**Минобрнауки России**

**Юго-Западный государственный университет**

Кафедра программной инженерии

Лабораторная работа №1

по дисциплине «Компьютерная графика»

«Введение в компьютерную графику»

Выполнили: студенты группы ПО-32б

Чаплыгин Т.А.  
Чеховская Ю.М.

Проверил: ст. преподаватель

Ефремов В.В.

Курск 2025 г.

Цель работы – освоение базовых навыков обработки растровых изображений с помощью алгоритмических языков программирования.

**ЗАДАНИЕ**

1. Сфотографировать что-нибудь симпатичное на свой смартфон

или цифровой фотоаппарат. Убедиться, что формат фотографии читается в системе программирования.

2. Разработать алгоритм и программу для обработки графической

информации по варианту. Программа включает в себя:

- чтение изображения из файла;

- добавление точек заданного цвета в заданную позицию изображения;

- сохранение созданного изображения в файл на диске.

3. Обработать фотографию с помощью разработанной программы.

4. Найти изменения в результате и описать цвет точек.

5. Ответить на контрольные вопросы.

6. Оформить отчёт.

Вариант 11  
Разместить точку цвета (64,255,255) в верхний левый угол,

точку цвета (64,127,64) в правый верхний угол, точку цвета (127,64,64)

в центр нижней строки.

Исходное изображение (рис.1) получаем с помощью камеры телефона.



Рис. 1 Изображение, полученное с камеры телефона

Изучаем свойства файла (рис. 2) и выясняем, что формат файла

поддерживается системой программирования

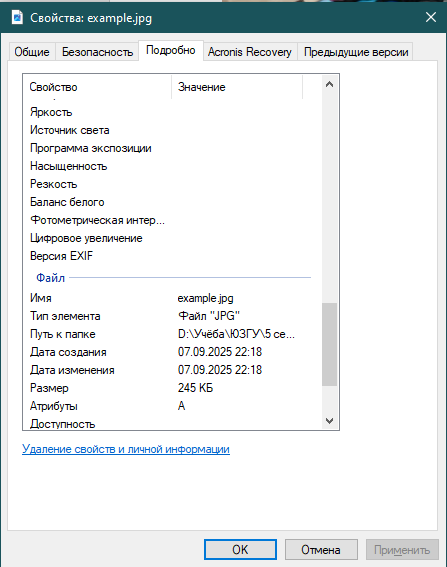


Рис. 2 Свойства исходного изображения

Разработку программы начинаем с разработки формы.

На форме (рис. 3) размещаем:

– три кнопки — для чтения, обработки и записи файла;

– объект Canvas для хранения графического объекта.

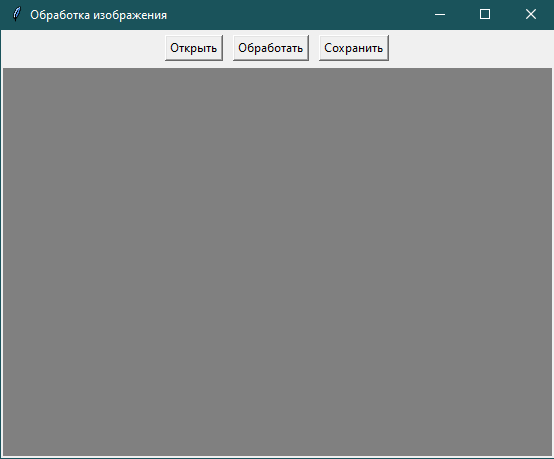


Рис. 3. Разработка формы для программы

Текст программы на языке Python

import tkinter as tk

from tkinter import filedialog

from PIL import Image, ImageTk

class ImageEditorApp:

def \_\_init\_\_(self, root):

self.root = root

self.root.title("Обработка изображения")

frame\_buttons = tk.Frame(root)

frame\_buttons.pack(side=tk.TOP, pady=5)

btn\_open = tk.Button(frame\_buttons, text="Открыть", command=self.open\_image)

btn\_open.pack(side=tk.LEFT, padx=5)

btn\_process = tk.Button(frame\_buttons, text="Обработать", command=self.process\_image)

btn\_process.pack(side=tk.LEFT, padx=5)

btn\_save = tk.Button(frame\_buttons, text="Сохранить", command=self.save\_image)

btn\_save.pack(side=tk.LEFT, padx=5)

self.canvas = tk.Canvas(root, bg="gray")

self.canvas.pack(fill=tk.BOTH, expand=True)

self.image = None

self.photo = None

self.file\_path = None

def open\_image(self):

file\_path = filedialog.askopenfilename(

filetypes=[("JPEG files", "\*.jpg;\*.jpeg")]

