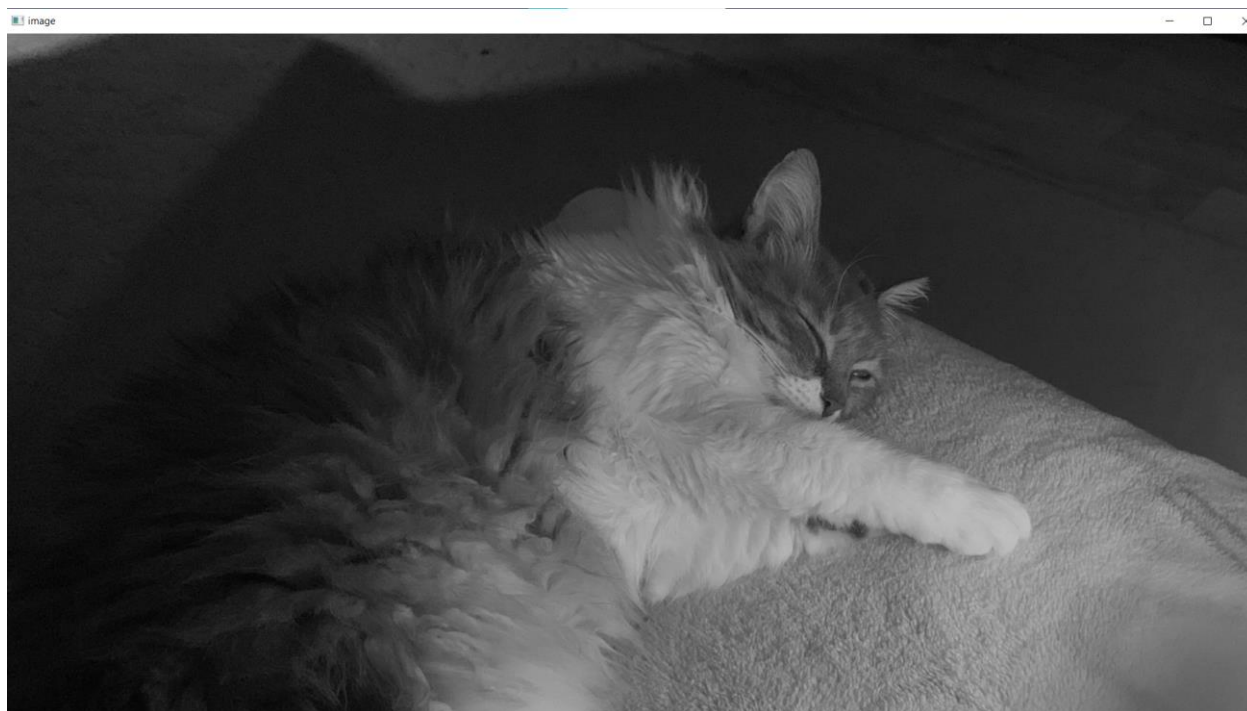


Рисунок 7.2 – Результат выполнения программы

Задание 1.1.

