**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ**



**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ «ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ ИМЕНИ П.А. ОВЧИННИКОВА»**

**Специальность:**

09.02.07 Информационные системы и программирование

**КУРСОВАЯ РАБОТА**

**МДК.01.01. Разработка программных модулей** тема: «Конвертер изображений»

|  |  |
| --- | --- |
|  | **Ф.И.О подпись** |
| Группа №: | 2ИС |
| Руководитель: | Олейник Владимир Иванович  **Ф.И.О подпись** |
| Оценка: | Дата: «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2023 г. |

Студент: Мерясов Евгений Евгеньевич

**Москва**

**2023**

**СОДЕРЖАНИЕ**

Стр.

Введение……………………………………………………………. 3

1. Определение Конвертера…………………………………………... 4
   1. Использование конвертера в наше время……………………….……… 4
   2. Особенности и недостатки……………………………………………… 5
   3. Возможности конвертера…………………….……………………….. 7
2. Разработка приложения………………………………………………….. 8
   1. План составления программы…………………………………………… 9
   2. Составление требований к приложению………………………………... 9
   3. Пользовательский интерфейс……………………………………………. 11
   4. Процесс кодирования программы……………………………………….. 14

Заключение………………………………………………………………… 20

Список использованной литературы и интернет – источников……….. 21

Приложение А…………………………………………………………….. 22

**Введение**

Конвертер изображений является одним из более применимых программ для конвертирования расширений и типов файлов в себе подобные. В данной курсовой работе я хочу представить Вам свою небольшую программу, над которой применялись не малое количество библиотек и функций и приложил достаточно знаний в языке программирования Python для её функционирования. Задача моей курсовой была в создании простого программного продукта состоящий из нескольких функционирующих кнопок с которыми пользователь может работать, в это число и входит конвертирование изображений.

Данная программа называется которую я разрабатывал этот месяц — Онлайн Конвертер изображений. Цели работы для моей курсовой заключается в создании простого конвертера что сможет выполнять нижеперечисленные задачи, а именно:

* Загрузка изображений с компьютера.
* Выбор формата выходного изображения.
* Конвертация изображения с сохранением качества.
* Возможность указания размера изображения и его пропорций.
* Удобный интерфейс для выбора файлов и настроек конвертации.
* Сохранение готовых изображений на компьютер.
* Поддержка разных типов изображений.

Курсовая была выполнена в довольно простом, но эффективном состоянии, где для вас каждая кнопка будет функционировать и правильно работать, а дальнейшие разработка и наработки модифицируют мою программу уже при возможной защите диплома, однако сейчас остановимся на главных вопросах, проще говоря объясним несколько базовые моментов. Для выполнения своей программы, я использовал язык программирования Python в среде программирования PyCharm.

Этот язык программирования полезен по следующим причинам:

* Широкая применимость: Python используется в различных областях, таких как веб-разработка, машинное обучение, анализ данных, автоматизация, научные вычисления и т.д.
* Простота изучения и использования: Python имеет простой и понятный синтаксис, что делает его легким для изучения и начала работы.
* Библиотеки и фреймворки: Python предлагает большое количество библиотек и фреймворков для решения различных задач, что упрощает разработку приложений.
* Открытый исходный код: Python является открытым программным обеспечением, что позволяет пользователям участвовать в его разработке и улучшении, а также использовать его бесплатно.
* Производительность: Несмотря на свою простоту, Python обладает хорошей производительностью, особенно в задачах, связанных с машинным обучением и анализом данных.
* Интерпретируемый язык: Python не требует компиляции перед исполнением, что упрощает процесс разработки и отладки.
* Кросс-платформенность: Python может быть использован на различных операционных системах без необходимости изменений в коде.

**1. Определение Конвертера**

**1.1. Использование конвертера в наше время**

И так, что из себя представляет конвертирование изображений? Всё довольно просто, если не углубляться. Конвертирование изображений процесс преобразования одного формата файла/изображения/аудио-файлов в другой. Проще говоря, если вы желаете перевести формат из JPG или WEBP в PNG, то конвертирование вам поможет с этим, а его использование является простейшим, для этого вам всего лишь нужно:

* Загрузить изображение.
* Выбрать желаемый формат, можете даже выбрать видео формата MP4 чтобы допустим перевести его в формат GIF.
* Нажать кнопку “Конвертировать”.

Ниже приведена сторонняя программа Онлайн Конвертера и принцип его работы.



Рисунок 1. Пример браузерного конвертера

В 21 веке, большое количество изображений, документов, видео материалов используются лишь в одном формате, из-за чего извлечение какой-либо информации из них становится труднодоступным, в том числе и невозможность адекватного редактирования, что допустим помогло бы сократить вес вашего изображения (Пример: вы допустим что-то обрезаете в видео, редактируете его, и в итоге в нем сокращается его размер/вес), поэтому люди додумались изобрести то, что поможет им передавать изображение из одного формата в другой и не только. Сейчас, конвертер используется для офисных работ, в учебе и даже в обычных домашних делах. Но какие же в нём особенности и недостатки?

Конвертер изображений можно использовать следующими способами: – Через графический интерфейс программы: выбрать файл, указать параметры преобразования и запустить процесс конвертации.

* С помощью командной строки или скрипта: написать команды для преобразования изображений, сохранить их в файл и выполнить.
* Через API или плагин для другой программы: интегрировать конвертер изображений с другим приложением и использовать его функции для обработки файлов.

