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

Институт цифрового развития
Кафедра инфокоммуникаций

«Цифровая обработка бинарных изображений»

ОТЧЕТ
по лабораторной работе №10
дисциплины
«Технологии распознавания образов»

Выполнил:
Борсуков Владислав Олегович
2 курс, группа ПИЖ-б-о-21-1,
011.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

_____
(подпись)

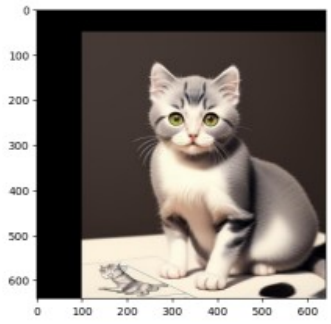
Проверил:
_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 г.

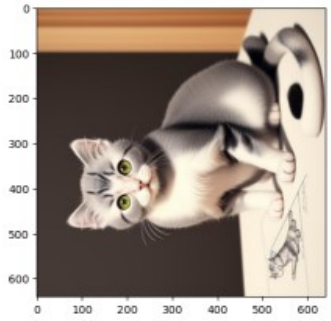
Задание 4.2 Определить размер изображения и сдвинуть изображение на 100 столбцов и 50 строк.

```
In [8]: rows,cols,colors = img.shape
        M = np.float32([[1,0,100],[0,1,50]])
        dst = cv2.warpAffine(img,M,(cols,rows))
        plt.imshow(dst);
```



Задание 4.3 Определить размер изображения, его центр и повернуть его на 90 градусов.

```
In [9]: M = cv2.getRotationMatrix2D((cols/2,rows/2),90,1)
        dst = cv2.warpAffine(img,M,(cols,rows))
        plt.imshow(dst);
```



Задание 4.4 Определить размер изображения, задать 3 точки, изменить их координаты и провести аффинное преобразование всего изображения по этим точкам

```
In [10]: pts1 = np.float32([[50,50],[200,50],[50,200]])
         pts2 = np.float32([[10,100],[200,50],[100,250]])

         M = cv2.getAffineTransform(pts1,pts2)
         dst = cv2.warpAffine(img,M,(cols,rows))

         img_print(img, dst);
         plt.show()
```

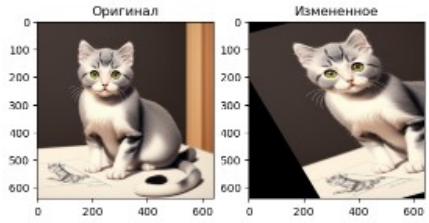
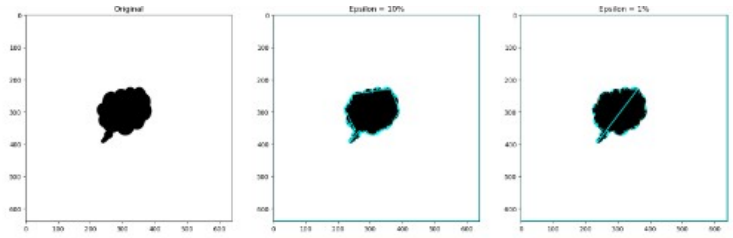
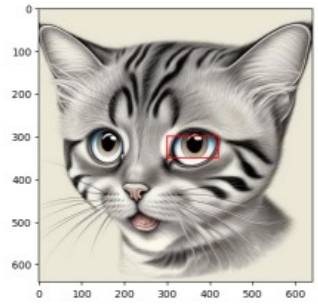


Рисунок 1 – Проработка заданий из учебника



Задание 4.11 Нарисовать прямоугольник в месте, где нужно вырезать фрагмент , вывести на экран фрагмент, ограниченный прямоугольником, увеличив этот фрагмент. Определить размер изображения, его центр и повернуть его на 90 градусов.

In [18]:
```
img = cv2.imread('imgs/cat2.jpg', 1)
image = cv2.rectangle(img, (300, 350), (420, 300), (0, 0, 255), 2)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
plt.imshow(image);
```

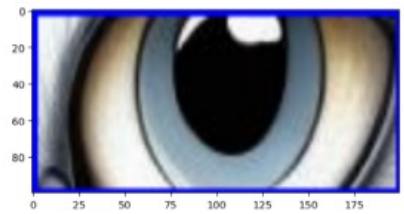


In [19]:
```
crop = img[300:350, 300:420]
piece = cv2.resize(crop, (200,100), interpolation=cv2.INTER_LINEAR)

(h, w) = piece.shape[:2]
print(w,h)

plt.imshow(piece);
```

```
200 100
```



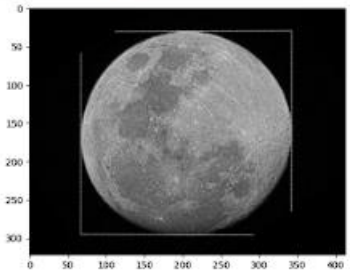
In [20]:
```
center = (w / 2, h / 2)
M = cv2.getRotationMatrix2D(center, 90, 1)
rotated = cv2.warpAffine(piece, M, (150, 150))

plt.imshow(rotated);
```