)

if file\_path:

self.file\_path = file\_path

self.image = Image.open(file\_path).convert("RGB")

self.show\_image(self.image)

def show\_image(self, img):

self.photo = ImageTk.PhotoImage(img)

self.canvas.delete("all")

self.canvas.create\_image(0, 0, anchor="nw", image=self.photo)

def process\_image(self):

if self.image:

img = self.image.copy()

width, height = img.size

pixels = img.load()

pixels[0, 0] = (64, 255, 255)

pixels[width - 1, 0] = (64, 127, 64)

pixels[width // 2, height - 1] = (127, 64, 64)

self.image = img

self.show\_image(img)

def save\_image(self):

if self.image:

file\_path = filedialog.asksaveasfilename(

defaultextension=".png",

filetypes=[

("PNG files", "\*.png"),

("PBM files", "\*.pbm")

]

)

if file\_path:

if file\_path.lower().endswith(".pbm"):

bw\_image = self.image.convert("1")

bw\_image.save(file\_path)

else:

self.image.save(file\_path, "PNG")

if \_\_name\_\_ == "\_\_main\_\_":

root = tk.Tk()

app = ImageEditorApp(root)

root.geometry("1300x1000")

root.mainloop()

Запускаем программу и выбираем файл и открываем его

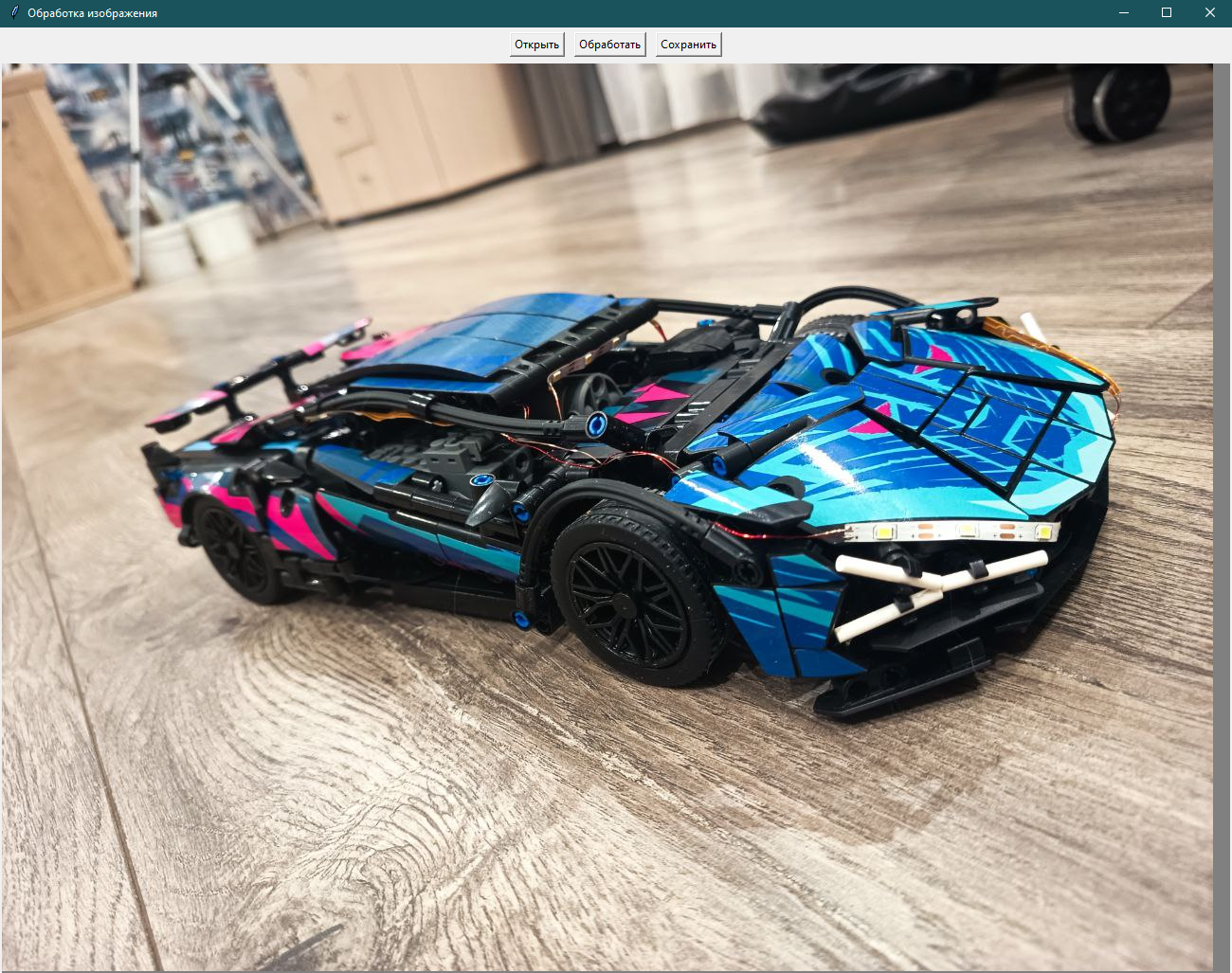


Рис. 4. Выбрали фото, и программа открыла её

Затем нажимаем кнопку «Обработать» и на изображении появляются едва различимые точки

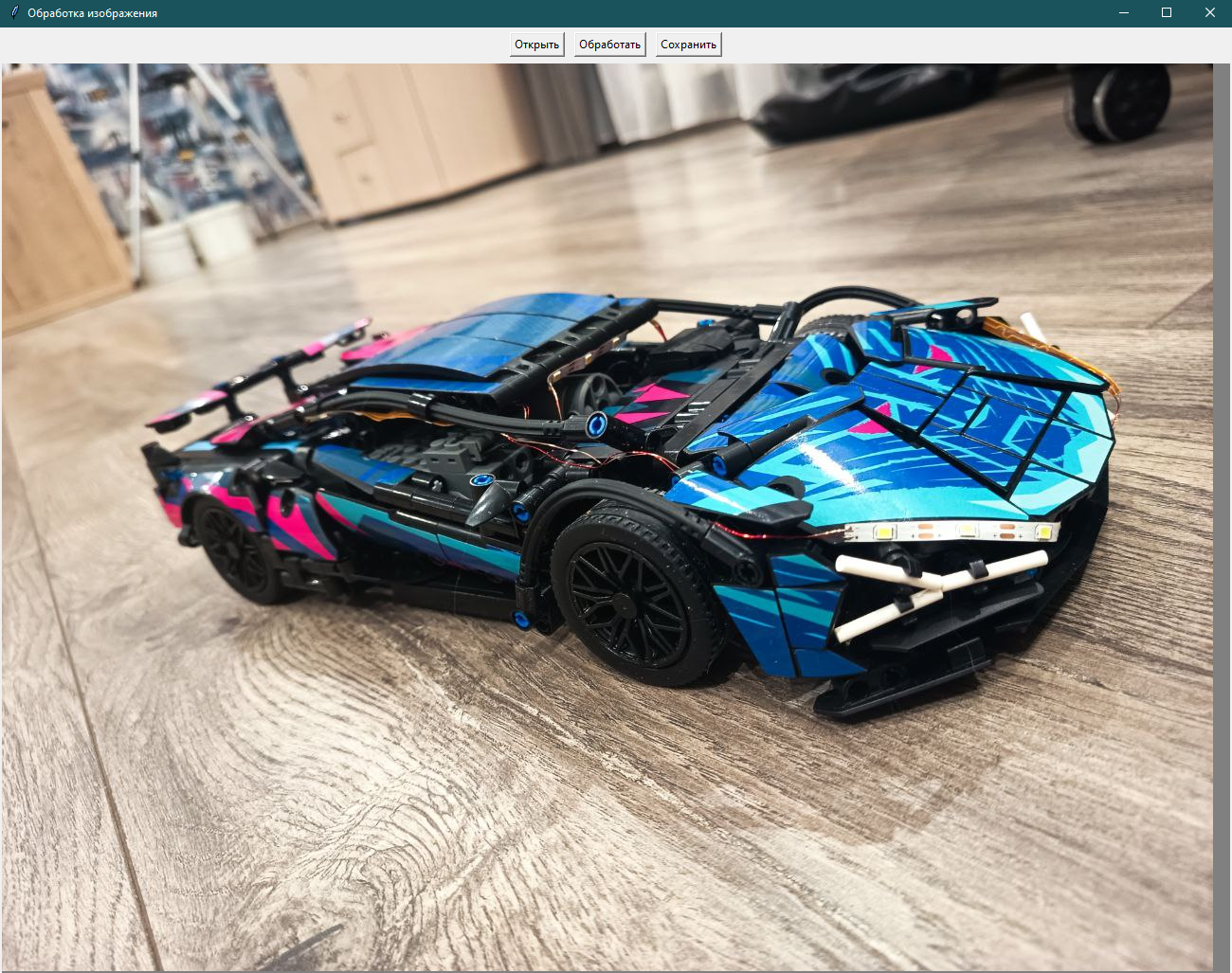


Рис. 5. Изображение после обработки

Затем сохраняем изображение и открываем его через Paint, чтобы можно было различить искомые пиксели. В левом верхнем углу видим точку с цветом «Электрик» (64, 255, 255)