**1.2. Особенности и недостатки**

|  |  |
| --- | --- |
| Преимущества | Недостатки |
| Удобство использования | Ограничение на размер и формат |
| Быстрота | Невозможность выполнения сложных задач |
| Обширный выбор инструментов | Зависимость от скорости интернета |
| Бесплатность | Конфиденциальность данных |

Преимущества:

* + - Удобство. Онлайн конвертеры не требуют установки на компьютер и доступны в любой точке мира. Нет необходимости искать специалиста, если надо преобразовать формат, с помощью онлайн конвертеров это можно сделать самостоятельно.
    - Быстрота. Онлайн конвертеры быстро выполняют преобразование изображения, меняют его формат и уменьшают размер.
    - Большой выбор инструментов. Онлайн-конвертеры предоставляют широкий спектр функций, которые могут быть использованы для настройки изображения, например, наложение фильтров и корректировка цвета.
    - Бесплатность. Онлайн конвертеры предоставляют возможность работать с изображениями бесплатно.

Недостатки:

* + - Нарушение конфиденциальности. Конвертирование или изменение изображений в онлайн-режиме может представлять угрозу конфиденциальности и безопасности. Сервис может использовать загруженные изображения для рекламных целей или осквернения данных пользователей.
    - Ограничения на размер и формат. Онлайн-конвертеры имеют ограничения на размер и формат обрабатываемых изображений. Недостаточно мощный конвертер может не справиться с большой картинкой или не обрабатывать нестандартные форматы файлов.
    - Невозможность выполнения сложных задач. Тем более что онлайн-конвертеры не могут выполнять сложные задачи, такие как удаление фона изображения или работа с слоями.
    - Зависимость от скорости интернета. Для работы с онлайн-конвертерами требуется высокоскоростное интернет-соединение. Медленное соединение может привести к длительным задержкам при загрузке изображений.

**1.3. Возможности конвертера и его функциональность**

Возможности конвертера очень обширны, разновидность может исходить с обычных конвертеров изображений до конвертирования тяжелых файлов, видео материалов и так далее. Их алгоритм работает довольно просто, а функциональность приложения остается очень эффективной, что помогает пользователям легко обрабатывать, редактировать и менять размер своего файла в тот который он захочет.

Рис. 2 Пример расширений



Конвертер изображений может обладать следующими возможностями:

* Поддержка различных форматов изображений: JPEG, PNG, GIF, BMP и другие.
* Возможность изменения размера изображений.
* Преобразование изображений в различные форматы (например, JPEG в PNG).
* Обработка изображений с использованием фильтров и эффектов (например изменение яркости, контраста, насыщенности).
* Наличие инструментов для обрезки.
* Простой и удобный интерфейс для работы с изображениями.

1. **Разработка приложения**

**2.1. План составления программы**

Что обязательно должен включать в себя каждый онлайн конвертер, разберем поэтапно:

* Наличие интерфейса (Чтобы начать работать в Конвертере, необходимо иметь при себе какой никакой интерфейс/фундамент на котором мы сможем выполнять конвертирование и остальные задачи.

В данном случае мы смело используем библиотеку TKinter).

* Наличие кнопок (Очевидно, для того чтобы работала наше приложение, нам стоит иметь при себе кнопки, а именно: Загрузить изображение и Конвертировать. Конечно, чтобы усовершенствовать можно и добавить множество других функций для нашей программы, по типу обрезки изображения, изменения его качества и т.д.).
* Работоспособность (С учетом всех вышеперечисленных пунктов мы можем смело запустить наш Конвертер и проверить его на работоспособность).

После соблюдения всех необходимых пунктов, мы можем переходить к составлению требований к приложению.

**2.2. Составление требований к приложению**

Наименование программы: Онлайн Конвертер

Основание для разработки: Углубление знаний в языке программирования Python и среде программирования PyCharm. Создание программы “Онлайн Конвертер”.

Назначение разработки: Создание программного средства для конвертации файлов в различные виды форматов.

Назначение и область применения: Предоставление пользователю полного функционала данного приложения.

Программа используется в различных отраслях и сферах деятельности, включая мультимедиа, веб-разработку, научно-исследовательскую работу, финансы и многое другое.

**Требования к программе:**

* Приложение “Онлайн Конвертер”
* Графический интерфейс должен быть доступным для понимания пользователем и корректно отображать кнопки.
* Приложение должно иметь свое уникальное имя.

**Спецификации качества программного обеспечения:**

* Производительность (Программа должна соблюдать требования к аппаратным ресурсам и системным требованиям для поддержания её производительности).
* Практичность (Программа должна иметь понятный пользовательский графический интерфейс).
* Надежность (Программа не является автономной, так как полностью будет завязана на Интернетном обеспечении и связана с любым браузером).
* Функциональность (Программа должна являться полностью завершенным ПО).

**Требования к программной документации:**

* Документация прилагается.

**Стадии и этапы разработки программы:**

* Формирование технического задания ✔
* Проектирование ✔
* Разработка ✔
* Тестирование ✔
* Руководство по эксплуатации программы ✔

**Проектирование программы**

На данном этапе разработки моей программы исследуется структура разрабатываемой системы. В конечном итоге в данном разделе разъясняется полная информация о реализации системы и содержащая в себе достаточное количеств объяснений об её функциональности.