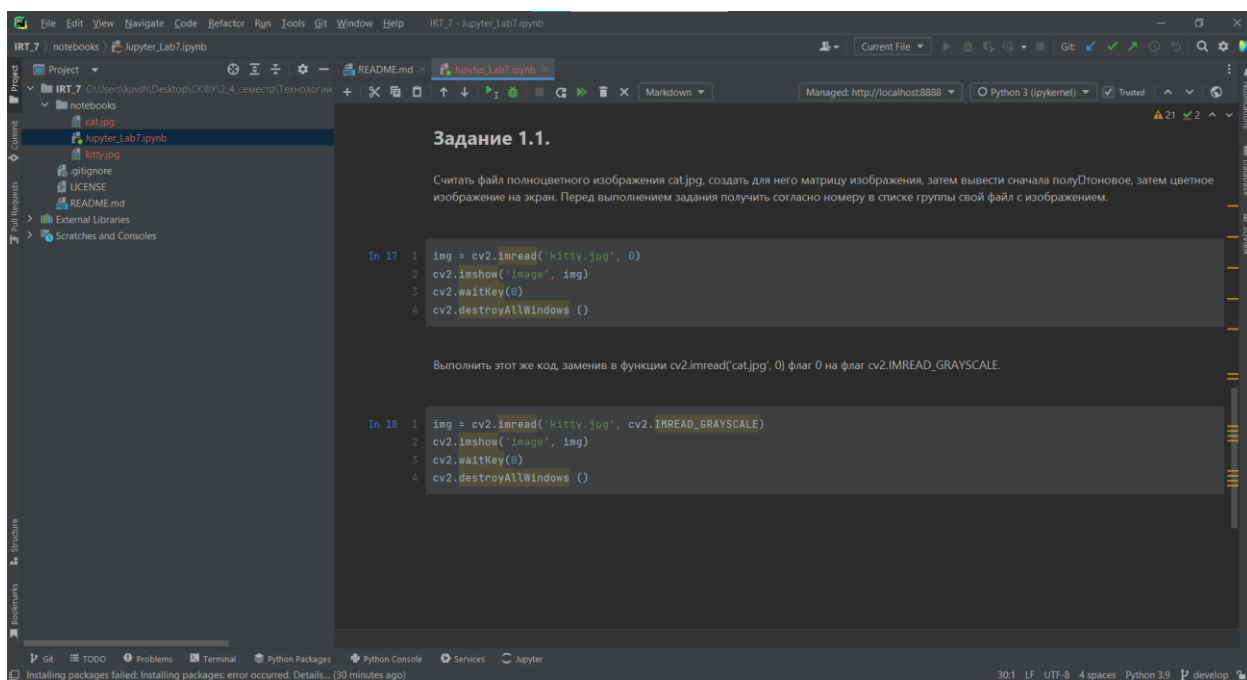


Рисунок 7.3 – Код программы

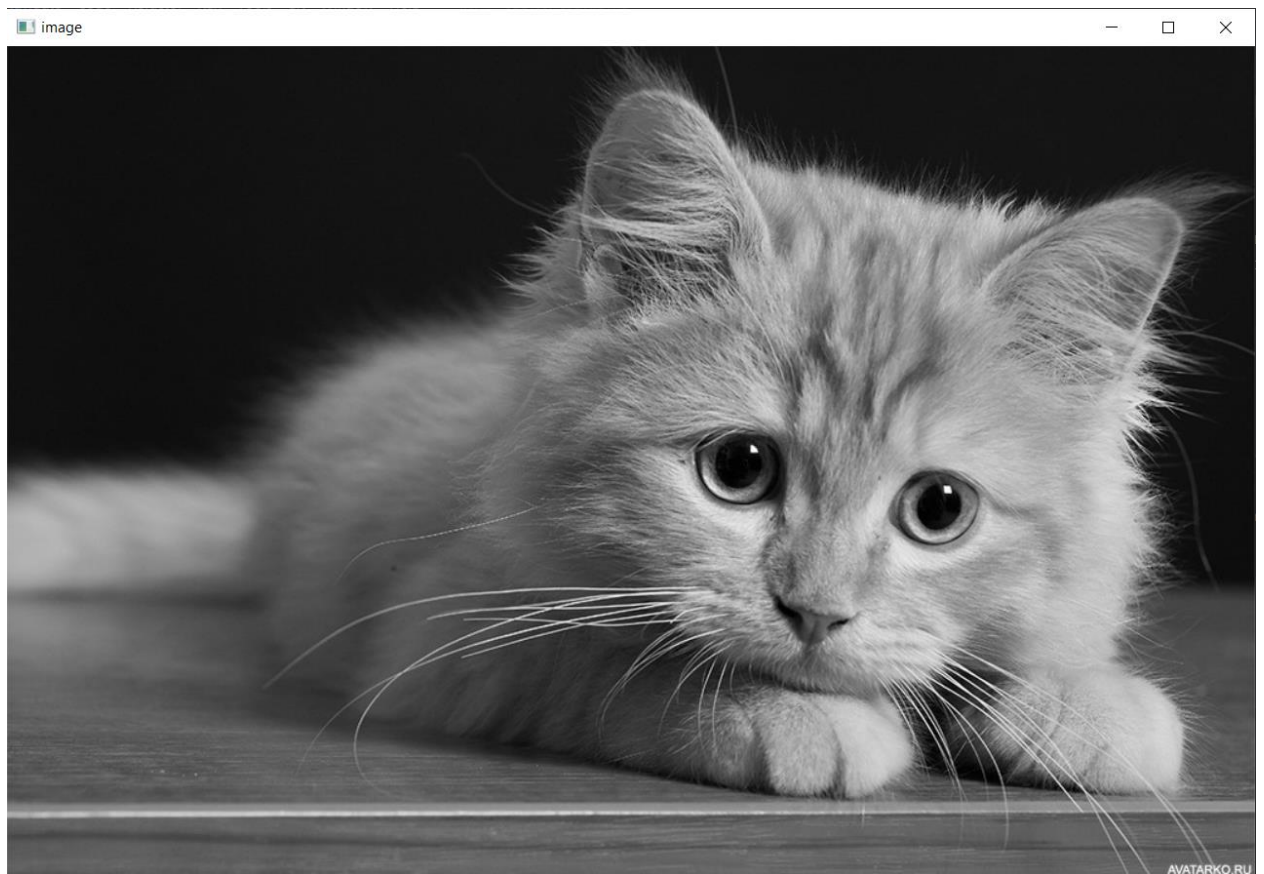


Рисунок 7.4 – Результат выполнения программы

Задание 1.2.

```
In 33: 1 img = cv2.imread('kitty.jpg', 1)
      2 cv2.imshow('image', img)
      3 cv2.waitKey(0)
      4 cv2.destroyAllWindows()

In 34: 1 img = cv2.imread('kitty.jpg', cv2.IMREAD_COLOR)
      2 cv2.imshow('image', img)
      3 cv2.waitKey(0)
      4 cv2.destroyAllWindows()
```

Рисунок 7.5 – Код программы

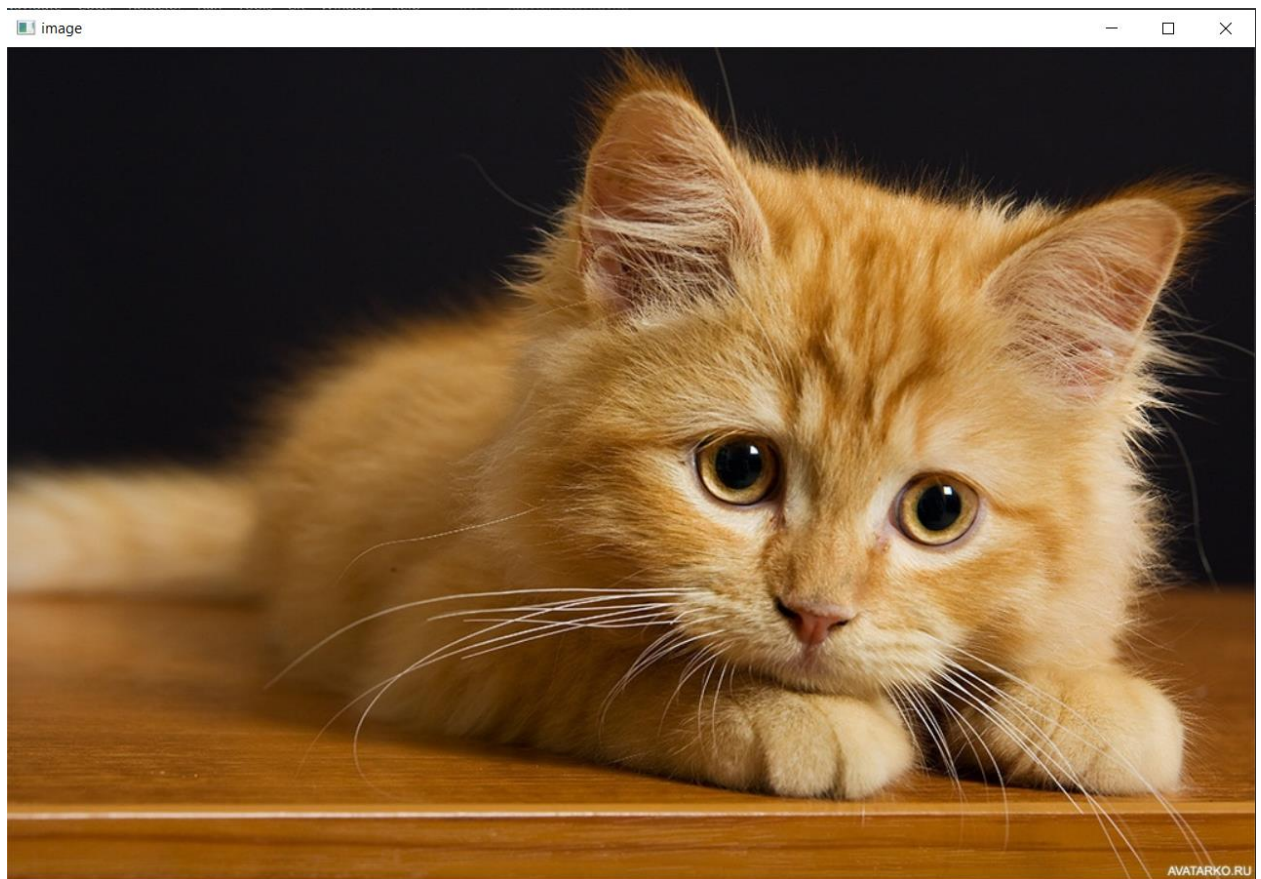


Рисунок 7.6 – Результат выполнения программы

1.3. Запись изображения в файл

Задание 1.3.

```
In 11: 1 img = cv2.imread('kitty.jpg')
      2 cv2.imwrite('img.png', img)
      3 img = cv2.imread('img.png')
      4 cv2.imshow('image', img)
      5 cv2.waitKey(0)
      6 cv2.destroyAllWindows()
```

