Лабораторная работа №3

Тема: Оконное приложение «Копирайтер» и «Игровой стол с картами».

Цель работы: Научится разрабатывать Win-приложения на Visual C#.

Ход работы

1) Разработать приложение, позволяющее открывать указанное изображение (или все изображения в указанной директории) и сохранять новую версию изображения с добавленным текстом копирайта (рис.1).

Меню:	File	Operations	Settings	Help
	Open...	Add copyright	Copyright text...	
	Open directory...	Save image...	Copyright directory...	
	_____	Batch mode		
	Exit			

Главное окно приложения «Копирайтер», особенности:

1) Пункты меню Open… – открыть файл изображения и Open directory… – открыть директорию. При этом в левой панели должна отобразиться галерея уменьшенных изображений (всех найденных в директории). При выделении одной из картинок в галерее она должна в полном размере отобразиться в центральной области окна.

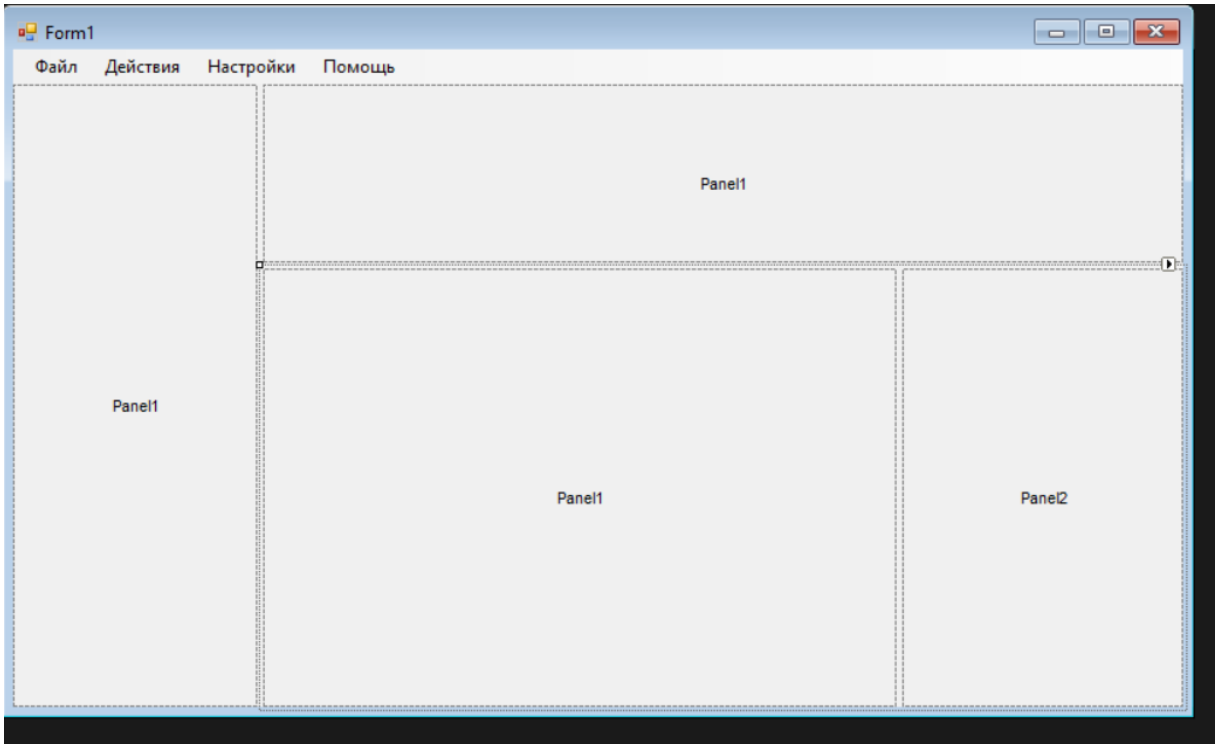
2) При нажатии на кнопку «Add Copyright» на картинку долен наложиться текст копирайта, а также в галерее на уменьшенном изображении должен появиться значок, сигнализирующий о том, что в данном файле был проставлен копирайт (рис.1). Кроме того, в таблице, в верхней части окна, должна добавиться строка с информацией о только что измененном файле.