Проектирование разделяется на:

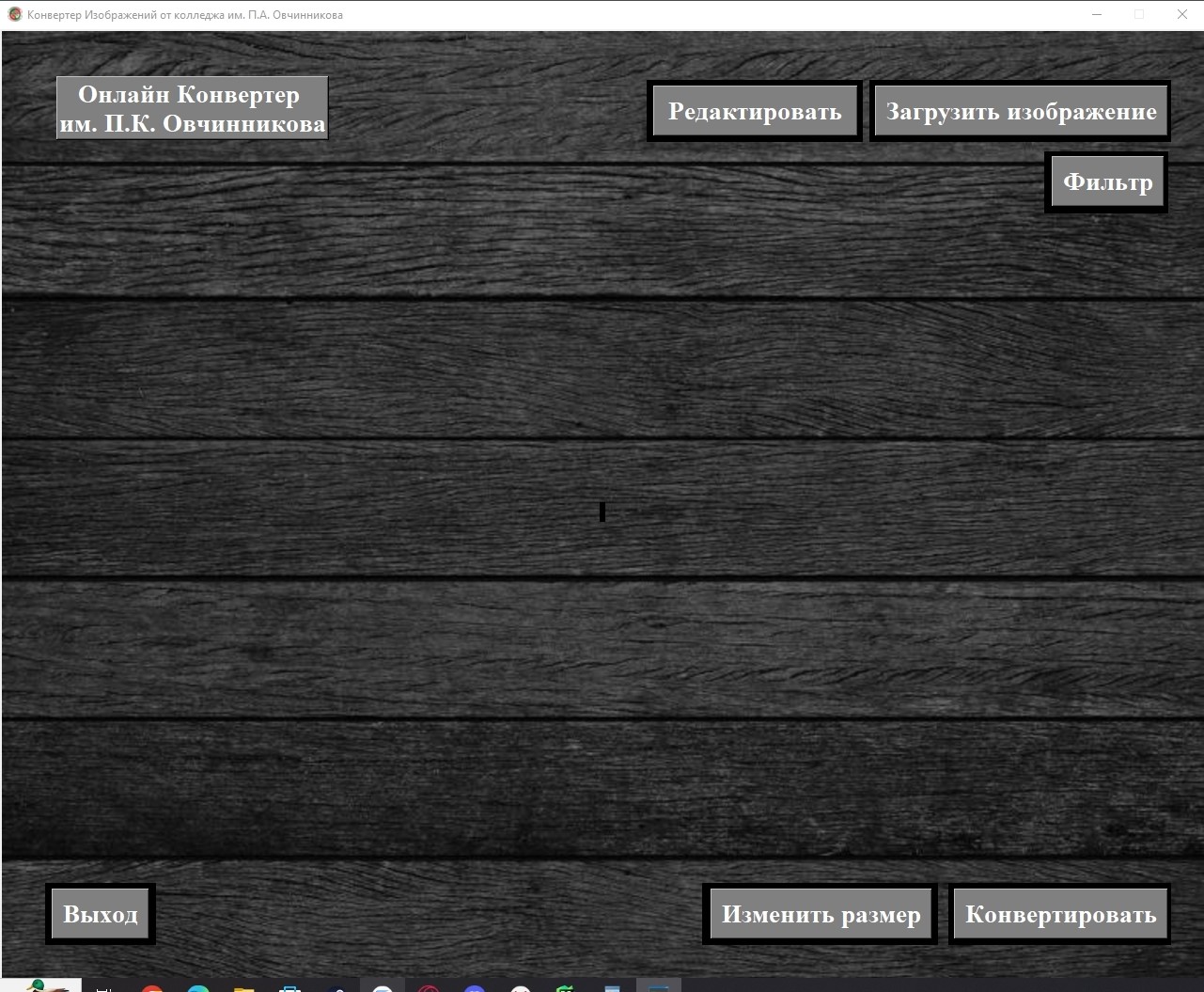
* Архитектура приложения
* Пользовательский интерфейс

**2.3. Архитектура приложения**

Все функционирующие кнопки обрабатываются через функции def, что дает пользователю возможность достичь полной функциональности приложения, а также уверенность в её практичности и эффективности использования.

Смена состояния нажатых кнопок, постоянное обновление окон.

**2.3. Пользовательский интерфейс**



Так выглядит основной интерфейс моей программы, а ниже будут перечислены название основных виджетовы.

Кнопка 1 - “Загрузить изображение”

Кнопка 2 - “Конвертировать”

Кнопка 3 - “Редактировать”

Кнопка 4 - “Изменить размер

Кнопка 5 - “Фильтр”

Кнопка 6 - “Выход”

Label - (Место для вставки изображения)

При нажатии на кнопку “Загрузить изображение”, выводится проводник с которого мы берем наше изображение любого формата.

При нажатии на кнопку “Конвертировать”, выводится проводник где мы конвертируем наше изображение в тот формат файла который нам нужен. При нажатии на кнопку “Редактировать”, пользователю выводится окно в котором он должен ввести координаты для обрезки изображения.

При нажатии на кнопку “Фильтр” выводится новое окно в котором пользователь может смело выбрать фильтр который ему нужен, ОДНАКО сразу же после нажатия “Применить фильтр”, ему выводится окно с дальнейшим конвертированием изображения, но с наложенным ему фильтром. Ниже представлен блок кода для созданий кнопок и оформления для них в целом.

