

# Мой проект - фоторедактор

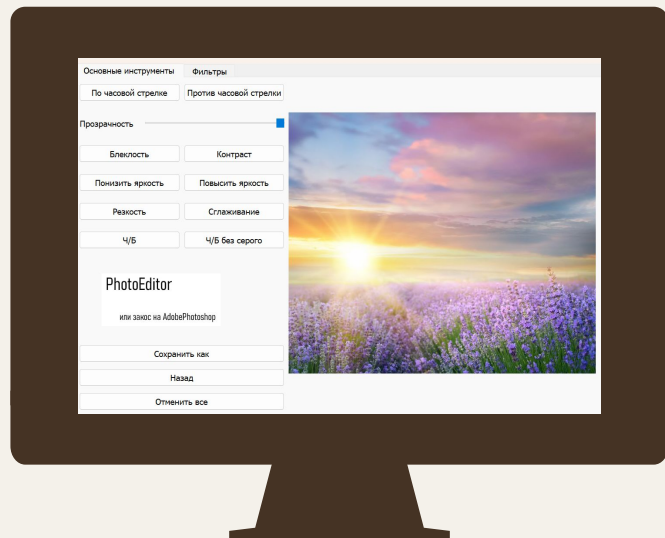
Или закос на  
Adobe Photoshop



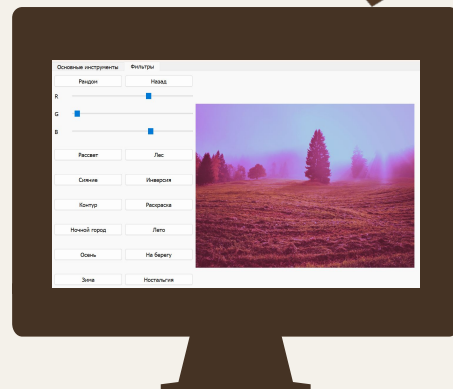
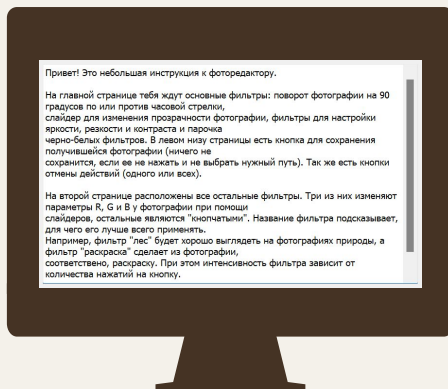
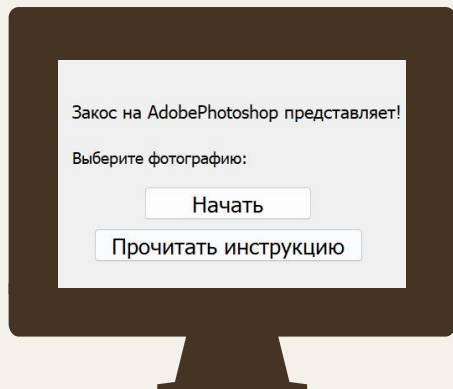
# Какой была цель проекта?

## Сделать фоторедактор с:

1. базовыми функциями;
2. большим количеством фильтров;
3. основными настройками фотографии (яркость, контраст,...);
4. возможностью поворачивать фотографию;
5. возможностью отменять последнее или все действия;
6. возможностью сохранять получившийся результат.



# Как это реализовано?



# Виды фильтров с точки зрения создания

01

Pillow

03

Luts

02

Ручками  
написанные

04


QImage





# Pillow фильтры

Используются фильтры из библиотеки PIL. Для этого изображение QImage переводится в изображение Pillow Image, фильтр применяется и изображение снова становится QImage



```
def contour(self): # создает фотографию с контуром (получается похоже на раскраску)
    self.last_images.append(self.curr_image.copy())


    self.curr_image.save('img.png', "PNG", -1)
    img = Image.open('img.png')
    img = img.filter(ImageFilter.CONTOUR)
    img.save('img.png')

    self.curr_image = QImage('img.png')
    self.pixmap = QPixmap.fromImage(self.curr_image)
    self.picture1.setPixmap(self.pixmap)
    self.picture2.setPixmap(self.pixmap)
```

```
def make_less_bright(self): # понижает яркость фотографии
    self.last_images.append(self.curr_image.copy())

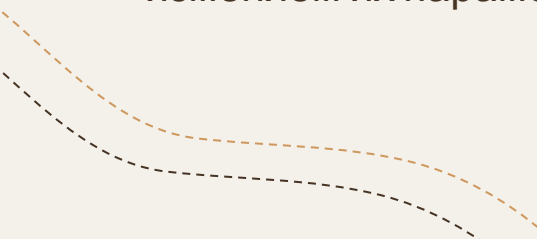
    self.curr_image.save('img.png', "PNG", -1)
    img = Image.open('img.png')
    img = ImageEnhance.Brightness(img)
    img = img.enhance(0.95)
    img.save('img.png')

    self.curr_image = QImage('img.png')
    self.pixmap = QPixmap.fromImage(self.curr_image)
    self.picture1.setPixmap(self.pixmap)
    self.picture2.setPixmap(self.pixmap)
```