Рисунок 7.7 – Код программы

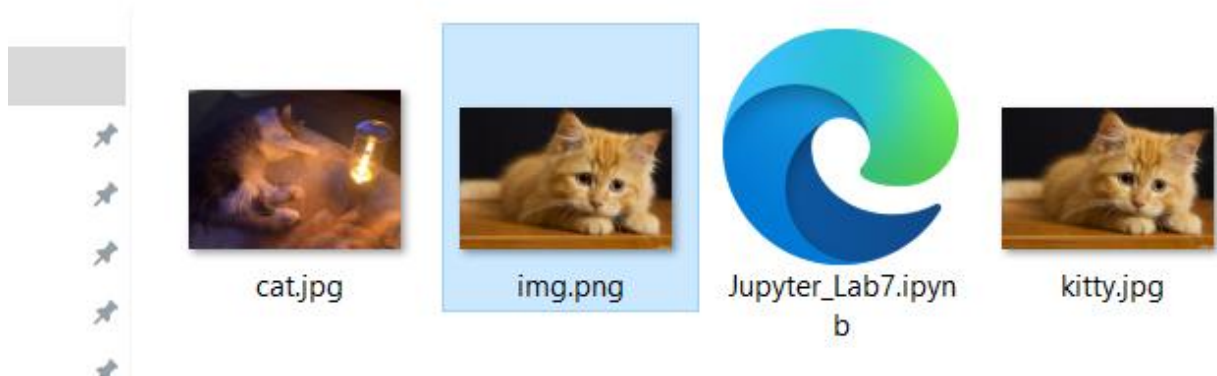


Рисунок 7.8 – Результат выполнения программы

1.4. Вывод сформированной матрицы на экран

Задание 1.4.

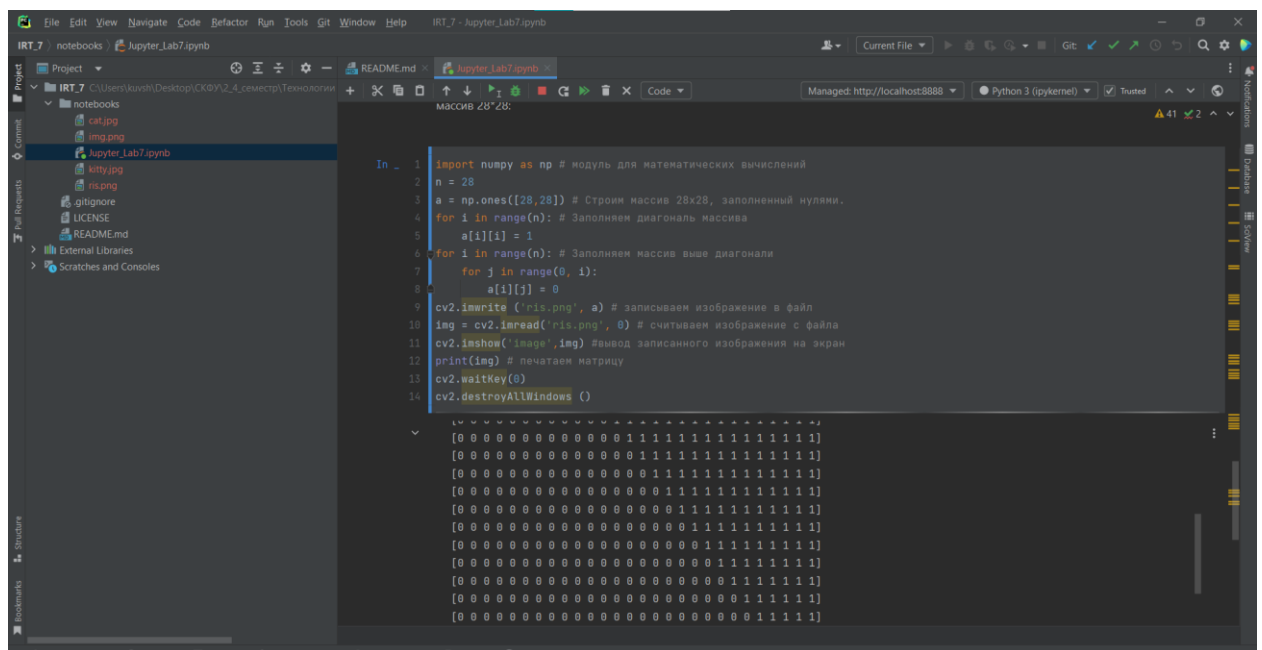


Рисунок 7.9 – Код программы

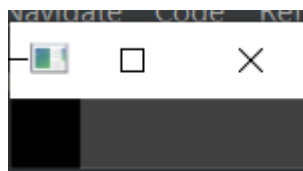


Рисунок 7.10 – Результат выполнения программы

1.5. Вывод основных свойств матрицы изображения на экран

Задание 1.5.