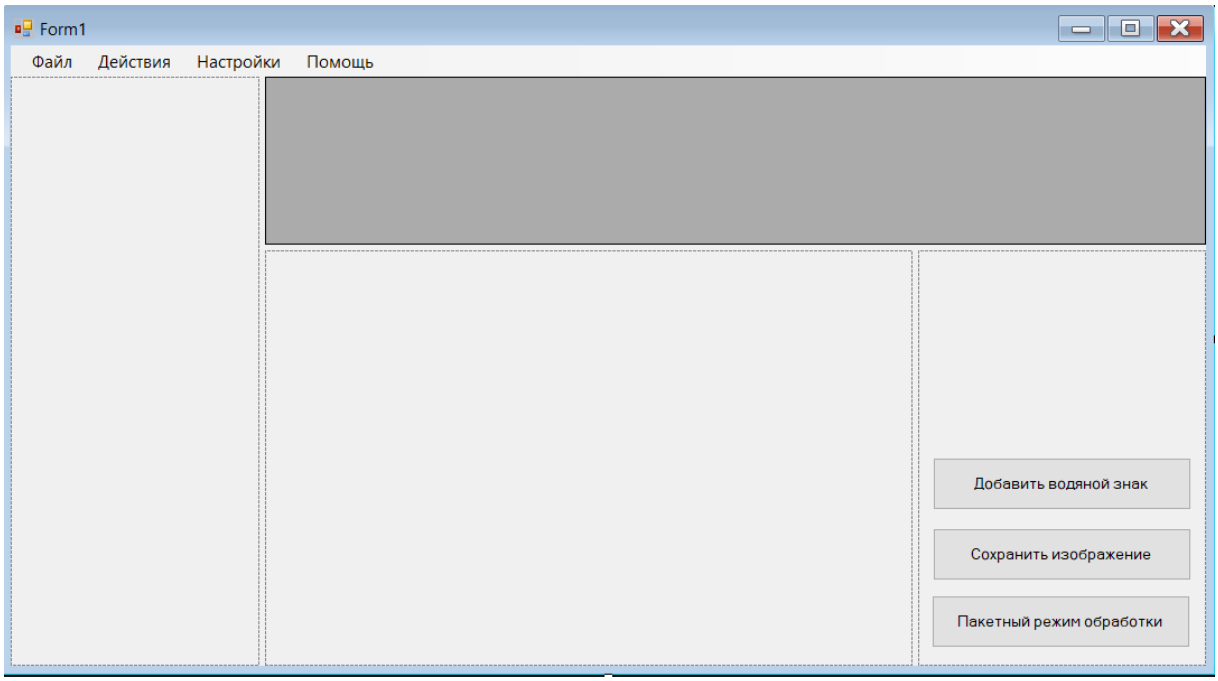
3) Кнопка «Save image…» служит для того, чтобы сохранить текущую картинку (с копирайтом) в новом файле, имя которого должно задаться пользователем.

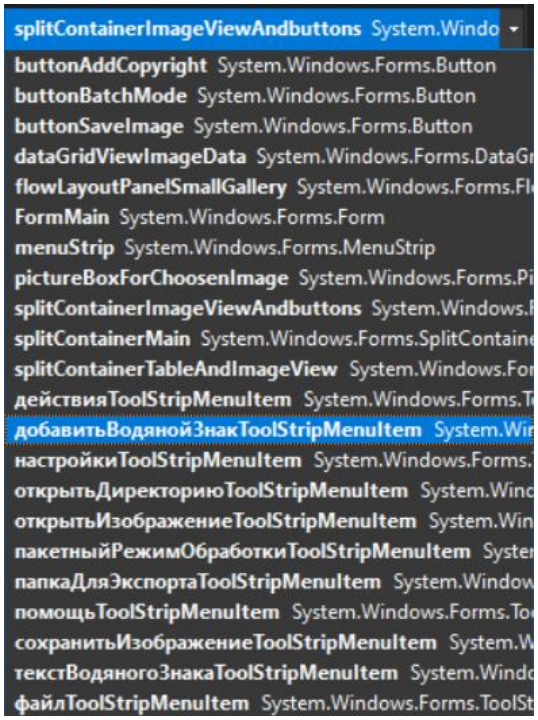
4) Кнопка «Batch mode» запускает «пакетный» режим обработки картинок – все картинки из галереи автоматически обрабатываются и сохраняются в свои новые версии с копирайтом, в директорию, которую можно указать с помощью пункта меню «Copyright directory…». С помощью пункта

меню «Copyright text…» можно задать сам текст копирайта, который нужно накладывать на картинки.

