**Министерство науки и высшего образования Российской Федерации**

**ФГАОУ ВО «Севастопольский государственный университет»**

Институт информационных технологий

Кафедра «Информационные системы»

**Отчет по лабораторной работе №5**

по дисциплине «Технологии обработки информации»

Выполнил: студент группы

ИС/б-21-2-о

Мельничук В.В.

Принял:

г. Севастополь

**Лабораторная работа №5**

Тема: Обработка графических данных

Цель работы: разбить исходное изображение на объекты и сохранить каждый выделенный объект в отдельный файл.

Вариант 1

«Мозаика»

Листинг программы:

using System;

using System.Windows.Forms;

namespace WinFormsApp1

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

}

private void OpenButton\_Click(object sender, EventArgs e)

{

// Открываем диалоговое окно выбора файла

DialogResult result = openFileDialog.ShowDialog();

if (result == DialogResult.OK)

{

// Получаем путь выбранного файла

string filePath = openFileDialog.FileName;

try

{

// Загружаем изображение из файла

Image image = Image.FromFile(filePath);

// Отображаем изображение в PictureBox

imagePictureBox.Image = image;

// Включаем кнопку "Process"

processButton.Enabled = true;

}

catch (Exception ex)

{

// В случае ошибки выводим сообщение об ошибке

MessageBox.Show("Ошибка при загрузке изображения: " + ex.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void ProcessButton\_Click(object sender, EventArgs e)

{

// Получаем текущее отображаемое изображение

Image currentImage = imagePictureBox.Image;

if (currentImage != null)

{

try

{

// Показываем диалог выбора пути сохранения

using (var folderBrowserDialog = new FolderBrowserDialog())

{

if (folderBrowserDialog.ShowDialog() == DialogResult.OK)

{

string savePath = folderBrowserDialog.SelectedPath;

// Разбиваем исходное изображение на объекты

List<Image> objects = SplitImageIntoObjects(currentImage);

// Сохраняем каждый выделенный объект в отдельный файл с выбранным путем сохранения

for (int i = 0; i < objects.Count; i++)

{

string fileName = $"object\_{i + 1}.png";

string filePath = Path.Combine(savePath, fileName);

objects[i].Save(filePath, System.Drawing.Imaging.ImageFormat.Png);

}

// Выводим сообщение об успешном сохранении объектов

MessageBox.Show($"Объекты успешно сохранены в выбранную папку.", "Информация", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

}

catch (Exception ex)

{

// В случае ошибки выводим сообщение об ошибке

MessageBox.Show("Ошибка при обработке изображения: " + ex.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

{

// Если нет отображаемого изображения, выводим предупреждение

MessageBox.Show("Выберите изображение перед выполнением обработки.", "Предупреждение", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private List<Image> SplitImageIntoObjects(Image image)

{

List<Image> objects = new List<Image>();

// Ваш код для разбиения исходного изображения на объекты

// Пример: разбиение на объекты по цвету фона

// Предполагается, что фон имеет белый цвет (RGB: 255, 255, 255)

Bitmap bitmap = new Bitmap(image);

int width = bitmap.Width;

int height = bitmap.Height;

Color backgroundColor = Color.FromArgb(255, 255, 255);

List<Point> objectPixels = new List<Point>();

for (int y = 0; y < height; y++)

{

for (int x = 0; x < width; x++)

{

Color pixelColor = bitmap.GetPixel(x, y);

// Если текущий пиксель не является фоновым, добавляем его в список точек объекта

if (pixelColor != backgroundColor)

{

objectPixels.Add(new Point(x, y));

}

else if (objectPixels.Count > 0)

{

// Если текущий пиксель является фоновым, но перед этим были точки объекта,

// создаем новое изображение объекта и добавляем его в список объектов

Image objectImage = CropImage(bitmap, objectPixels);

objects.Add(objectImage);

objectPixels.Clear();

}

}

}

// Если в конце были точки объекта, создаем и добавляем последний объект

if (objectPixels.Count > 0)

{

Image objectImage = CropImage(bitmap, objectPixels);

objects.Add(objectImage);

}

return objects;

}

private Image CropImage(Bitmap source, List<Point> objectPixels)

{

// Вычисляем границы объекта

int minX = int.MaxValue;

int minY = int.MaxValue;

int maxX = int.MinValue;

int maxY = int.MinValue;

foreach (Point pixel in objectPixels)

{

if (pixel.X < minX)

minX = pixel.X;

if (pixel.X > maxX)

maxX = pixel.X;

if (pixel.Y < minY)

minY = pixel.Y;

if (pixel.Y > maxY)

maxY = pixel.Y;

}

// Вырезаем объект из исходного изображения

int width = maxX - minX + 1;

int height = maxY - minY + 1;

Rectangle cropRect = new Rectangle(minX, minY, width, height);