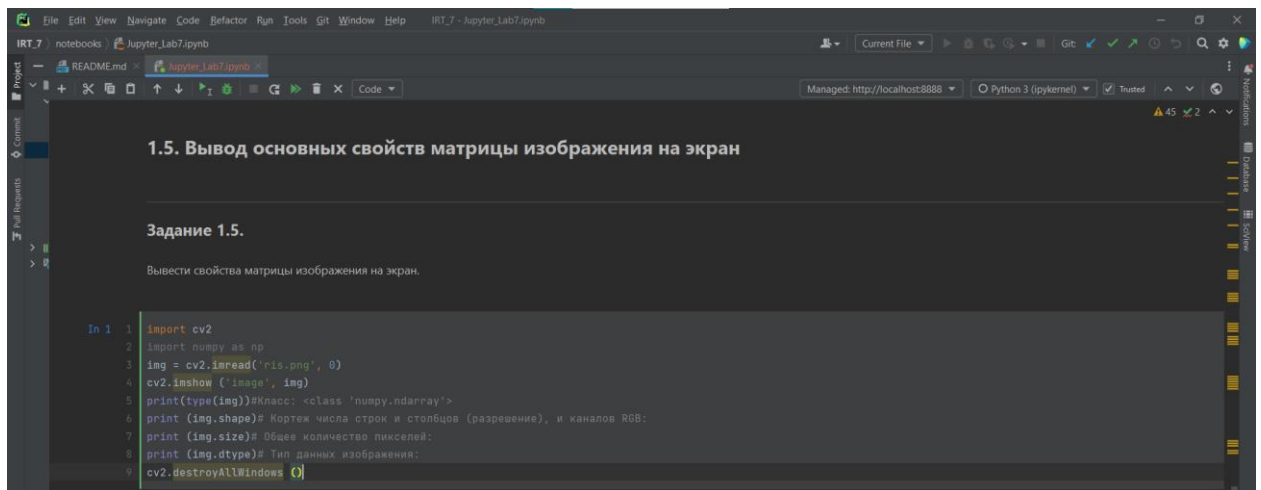


Рисунок 7.11 – Код программы

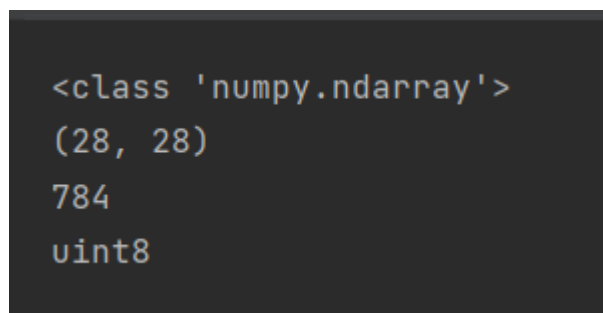


Рисунок 7.12 – Результат выполнения программы

1.6. Доступ к цифровому изображению для изменения значений пикселей

Задание 1.6.

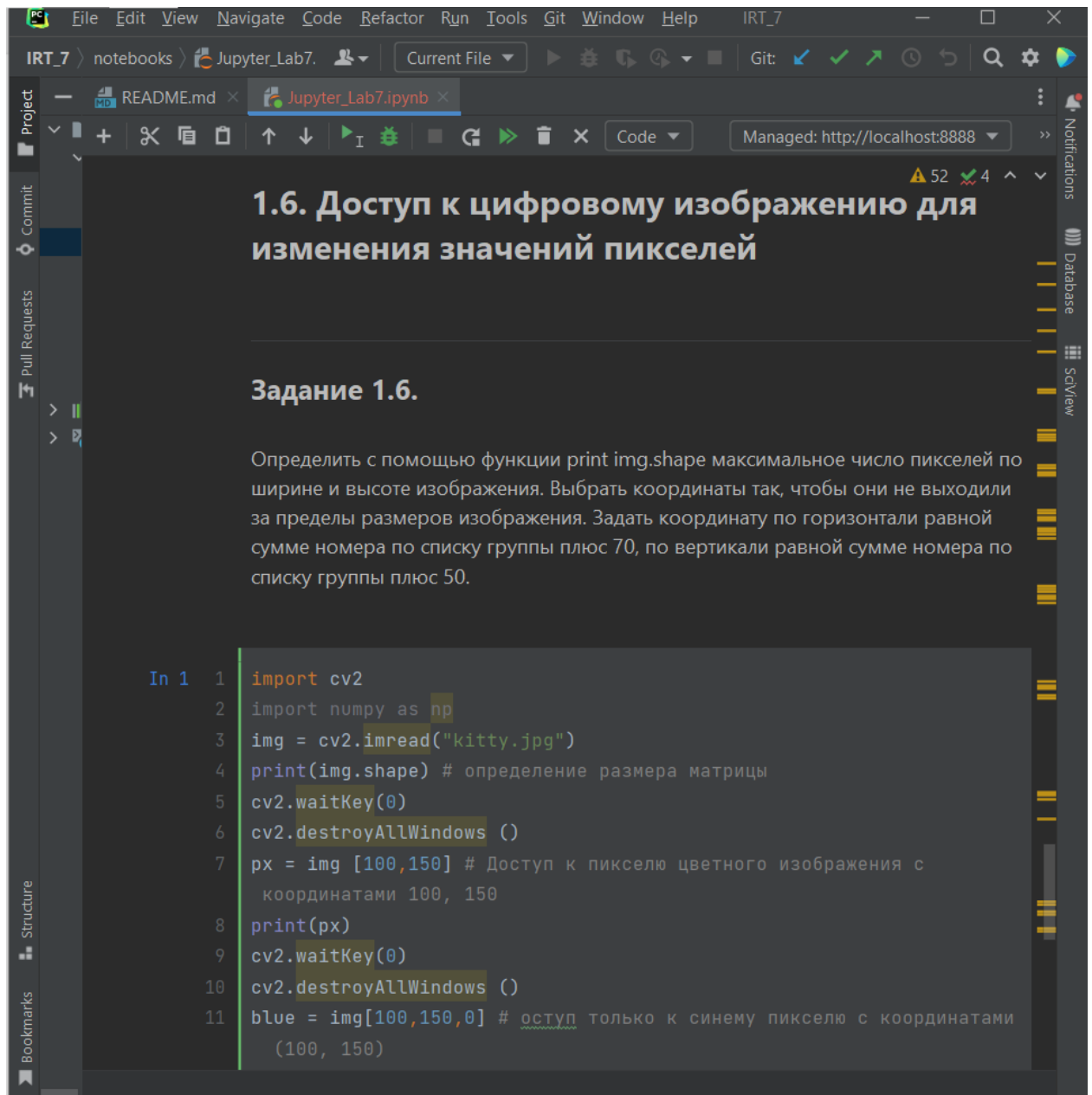


Рисунок 7.13 – Код программы

```
(683, 1024, 3)
[21 16 18]
21
```

Рисунок 7.14 – Результат выполнения программы

Приведенный доступ к значениям каждого пикселя занимает много времени. С помощью функций `array.item()` и `array.itemset()` доступ будет более быстрым, но они работают только для серого изображения. Для получения доступа ко всем значениям B, G, R, нужно вызывать `array.item()`.

```
In 10 1 import cv2
2 import numpy as np
3 img = cv2.imread("kitty.jpg")
4 print(img.shape) # определение размера матрицы
5 cv2.waitKey(0)
6 cv2.destroyAllWindows ()
7 px = img [100,150] # Доступ к пикселю цветного изображения с координатами 100, 150
8 print(px)
9 cv2.waitKey(0)
10 cv2.destroyAllWindows ()
11 #доступ к красному пикселю. Первые две цифры в скобках - координаты пикселя 100, 150 (взять из задания 1.5), третья цифра - флаг красного цвета.
12 red = img.item(100,150,2) # доступ только к красному пикселю с координатами (100, 150)
13 print(red)
14 img.itemset((100,150,2),100)
15 print(img [100, 150])
16
```

Рисунок 7.15 – Код программы

```
(683, 1024, 3)
[21 16 18]
18
[ 21  16 100]
```

Рисунок 7.16 – Результат выполнения программы

1.7. Создание бинарного изображения и его негатива, вывод нескольких изображений в общем окне

Задание 1.7.