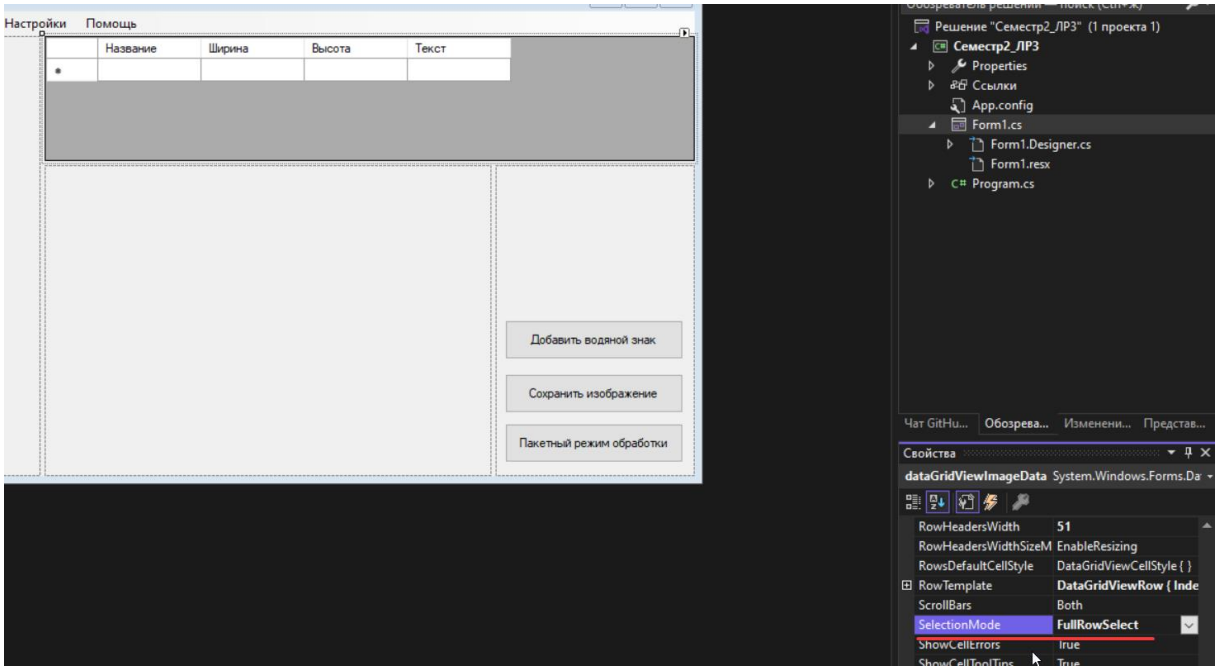
5) Файлы из галереи можно удалять с помощью клавиши Del на выделенной картинке (чтобы соответствующий файл не участвовал в пакетной обработке).