#Создаем главное окно. root = Tk()

root.geometry("1281x1025") root.title("Конвертер Изображений от колледжа им. П.А. Овчинникова") root.iconbitmap('pk3.ico') root.image = ImageTk.PhotoImage(file='doski2.png') root.resizable(width=False, height=False) logo = Label(root, image=root.image) logo.grid(row=0, column=0)

label6 = Label(root, bg="black") #Label который применяется в качестве места для вставки изображение. label6.grid(row=0, column=0)

label5 = Label(root, text="Онлайн Конвертер \nим. П.К. Овчинникова", fg="white", font="Times 20 bold", relief=RAISED, bg="gray") label5.grid(row=0, column=0, sticky="nw", pady=50, padx=60) label1 = Label(root, width=45, height=4, bg="black") label1.grid(row=0, column=0, sticky="ne", pady=54, padx=37)

label2 = Label(root, width=33, height=4, bg="black") label2.grid(row=0, column=0, sticky="se", pady=54, padx=37)

label3 = Label(root, width=32, height=4, bg="black") label3.grid(row=0, column=0, sticky="ne", pady=54, padx=365)

label4 = Label(root, width=18, height=4, bg="black") label4.grid(row=0, column=0, stick="ne", pady=130, padx=40)

#Создаем множество кнопок и присваиваем им команды, то есть наши функции в который содержится остальной код.

load = Button(root, text="Загрузить изображение", bg="gray", fg="white", font=('Times New Roman', 20,

'bold'),command=zagruzit) load.grid(row=0, column=0, sticky="ne", pady=60, padx=40)

konvert = Button(root, text="Конвертировать", bg="gray", fg="white", font='Times 20 bold', command=konv) konvert.grid(row=0, column=0, sticky="se", pady=60, padx=40)

filtr = Button(root, text='Фильтр', bg="gray", fg="white", font='Times 20 bold', command=nal\_filtr) filtr.grid(row=0, column=0, stick="ne", pady=135, padx=44)

edit = Button(root, text="Редактировать", bg="gray", fg="white", width=13, font='Times 20 bold',command=red) edit.grid(row=0, column=0, stick="ne", pady=60, padx=370)

exit = Button(root, text="Выход", bg="gray", fg="white", font='Times 20 bold', command=exit\_conv) exit.grid(row=0, column=0, stick="sw", pady=60, padx=55) root.mainloop()

label6 – Данный label используется как место для вставки загруженного изображения.

**2.3. Процесс кодирования программы**

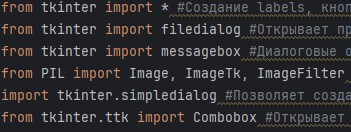


Рисунок 2. Используемые библиотеки

from tkinter import \* #Создание labels, кнопок и так далее from tkinter import filedialog #Открывает проводник через который мы загружаем изображение и применяем через него формат/расширение файла.

from tkinter import messagebox #Диалоговые окна для предоставления информации о том что было сделано, добавлено, отменено.

from PIL import Image, ImageTk, ImageFilter #Библиотека PIL в основном нужна для загрузки изображения на главное окно, а также для наложения фильтров.

import tkinter.simpledialog #Позволяет создать диалоговое окно в котором программа спрашивает пользователя о вводе данных (в данном случае координаты нашего изображения) from tkinter.ttk import Combobox #Открывает выбор того что мы хотим взять. Применяется для фильтров

#Функция zagruzit() используется для загрузки изображения на главное окно в Label6. def zagruzit():

global loaded, current, label6 #Используем global для обозначения переменных в одном коде в каждой функции.

#Создаем переменную path в которой размещаем пути которые доступны для загрузки изображения.

path = filedialog.askopenfilename(filetypes=[("Image files", "\*.jpg \*.png \*.jpeg \*.gif \*.bmp \*.tiff \*.mp4")]) if path: #Если такое расширение изображение существует то оно загружается на главное окно. loaded = Image.open(path) current = ImageTk.PhotoImage(loaded) label6.config(image=current)

#Функция konv() позволяет пользователю конвертировать изображение в любой доступный формат. def konv(): if loaded:

choice = [("JPEG", "\*.jpg"), ("PNG", "\*.png"), ("JPEG

2000", "\*.jpeg"), ("WEBP", "\*.webp"),

("GIF", "\*.gif"), ("PDF", "\*.pdf")] chosen\_file = filedialog.asksaveasfile(filetypes=choice, defaultextension=".jpg") if chosen\_file is not None:

chosen\_filename = chosen\_file.name format\_name = chosen\_file.name.split(".")[-1].lower() # Она разделяет имя файла по точке . и выбирает последний элемент списка с помощью [-1]. Затем метод lower() применяется для преобразования расширения файла в нижний регистр. if format\_name in [fmt[1][2:] for fmt in choice]: # В этой части кода мы обращаемся к элементу с индексом 1 в кортеже fmt. Затем на этом элементе применяем оператор среза [2:], чтобы получить подстроку, начиная с третьего символа до конца строки.

loaded.save(chosen\_filename, format=format\_name) messagebox.showinfo("Успех", "Изображение успешно конвертировано и сохранено.") else:

messagebox.showerror("Ошибка", "Выбран неподдерживаемый формат файла.")