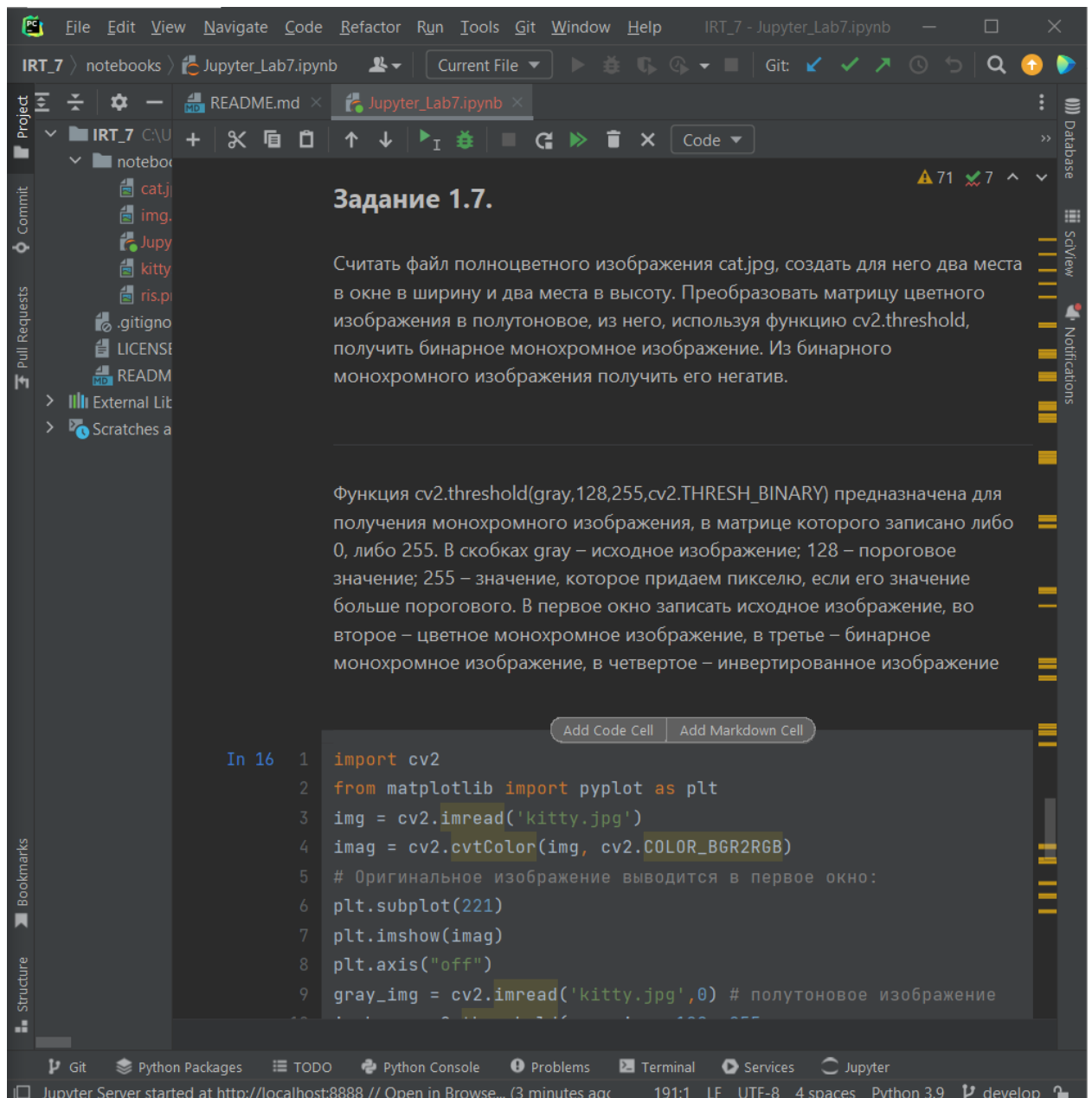


Рисунок 7.17 – Код программы

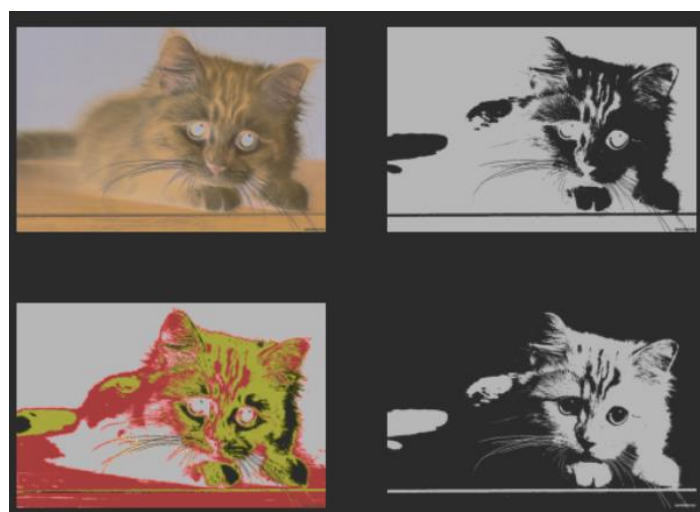


Рисунок 7.18 – Результат выполнения программы

1.8. Выделение и взятие в рамку определенного региона изображения, ROI изображения

Задание 1.8.

Задание 1.8.

На заданном изображении выделить его характерный участок.

```
import cv2
img=cv2.imread('kitty.jpg')
image=cv2.rectangle(img,(280,340),(330,390),(0,0,255), 2)
# Рисуем прямоугольник, (340,330), (470,430) - координаты левого
# верхнего и правого нижнего угла; (0,0,255) - красный цвет, 2 - толщина линии.
cv2.imshow('selected', img)
cv2.waitKey(0)
cv2.destroyAllWindows ()
```

Рисунок 7.19 – Код программы

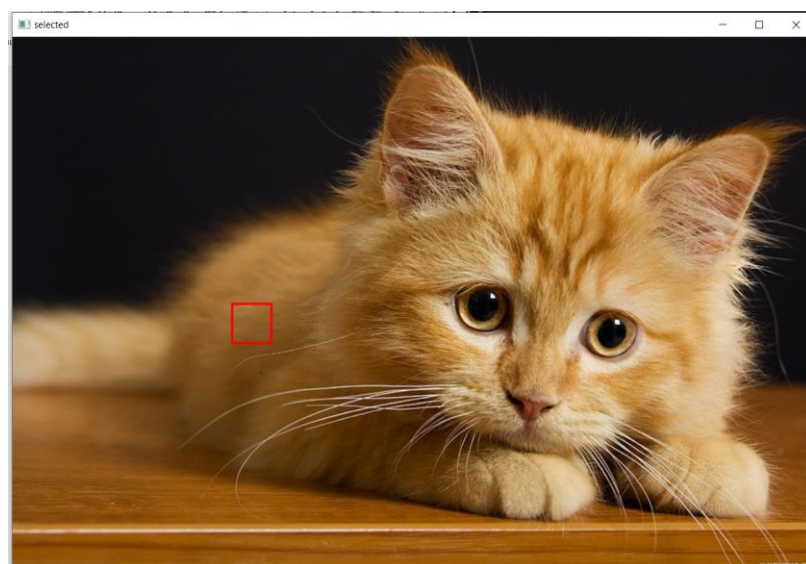


Рисунок 7.20 – Результат выполнения программы

1.9. Уменьшение размера изображения, вывод матрицы изображения на экран после уменьшения ее размеров

Задание 1.9.