# Написанные ручками фильтры

Мы проходимся по пикселям и  
изменяем их параметры



```
def change_color_blue(self): # добавляет больше синего оттенка фотографии
    self.last_images.append(self.curr_image.copy())


    value = int(self.BlueSlider.value())

    x, y = self.curr_image.size().width(), self.curr_image.size().height()

    con = sqlite3.connect('YLdb_1.db')
    cursor = con.cursor()
    m = cursor.execute("""SELECT id FROM ColorSliders WHERE color = 'blue' ORDER BY id DESC LIMIT 1""").fetchone()
    m = m[0]
    cursor.execute(f"INSERT INTO ColorSliders VALUES ({m + 1}, {value}, 'blue')")
    n = cursor.execute(f"SELECT value FROM ColorSliders WHERE color = 'blue' AND id = {m}").fetchone()
    n = n[0]
    k = value - n
    con.commit()
    con.close()

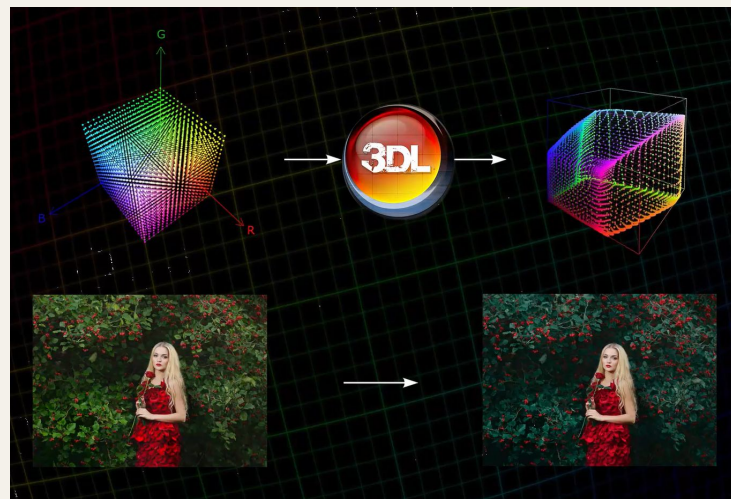
    for i in range(x):
        for j in range(y):
            r, g, b, _ = self.curr_image.pixelColor(i, j).getRgb()
            b += k
            b = 255 if b > 255 else b
            b = 0 if b < 0 else b
            self.curr_image.setPixelColor(QPoint(i, j),
                                           QColor(r, g, b))

    self.pixmap = QPixmap.fromImage(self.curr_image)
    self.picture1.setPixmap(self.pixmap)
    self.picture2.setPixmap(self.pixmap)
```



# Luts - что это вообще такое?

По сути, это уже готовый cube-файл, в котором подобраны цвета, на которые заменять каждый цвет на фото. Писать много, лучше расскажу это устно.



```
def nostalgia(self): # добавляет фильтр "ностальгия"
    self.last_images.append(self.curr_image.copy())

    self.curr_image.save('img.png', "PNG", -1)
    lut = load_cube_file('nostalgia.cube')
    img = Image.open('img.png').filter(lut)
    img.save('img.png')

    self.curr_image = QImage('img.png')
    self.pixmap = QPixmap.fromImage(self.curr_image)
    self.picture1.setPixmap(self.pixmap)
    self.picture2.setPixmap(self.pixmap)
```



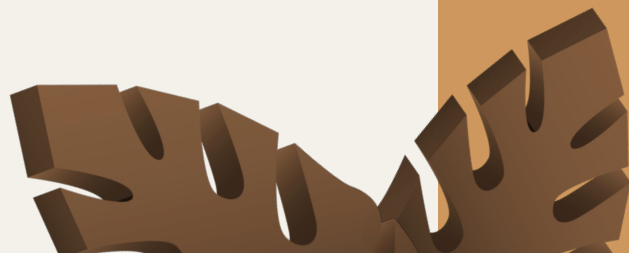
# QImage фильтры

Применяем уже  
существующие для QImage  
изображений фильтры

```
def right_rotate(self): # поворачивает фото на 90 градусов вправо (по часовой стрелке)
    self.last_images.append(self.curr_image.copy())

    t = QTransform().rotate(90)
    self.curr_image = self.curr_image.transformed(t)

    self.pixmap = QPixmap.fromImage(self.curr_image)
    self.picture1.setPixmap(self.pixmap)
    self.picture2.setPixmap(self.pixmap)
```







# На этом все!

Можем перейти к  
практической части