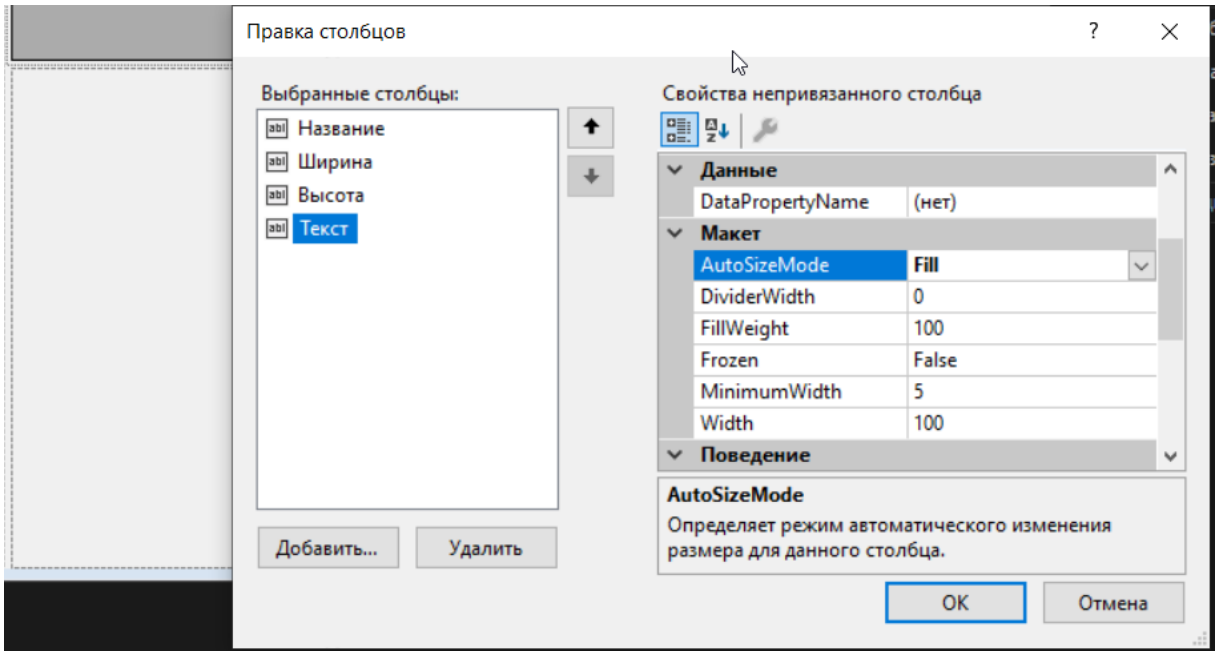


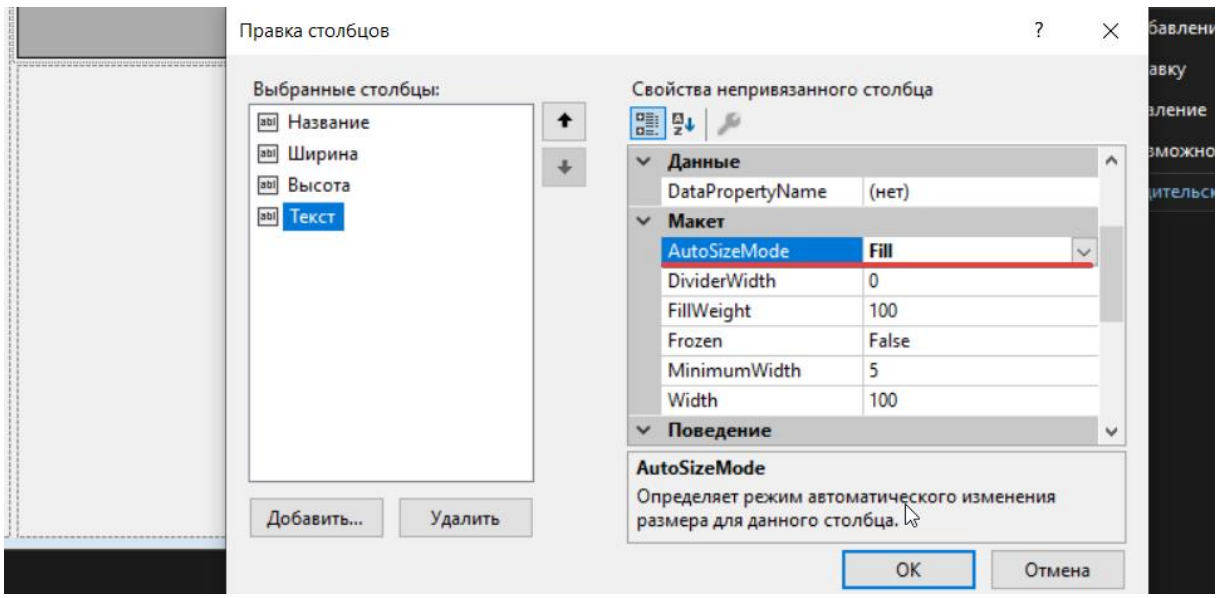


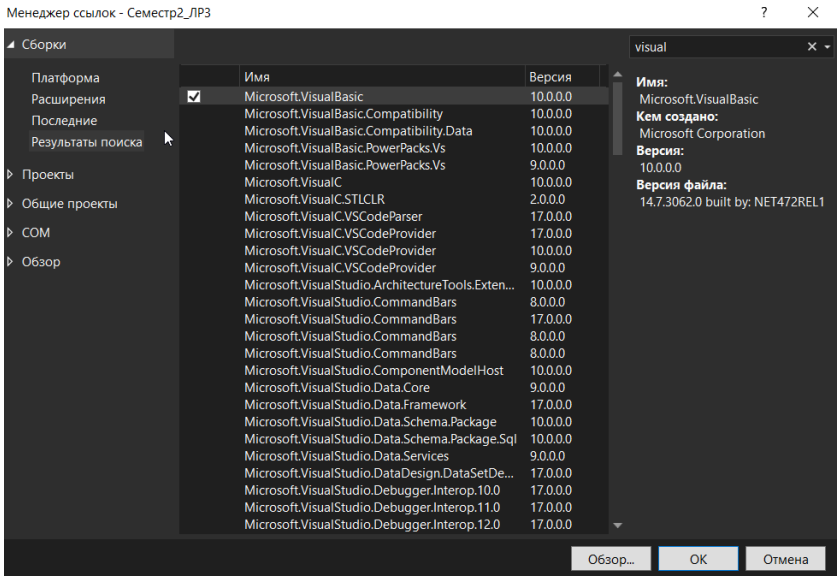

splitContainerImageViewAndbuttons System.Windo
buttonAddCopyright System.Windows.Forms.Button
buttonBatchMode System.Windows.Forms.Button
buttonSaveImage System.Windows.Forms.Button
dataGridViewImageData System.Windows.Forms.DataGr
flowLayoutPanelSmallGallery System.Windows.Forms.Fl
FormMain System.Windows.Forms.Form
menuStrip System.Windows.Forms.MenuStrip
pictureBoxForChoosenImage System.Windows.Forms.Pi
splitContainerImageViewAndbuttons System.Windows.F
splitContainerMain System.Windows.Forms.SplitContain
splitContainerTableAndImageView System.Windows.For