Задание 1.9.

Уменьшить заданное изображение и вывести на печать матрицу уменьшенного изображения.

Нам надо сохранить соотношение сторон, чтобы изображение не исказилось при уменьшении. Для этого необходимо вычислить коэффициент уменьшения сторон.

```
In 24 1 import cv2
2 img=cv2.imread('kitty.jpg')
3 final_wide = 200
4 r=float(final_wide)/img.shape[1]
5 dim=(final_wide, int(img.shape[0]*r))#уменьшаем изображение
   до подготовленных размеров
6 resized=cv2.resize(img,dim,interpolation=cv2.INTER_AREA)
7 cv2.imshow("Resize image", resized)
8 cv2.waitKey(0)
9 print(resized.shape)
10 print(resized)
11 img=cv2.imread('kitty.jpg', 0)
12 cv2.imshow('image', img)
13 print(img)
14 cv2.waitKey(0)
15 cv2.destroyAllWindows ()
```

Рисунок 7.21 – Код программы

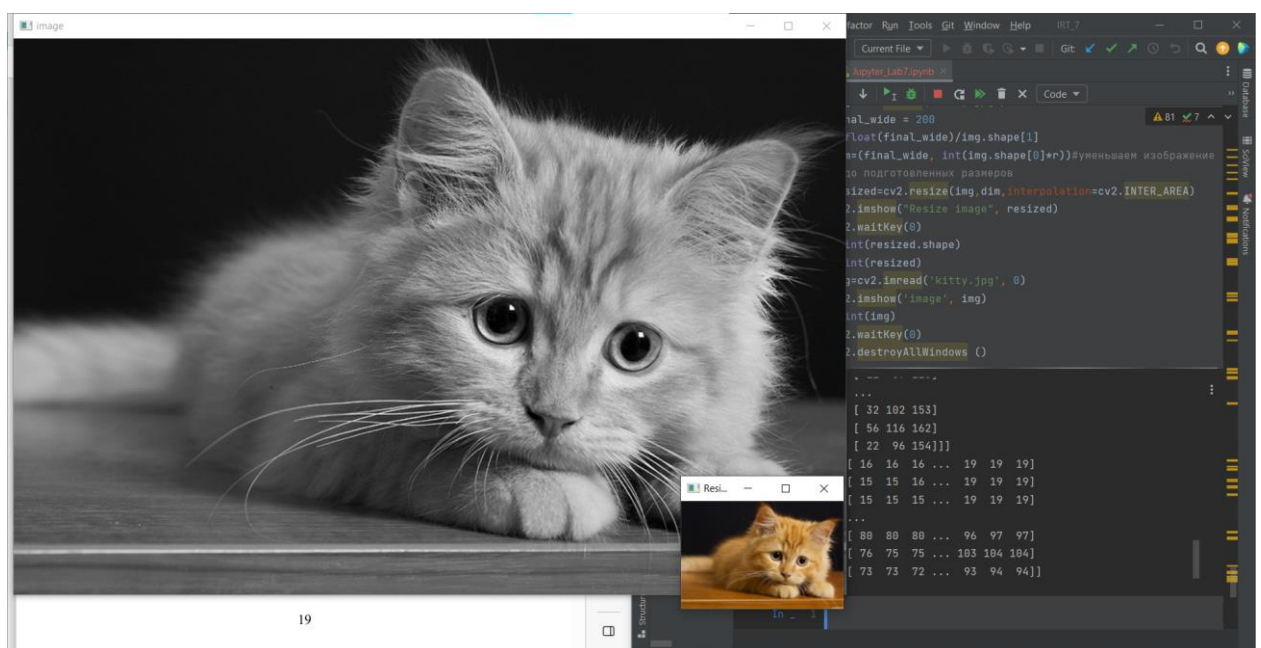


Рисунок 7.22 – Результат выполнения программы

Задание 1.10.