#Функция nal\_filtr() отвечает за наложение фильтра на изображение def nal\_filtr(): if loaded:

win\_filtr = Toplevel(root) #Создание дополнительного окна с Фильтрами.

win\_filtr.title("Выберите фильтр") win\_filtr.geometry("250x100") win\_filtr.transient(root)

#Дается выбор фильтра который будет отображаться с помощью Combobox options = ["DETAIL", "BLUR", "CONTOUR", "SMOOTH"] chosen\_filtr = Combobox(win\_filtr, values=options) chosen\_filtr.pack(pady=10)

#Создается кнопка "Применить фильтр" которая привязана с функцией primenit\_filtr().

apply = Button(win\_filtr, text="Применить фильтр", bg="gray", fg="white", font=('Times New Roman', 15, 'bold'), command=lambda: primenit\_filtr(chosen\_filtr.get())) apply.pack(pady=10)

#Функция primenit\_filtr() нужна чтобы фильтры успешно накладывались на изображение. Для этого у нас есть библиотека

PIL

#В которую мы добавляем модуль ImageFilter и используем его для каждого фильтра. def primenit\_filtr(type):

global loaded, current, label6 if loaded:

filtr\_image = None if type == "DETAIL":

filtr\_image = loaded.filter(ImageFilter.DETAIL) elif type == "BLUR":

filtr\_image = loaded.filter(ImageFilter.BLUR) elif type == "CONTOUR":

filtr\_image = loaded.filter(ImageFilter.CONTOUR) elif type == "SMOOTH": filtr\_image = loaded.filter(ImageFilter.SMOOTH)

#Применение фильтра и отображение измененного изображения на главном окне. if filtr\_image:

current = ImageTk.PhotoImage(filtr\_image) label6.config(image=current)

#Перед сохранением выводится диалоговое окно для окончательного ответа от пользователя. res = messagebox.askyesno("Сохранение", "Желаете сохранить измененное изображение?") if res:

# После нажатия на кнопку Применить фильтр, выскакивает проводник который по факту является тем же Конвертером и конвертирует измененный результат в доступное расширение.

save\_path = filedialog.asksaveasfilename(initialdir="/", title="Сохранить измененное изображение", filetypes=(("PNG", ".png"), ("JPEG",

".jpg;.jpeg"),

("WEBP", ".webp"), ("GIF", ".gif"),

("All files", ".")))

#Если формат выбран, изображение сохраняется if save\_path:

filtr\_image.save(save\_path) messagebox.showinfo("Успех", "Изображение успешно конвертировано и сохранено.")

#Создание функции red(), которая отвечает за обрезание нашего изображения по координатам. Координаты идут от центра изображения. При нажатии на кнопку, будет поочередно выводиться окна ввода. def red():

global loaded, current, label6 if loaded: #С помощью параметра parent, мы связываем всплывающие окна с главным окном чтобы они всплывали над ним, а не за ним.

x = tkinter.simpledialog.askinteger("Обрезать изображение", "Введите координату верхнего левого угла X:", parent=root) y = tkinter.simpledialog.askinteger("Обрезать изображение", "Введите координату верхнего левого угла Y:", parent=root) bottomx = tkinter.simpledialog.askinteger("Обрезать изображение", "Введите координату нижнего правого угла X:", parent=root) bottomy = tkinter.simpledialog.askinteger("Обрезать изображение", "Введите координату нижнего правого угла Y:", parent=root)

# Если пользователь хотя бы раз нажал на кнопку "Отменить" в любом порядке, то программа в конце отменит обрезание.

if x is not None and y is not None and bottomx is not None and bottomy is not None: try:

#С помощью метода crop() изображение обрезается по введенным нами координатам.

crop\_img = loaded.crop((x, y, bottomx, bottomy)) current = ImageTk.PhotoImage(crop\_img) label6.config(image=current) loaded = crop\_img

except ValueError: #Выводит ошибку в случае если координаты были неправильно введены.

messagebox.showerror("Ошибка", "Неверно заданы координаты обрезки.") else:

# И выведет это если пользователь отменяет действие.

messagebox.showinfo("Информация", "Отменено пользователем.") def res\_change(): #Функция для изменения размера изображения по ширине и высоте.

global loaded, current new\_size = tkinter.simpledialog.askstring("Введите новый размер", "Введите новый размер изображения в формате (ширина, высота):") if new\_size:

new\_size = tuple(map(int, new\_size.split(','))) #Создание кортежа целых чисел разделенных запятой. Позволяет получить ширину и высоту введенного размера как отдельные целые числа.

loaded = loaded.resize(new\_size) current = ImageTk.PhotoImage(loaded)

**Листинг кода. Функции.**

def zagruzit(): - Функция для загрузки изображения на главное окно.

def konv(): - Функция для конвертирования изображения в любое доступное расширение.

def nal\_filtr(): - Функция для наложения фильтров на изображение. Пользователю предоставлен выбор конкретных фильтров.

def primenit\_filtr(): - Функция для кнопки “Применить фильтр” дающая возможность функционировать кнопке “Фильтр” в целом, взаимосвязана с предыдущей функцией.

def red(): - Функция редактирования изображения, проще говоря его обрезки по координатам. Обрезание начинается с центра изображения.

def res\_change(): - Функция для изменения размера изображения по координатам (ширина, высота). def exit\_conv(): - Функция для выхода из программы по кнопке.

global loaded, current, label6 – обозначаем переменные в одном коде внутри каждой функции.

**Структура кода**

* Подключение библиотек
* Функции каждой кнопки и выполнение базовых задач
* Создание кнопок и привязка к ним функций и аргументов

**Тестирование программы**

Тестирование моего приложения прошло успешно. Все кнопки функционируют и выводят что необходимо.

После переброски программы в exe расширение, программы все также одинаково функционировала.