действияToolStripMenuItem System.Windows.Forms.T
добавитьВодянойЗнакToolStripMenuItem System.Wir
настройкиToolStripMenuItem System.Windows.Forms.
открытьДиректориюToolStripMenuItem System.Windo
открытьИзображениеToolStripMenuItem System.Win
пакетныйРежимОбработкиToolStripMenuItem Syste
папкаДляЭкспортаToolStripMenuItem System.Window
помощьToolStripMenuItem System.Windows.Forms.To
сохранитьИзображениеToolStripMenuItem System.W
текстВодяногоЗнакаToolStripMenuItem System.Windo
файлToolStripMenuItem System.Windows.Forms.ToolSt

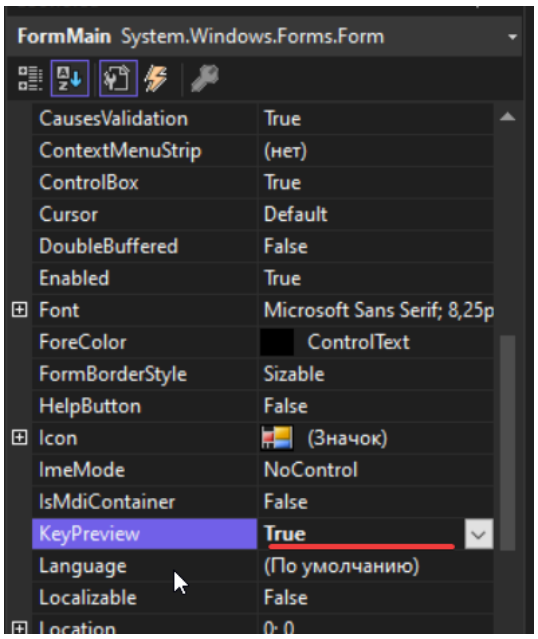

ContextMenuStrip
Удалить
Просмотр
Наложить водяной знак
Вводить здесь
Вводить здесь