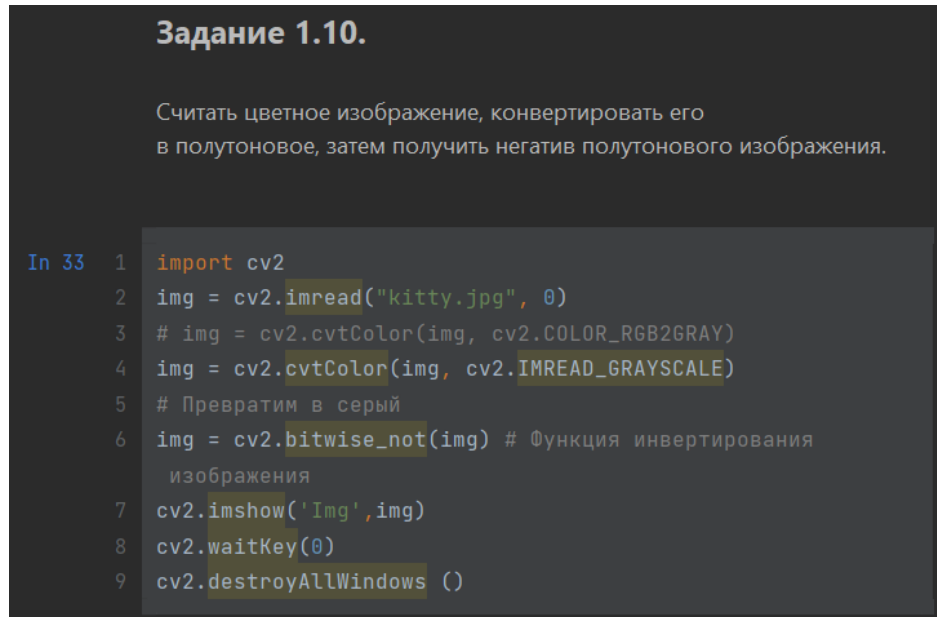


Рисунок 7.23 – Код программы

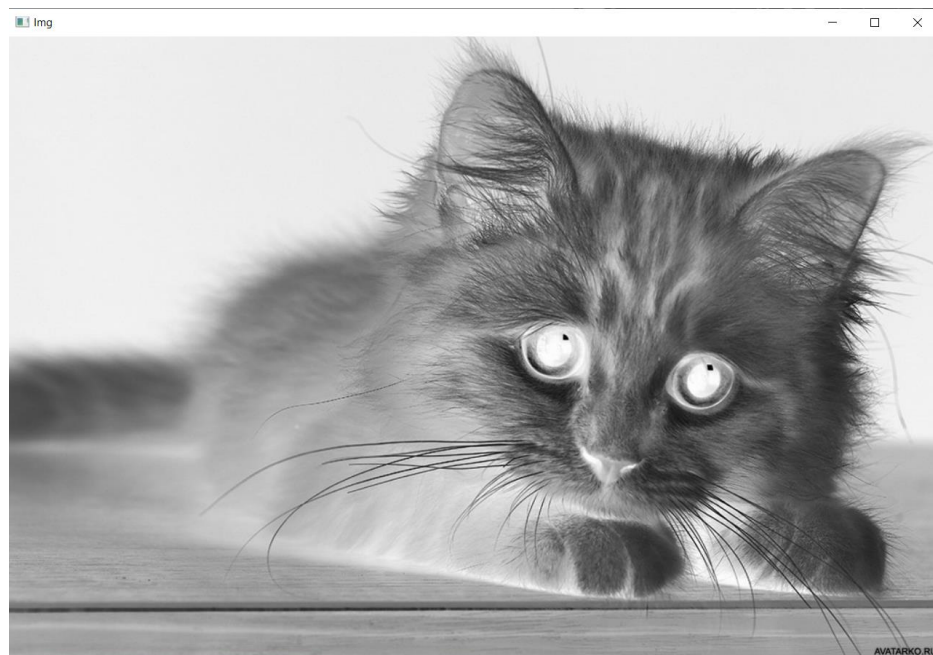


Рисунок 7.24 – Результат выполнения программы

Индивидуальное задание

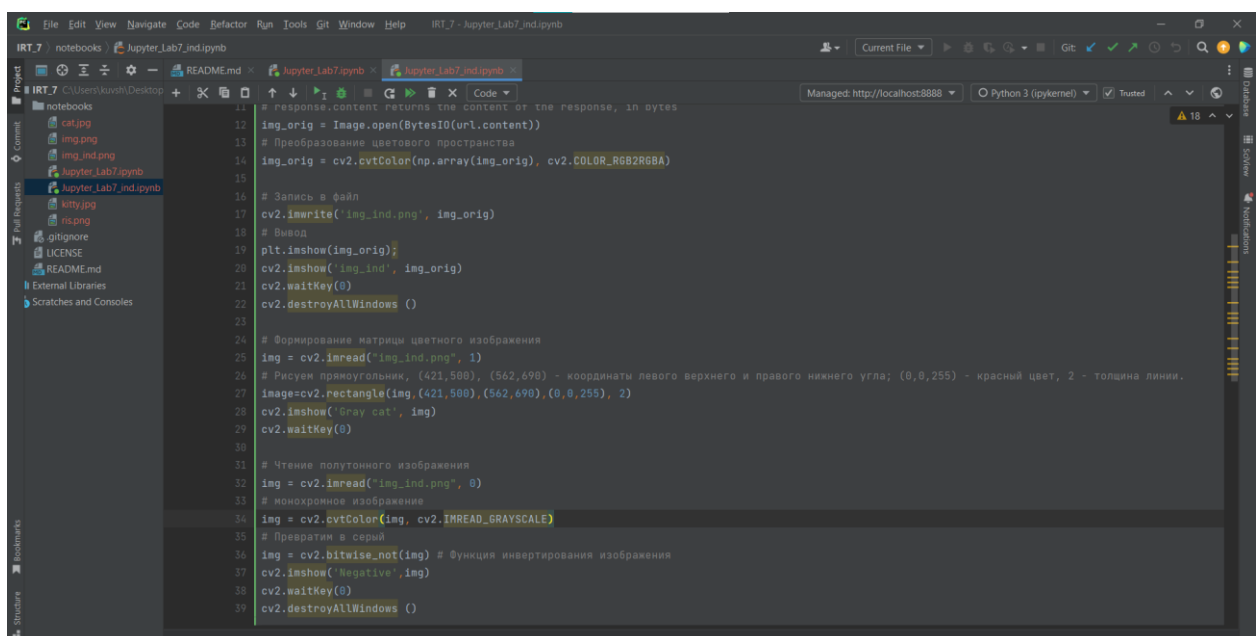
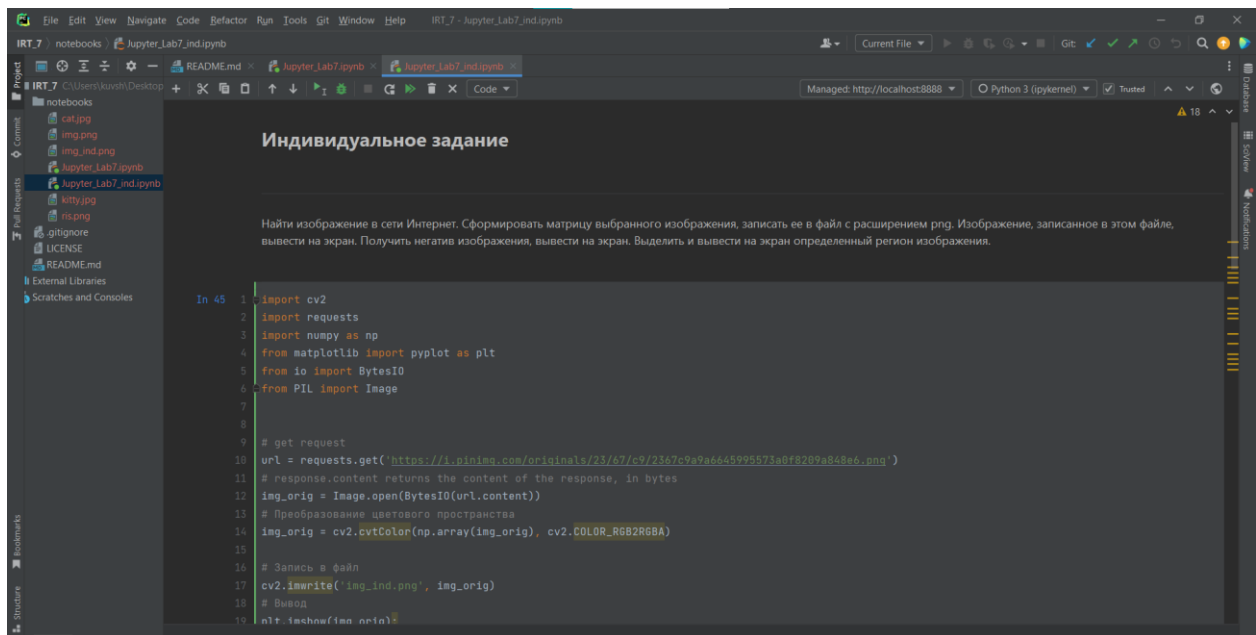