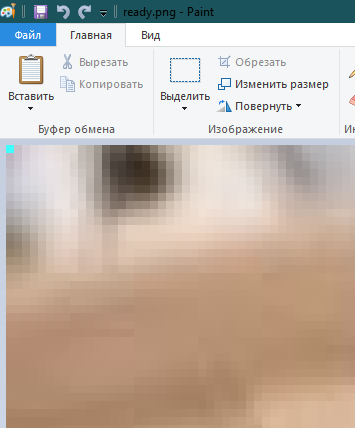


Рис. 6. Первая точка

Затем находим в правом верхнем углу вторую точку с цветом «Зеленый папоротник»

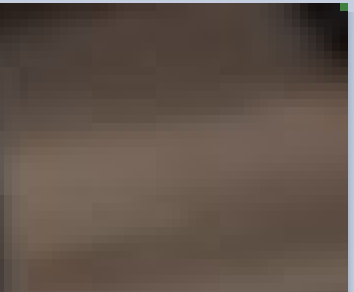


Рис. 7 Вторая точка

И последняя третья точка находится посередине изображения внизу с цветом «Известковая глина»

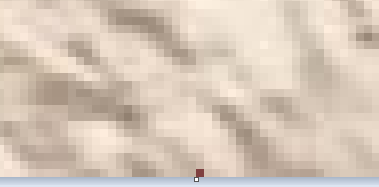


Рис. 8 Третья точка

Контрольные вопросы:

1. Что такое графика?

Графика - это способ представления информации с помощью изображений, линий, фигур и цветов.

2. Что такое оптическое изображение?

Оптическое изображение - это видимое отражение или проекция предмета, получаемое с помощью линз, зеркал или других оптических систем.

3. Что такое сцена в компьютерной графике?

Сцена - это набор объектов, их свойств (геометрия, текстуры, материалы) и источников света, из которых формируется изображение.

4. Что такое визуализация?

Визуализация - процесс преобразования данных или моделей в наглядное изображение.

5. Зачем визуализировать то, что нереально воплотить физически?

Чтобы исследовать теоретические модели, прогнозировать поведение систем, создавать искусство, обучающие материалы и спецэффекты.

6. Что такое рендеринг?

Рендеринг - это вычислительный процесс построения изображения из 3D-сцены с учётом геометрии, освещения и материалов.

7. Что понимается под 3D-графикой на 2D-мониторе?

Это отображение трёхмерных моделей с использованием проекций и перспективы на плоский экран.

8. Как получить движущееся изображение?

Путём последовательного показа серии кадров (анимации) с достаточной частотой (обычно 24–60 кадров в секунду).

9. Как компьютерная графика различается по назначению?

Научная и техническая (визуализация данных, моделирование)

Деловая (диаграммы, отчёты)

Компьютерное искусство и дизайн

Игровая и развлекательная

Обучающая и образовательная

10. Каковы особенности векторной и растровой графики?

Векторная: основана на математических формулах, не теряет качества при масштабировании, подходит для схем, чертежей, логотипов.

Растровая: состоит из пикселей, теряет качество при увеличении, лучше для фотографий и живописи.

11. Как отобразить 4D-объекты?

Через их проекции или сечения в трёхмерное пространство, а затем - в двумерное изображение.

12. Как кодируется цвет?

Цвет кодируется числовыми значениями в цветовых моделях (RGB, CMYK, HSV и др.), где каждая компонента описывает интенсивность определённого цвета.

13. Чем стереоизображения отличаются от трёхмерных?

Стереоизображение создаёт иллюзию глубины для наблюдателя (два изображения для левого и правого глаза). Трёхмерное изображение - это математическая модель объекта в 3D-пространстве.

14. Что такое фрактал?

Фрактал - это самоподобная структура, в которой части повторяют форму целого на разных масштабах.

15. Что такое разрешение?

Разрешение - это количество точек (пикселей), используемых для построения изображения.

16. В каких единицах измеряется разрешающая способность?

В dpi (dots per inch - точек на дюйм) или в пикселях на единицу длины (ppi).

Ссылка на GitHub https://github.com/Tima2005-rus/ComputerGraphicsLab1