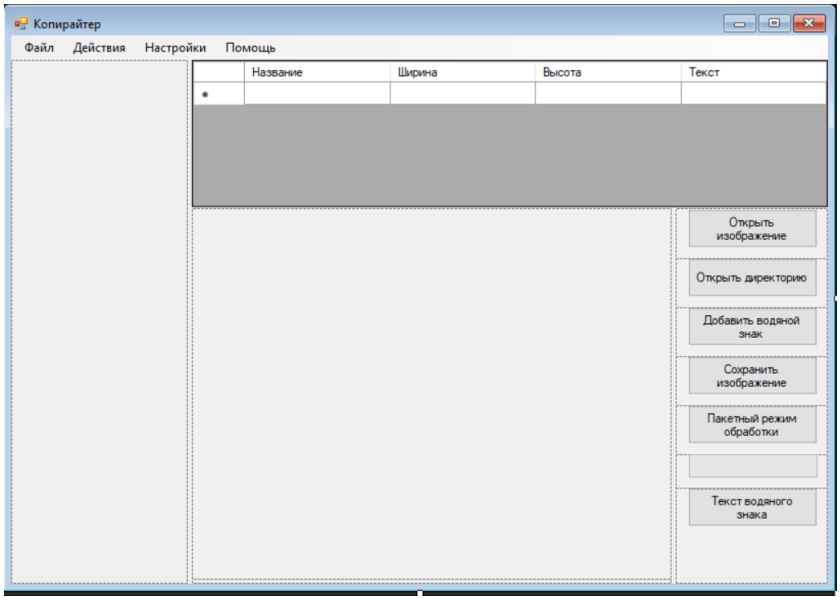

Настройки
Помощь
Название
Ширина
Высота
Текст
Добавить водяной знак
Сохранить изображение
Пакетный режим обработки
Решение "Семестр2_ЛР3" (1 проекта 1)
Семестр2_ЛР3
Properties
Ссылки
App.config
Form1.cs
Form1.Designer.cs
Form1.resx
Program.cs
Свойства
dataGridViewImageData System.Windows.Forms.Da
RowHeadersWidth 51
RowHeadersWidthSizeM EnableResizing
RowsDefaultCellStyle DataGridViewCellStyle { }
RowTemplate DataGridViewRow { Inde
ScrollBars Both
SelectionMode FullRowSelect


Правка столбцов
Выбранные столбцы:
Название
Ширина
Высота
Текст
Добавить...
Удалить
Свойства непривязанного столбца
Данные
DataPropertyName (нет)
Макет
AutoSizeMode Fill
DividerWidth 0
FillWeight 100
Frozen False
MinimumWidth 5
Width 100
Поведение
AutoSizeMode
Определяет режим автоматического изменения размера для данного столбца.
OK
Отмена


Правка столбцов
Выбранные столбцы:
Название
Ширина
Высота
Текст
Добавить...
Удалить
Свойства непривязанного столбца
Данные
DataPropertyName (нет)
Макет
AutoSizeMode Fill
DividerWidth 0
FillWeight 100
Frozen False
MinimumWidth 5
Width 100
Поведение
AutoSizeMode
Определяет режим автоматического изменения размера для данного столбца.
OK
Отмена