Рисунок 7.25 – Код программы

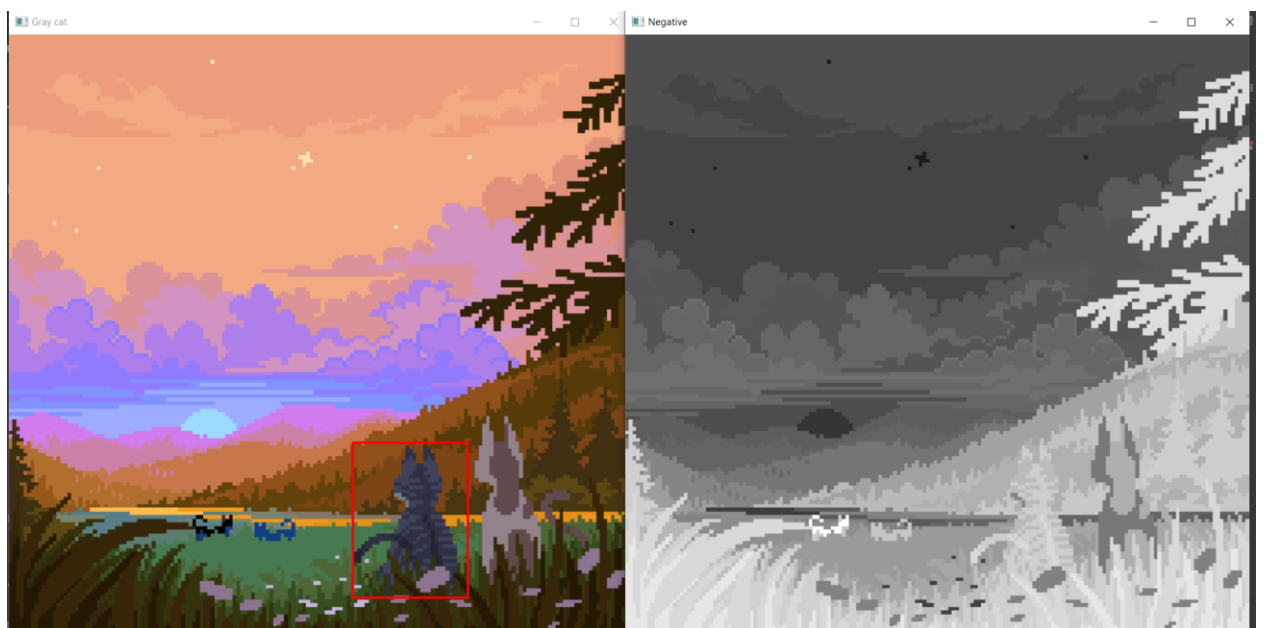
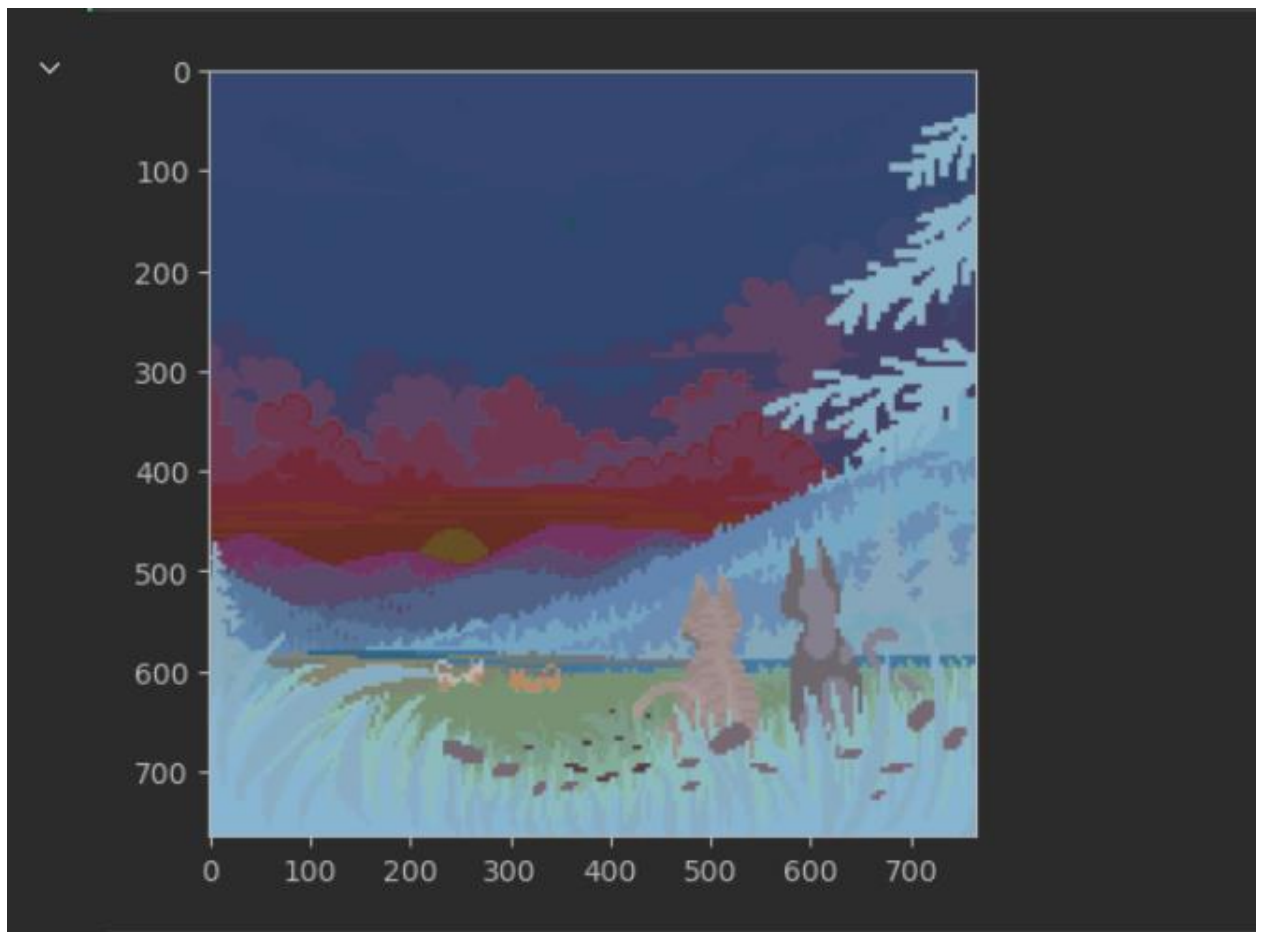


Рисунок 7.26 – Результат выполнения программы