

# КУРЕЦ Любовь ИСУ

## ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

### Выделение границ объектов на полутоновом изображении

Используя python библиотеку **OpenCV** и **jupyterthemes** можно выделить границы объектов на изображении.

### ВХОДНОЕ ИЗОБРАЖЕНИЕ



### РЕЗУЛЬТАТ



## Листинг программы

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
from jupyterthemes import jtplot

jtplot.style()
In [2]:
def show_images(im_dict, cols, shape = (1,1), length = None, size = 8):
    if not length:
        length = len(im_dict)
    else:
        length = max(length, len(im_dict))
    w_size = shape[1] / sum(shape) * size
    h_size = shape[0] / sum(shape) * size
    rows = np.ceil(length / cols)
    fig = plt.figure(figsize=(w_size*cols, h_size*rows))
    i = 0
    for key, value in im_dict.items():
        i += 1
        sub = fig.add_subplot(rows, cols, i)
        sub.set_title(key)
        plt.axis('off')
        plt.imshow(value)
    return fig
In [3]:
fn = 'image.jpg'
img = cv.imread(fn, 0)
ret,image = cv.threshold(img,127,255,cv.THRESH_BINARY)

_, contours, hierarchy = cv.findContours( image.copy(),
cv.RETR_EXTERNAL, cv.CHAIN_APPROX_SIMPLE)

img = cv.cvtColor(img, cv.COLOR_GRAY2RGB)
```

```
img_2 = img.copy()
_ = cv.drawContours( img_2, contours, -1, (255,0,0), 3, cv.LINE_AA,
hierarchy, 1 )
In [4]:
im_dict = {'Source':img, 'Contours':img_2}

cols = 1
size = 32
fig = show_images(im_dict, cols, shape = img.shape, size = size)
```