Bitmap croppedImage = source.Clone(cropRect, source.PixelFormat);

return croppedImage;

}

private void MainForm\_Load(object sender, EventArgs e)

{

// Логика загрузки формы

}

}

}

Результат работы программы:

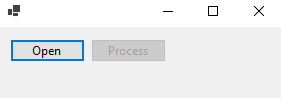


Рисунок 1 – Исходное окно

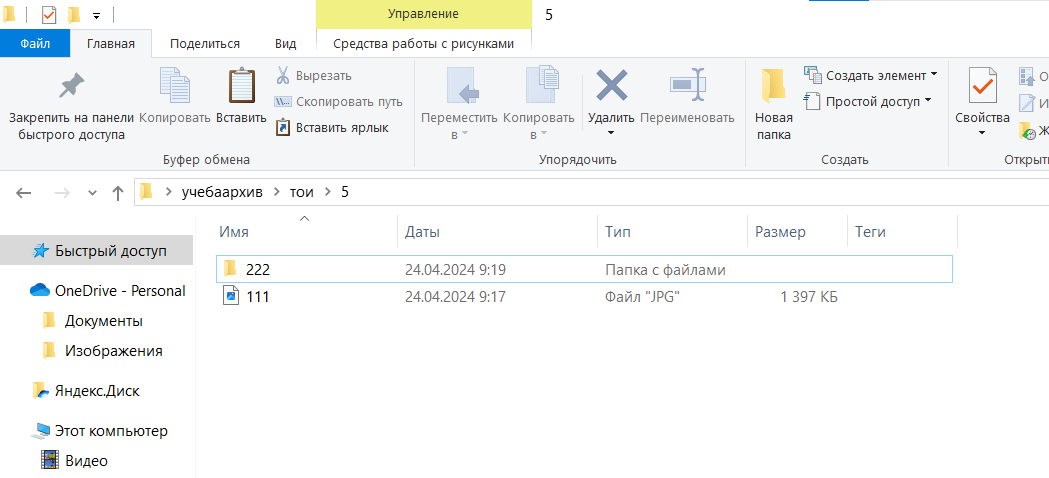


Рисунок 2 – Работа кнопки «Открыть»

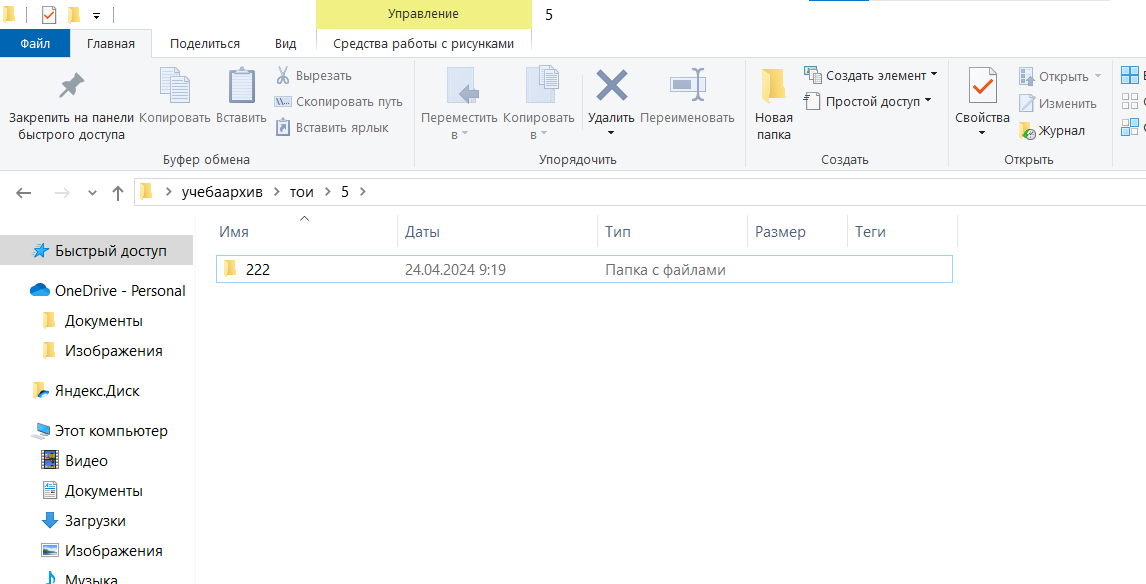


Рисунок 3 – Работа кнопки «Процесс»

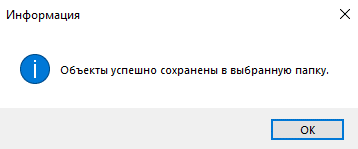


Рисунок 4 – Сохранение новых файлов



Рисунок 5 – Входные данные



Рисунок 6 – Результат работы программы

Вывод: в ходе выполнения работы разбили исходное изображение на объекты и сохранили каждый выделенный объект в отдельный файл.