**Заключение**

В процессе создания и исследования моей курсовой работы, я углубил свои знания не только в Python, но и в среде программирования PyCharm, ведь совместил множество библиотек, создал рабочий код и в ходе работы оптимизировал его, чтобы он выглядел просто, что в итоге это помогло мне создать функционал полноценной программы “Онлайн Конвертер изображений”.

Однако, что хочу отметить, так это то, что в моей программе присутствуют недочеты и неполный функционал который я не смог применить, но хотелось бы использовать возможные варианты оптимизации моего кода для дальнейшего развития своей темы и его продвижения в деловых и домашних целях.

**Источники**

[**https://www.geeksforgeeks.org/convert-files-from-jpg-to-png-and-vice-1. versa**](https://www.geeksforgeeks.org/convert-files-from-jpg-to-png-and-vice-1.%20versa)[**using-python/**](https://www.geeksforgeeks.org/convert-files-from-jpg-to-png-and-vice-versa-using-python/) [**https://convertio.co/ru/**](https://convertio.co/ru/) [**https://metanit.com/python/tkinter/2.14.php**](https://metanit.com/python/tkinter/2.14.php) [**https://habr.com/ru/articles/681248/**](https://habr.com/ru/articles/681248/) [**https://metanit.com/python/tutorial/7.1.php**](https://metanit.com/python/tutorial/7.1.php)

[**https://ru.anyquestion.info/a/preimuschestva-i-nedostatki-onlayn-konvertero v-dlya-obrabotki-foto**](https://ru.anyquestion.info/a/preimuschestva-i-nedostatki-onlayn-konverterov-dlya-obrabotki-foto) [**https://www.youtube.com/watch?v=o0eMvJqPvAI**](https://www.youtube.com/watch?v=o0eMvJqPvAI) [**https://metanit.com/python/tkinter/5.3.php**](https://metanit.com/python/tkinter/5.3.php)