Рисунок 2 – Проработка заданий из учебника

Выравниваем изображение

In [157]:
```
rows,cols = img.shape
M = cv2.getRotationMatrix2D((cols/2,rows/2),32.2,1)
dst = cv2.warpAffine(img,M,(cols,rows))
plt.imshow(dst, cmap='gray');
```



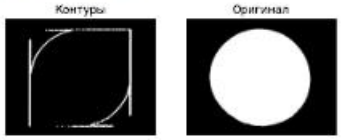
Получаем новые контуры

In [158]:
```
ret,thresh = cv2.threshold(dst,30,255, cv2.THRESH_BINARY)
contours, hierarchy = cv2.findContours(thresh, 5, 5)

cont = np.zeros_like(img)
cv2.drawContours(cont, contours[0], -1, 255, 3)

plt.subplot(121)
plt.title('Контуры')
plt.axis('off')
plt.imshow(cont, cmap='gray');

plt.subplot(122)
plt.title('Оригинал')
plt.axis('off')
plt.imshow(mask, cmap='gray');
```



Обрезаем все, что не входит в контур

In [159]:
```
(y, x) = np.where(mask == 255)
min_y = np.min(contours[0],  axis=0)[0][1]
min_x = np.min(contours[0],  axis=0)[0][0]
max_y = np.max(contours[0],  axis=0)[0][1]
max_x = np.max(contours[0],  axis=0)[0][0]

out = out[min_y:max_y+1, min_x:max_x+1]

plt.axis('off')
plt.imshow(out, cmap='gray');
```

Рисунок 3 – Индивидуальное задание

1. С помощью какой функции можно совершить изменение размера изображения?

cv.resize (img, dim, interpolation=…) Первый аргумент – матрица изображения, второй dim либо width, height – размер изображения, третий – метод интерполяции

2. Какие существуют способы изменения размера?

− Размер нового изображения указывается в процентах (например: 50%): scale_percent = 50.

− Размер изображения задается вручную: width=58, height=71.

− Размер изображения задается с помощью коэффициента масштабирования.

3. Перечислите основные методы интерполяции.

cv.INTER_AREA – для сжатия, cv.INTER_CUBIC и cv.INTER_LINEAR – для масштабирования. По умолчанию используется метод интерполяции cv.INTER_LINEAR.

4. С помощью какой функции можно осуществить сдвиг изображения?

cv2.warpAffine(src, M, dsize[, dst[, flags[, borderMode[, borderValue]]]]) src - изображение. Матрица M - преобразования. dsize - размер выходного изображения. flags-комбинация методов интерполяции (тип int!) borderMode - режим пикселей границы (тип int!) borderValue - (выделение) Значение заполнения границы; по умолчанию это 0.

5. С помощью какой функции можно осуществить вращение изображения?

cv2.getRotationMatrix2D(center, angle, scale) center: Центр вращения angle(θ): угол поворота. scale: коэффициент масштабирования

6. Что происходт при аффинной трансформации изображения?

При аффинном преобразовании все параллельные линии исходного изображения остаются параллельными и в выходном изображении.

7. Какие функции позволяют выполнить охват объекта?

Функция cv2.drawContours() возвращает структуру box, которая содержит 37 следующие аргументы: верхний левый угол (x, y), ширину, высоту, угол поворота. Чтобы нарисовать прямоугольник, нужны 4 угла прямоугольника, которые задаются функцией cv2.boxPoints (). Окружность с минимальной площадью, охватывающей объект, можно нарисовать м с помощью функции cv2.minEnclosingCircle (). Используя функцию cv2.ellipse(), можно вписать изображение в эллипс с минимальной площадью.

8. Опишите процесс создания выпуклой оболочки вокруг контура

Чтобы нарисовать выпуклую оболочку вокруг контура некоторого изображения, выделяем все его крайние точки и соединяем их ломанной прямой линией. Ни одна точка изображения не должна выходить за пределы выпуклой оболочки. Импортируем цветное изображение и трансформируем его в полутоновое изображение. Функция Canny выделяет контуры, а с помощью функции cv2.findContours() создаем иерархию контуров. Выделяем только внешние контуры изображения. Затем, используя цикл for, проходим по каждому из контуров изображения. С помощью переменной hull создаем выпуклую оболочку сначала для первого контура, затем для каждого другого контура. В результате получим контур, охватывающий изображение.

9. Какая функция позволяет аппроксимировать контур?

Функция cv2.approxPolyDP(cnt,epsilon,True). Первый аргумент cnt = contours [i] – массив с координатами пикселей контура, аргумент epsilon задается в процентах, с уменьшением epsilon максимальное расстояние между ломаной прямой, аппроксимирующей контур, и самим контуром также уменьшается. Значение этого аргумента вычисляется функцией epsilon = 0.1*cv2.arcLength(cnt,True).

10. Как осуществить выделение на изображении интересующей области, создание для нее отдельного изображения

Выделим на изображении интересующую нас область, заключив ее в прямоугольную рамку с помощью функции рисования cv2.rectangle. Фрагмент изображения, заключенный в рамке, выведем на экран. Используя функцию .shape, получим размер изображения и изменим его с помощью функции cv2.resize. Функция 42 cv2.getRotationMatrix2D предназначена для поворота изображения, а функция cv2.warpAffine – для аффинного преобразования.