**Приложение А**

Ниже представлены скриншоты тестирования моей программы

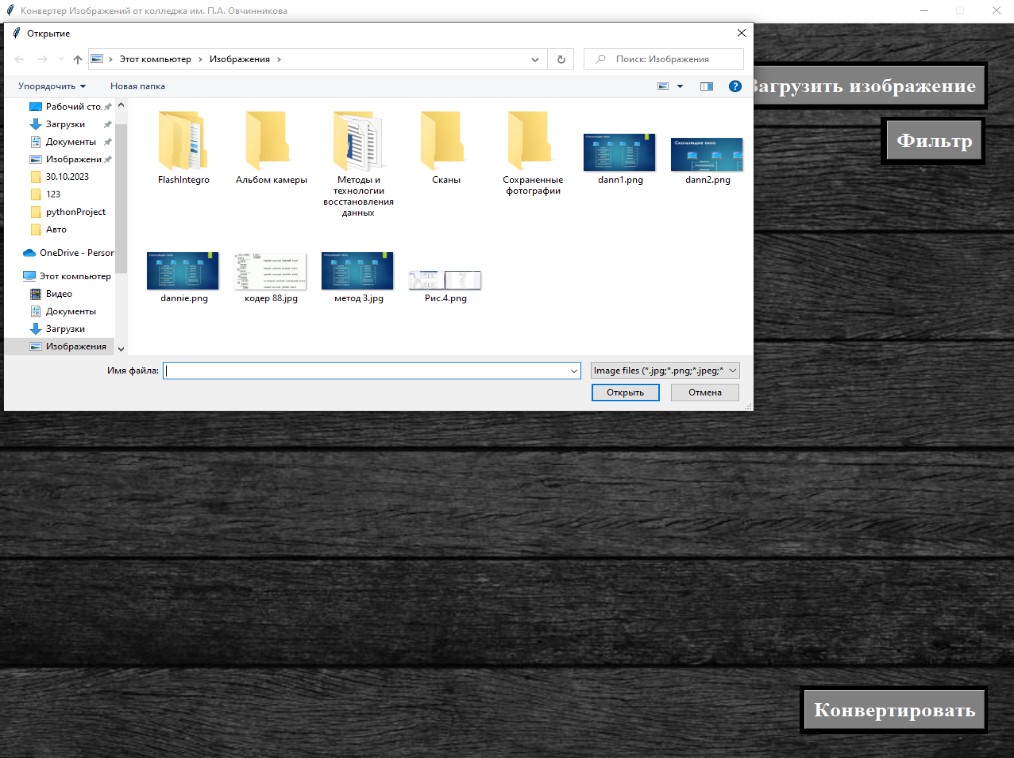


Рисунок 3. Загрузка изображения

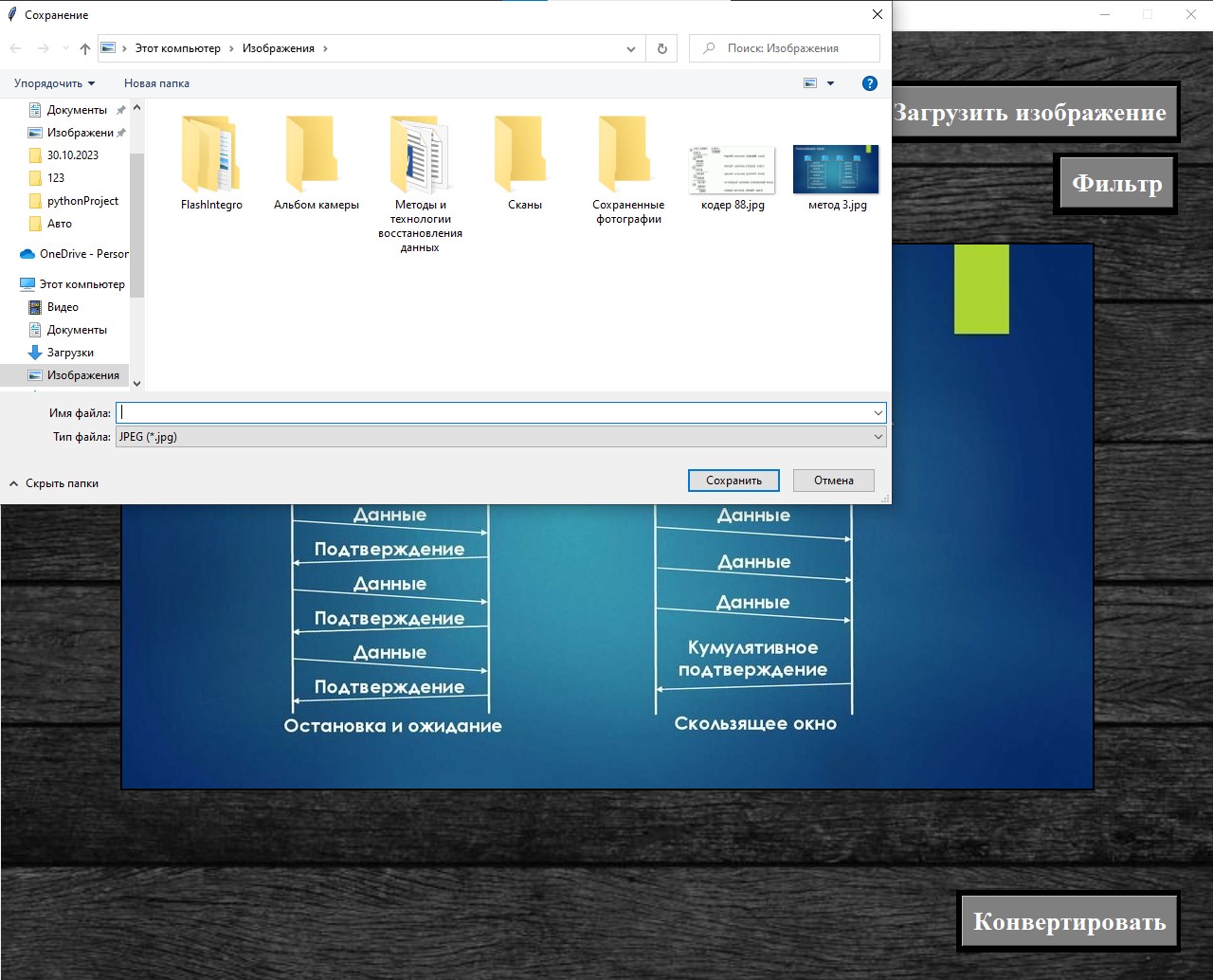


Рисунок 4. Конвертирование

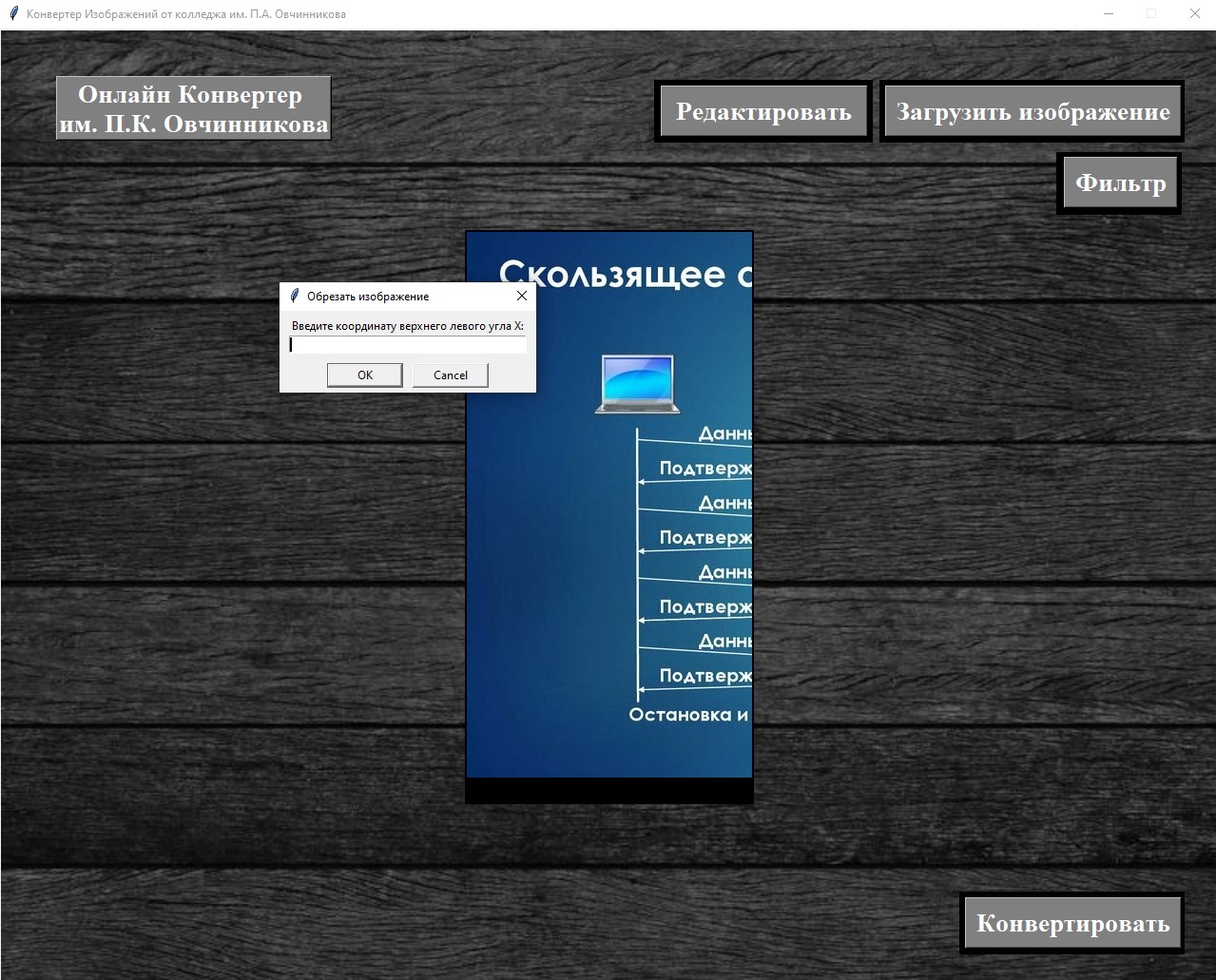


Рисунок 5. Обрезка изображения

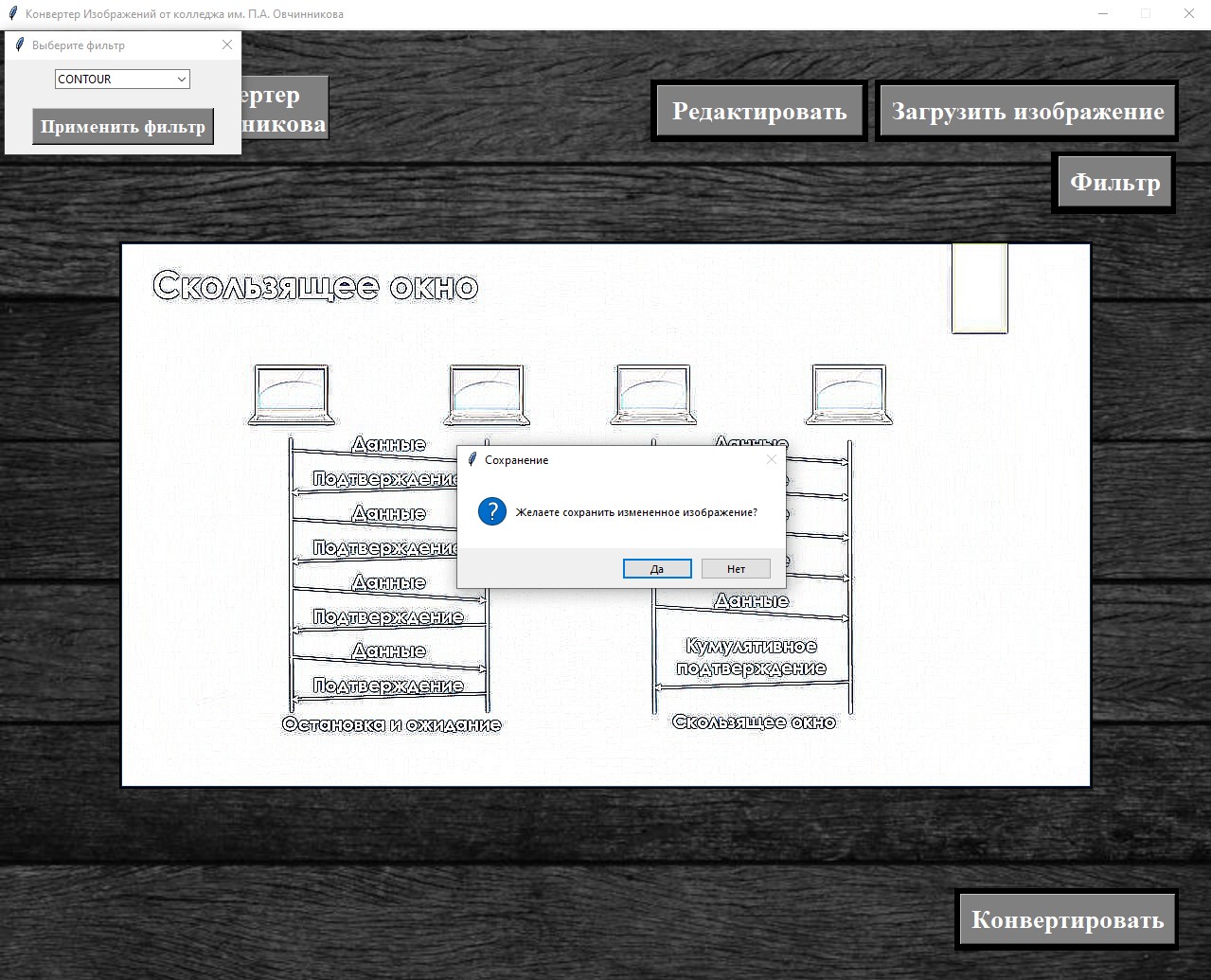


Рисунок . Наложение фильтра