Менеджер ссылок - Семестр2_ЛР3
Сборки
Платформа
Расширения
Последние
Результаты поиска
Проекты
Общие проекты
COM
Обзор
visual
Имя
Версия
Microsoft.VisualBasic 10.0.0.0
Microsoft.VisualBasic.Compatibility 10.0.0.0
Microsoft.VisualBasic.Compatibility.Data 10.0.0.0
Microsoft.VisualBasic.PowerPacks.Vs 10.0.0.0
Microsoft.VisualBasic.PowerPacks.Vs 9.0.0.0
Microsoft.VisualC 10.0.0.0
Microsoft.VisualC.STLCLR 2.0.0.0
Microsoft.VisualC.VSCodeParser 17.0.0.0
Microsoft.VisualC.VSCodeProvider 17.0.0.0
Microsoft.VisualC.VSCodeProvider 10.0.0.0
Microsoft.VisualC.VSCodeProvider 9.0.0.0
Microsoft.VisualStudio.ArchitectureTools.Exten... 10.0.0.0
Microsoft.VisualStudio.CommandBars 8.0.0.0
Microsoft.VisualStudio.CommandBars 17.0.0.0
Microsoft.VisualStudio.CommandBars 8.0.0.0
Microsoft.VisualStudio.CommandBars 8.0.0.0
Microsoft.VisualStudio.ComponentModelHost 10.0.0.0
Microsoft.VisualStudio.Data.Core 9.0.0.0
Microsoft.VisualStudio.Data.Framework 17.0.0.0
Microsoft.VisualStudio.Data.Schema.Package 10.0.0.0
Microsoft.VisualStudio.Data.Schema.Package.Sql 10.0.0.0
Microsoft.VisualStudio.Data.Services 9.0.0.0
Microsoft.VisualStudio.DataDesign.DataSetDe... 17.0.0.0
Microsoft.VisualStudio.Debugger.Interop.10.0 17.0.0.0
Microsoft.VisualStudio.Debugger.Interop.11.0 17.0.0.0
Microsoft.VisualStudio.Debugger.Interop.12.0 17.0.0.0
Имя:
Microsoft.VisualBasic
Кем создано:
Microsoft Corporation
Версия:
10.0.0.0
Версия файла:
14.7.3062.0 built by: NET472REL1
Обзор...
OK
Отмена


FormMain System.Windows.Forms.Form
CausesValidation True
ContextMenuStrip (нет)
ControlBox True
Cursor Default
DoubleBuffered False
Enabled True
Font Microsoft Sans Serif; 8,25p
ForeColor ControlText
FormBorderStyle Sizable
HelpButton False
Icon (Значок)
ImeMode NoControl
IsMdiContainer False
KeyPreview True
Language (По умолчанию)
Localizable False


Копирайтер
Файл
Действия
Настройки
Помощь
Название
Ширина
Высота
Текст
Открыть изображение
Открыть директорию
Добавить водяной знак
Сохранить изображение
Пакетный режим обработки
Текст водяного знака

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Microsoft.VisualBasic;


namespace Семестр2_ЛР3
{
    public partial class FormMain : Form
    {
        public FormMain()
        {
            InitializeComponent();
        }

        private List<string> imagePaths = new List<string>();
        private string currentImagePath = "";
        private string copyrightText = "(c) Моя прелесть";
        private string exportDirectory = "";
        private Dictionary<string, bool> imageCopyrighted = new Dictionary<string,
bool>();

        private Bitmap AddCheckIcon(Bitmap image, string checkIconPath)
        {
            if (!File.Exists(checkIconPath)) return image;

            using (Graphics g = Graphics.FromImage(image))
            using (Image checkIcon = Image.FromFile(checkIconPath))
            {
                int iconSize = Math.Max(24, image.Width / 6);
                int padding = 5;
                int x = image.Width - iconSize - padding;
                int y = image.Height - iconSize - padding;

                g.DrawImage(checkIcon, new Rectangle(x, y, iconSize, iconSize));
            }

            return image;
        }

        private Bitmap AddWatermark(Bitmap bitmap, string text)
        {
            using (Graphics g = Graphics.FromImage(bitmap))
            {
                float fontSize = bitmap.Height * 0.1f;
```

```
                using (Font font = new Font("Arial", fontSize, FontStyle.Bold,
GraphicsUnit.Pixel))
                using (SolidBrush brush = new SolidBrush(Color.Red))
                {
                    g.DrawString(text, font, brush, new PointF(10, bitmap.Height -
fontSize - 10));
                }
            }

            return bitmap;
        }

        private bool IsImageAlreadyInGrid(string fileName, string commentPrefix =
null)
        {
            foreach (DataGridViewRow row in dataGridViewImageData.Rows)
            {
                if (row.Cells[0].Value?.ToString() == fileName)
                {
                    if (commentPrefix == null ||
row.Cells[3].Value?.ToString().StartsWith(commentPrefix) == true)
                        return true;
                }
            }
            return false;
        }




        private void открытьИзображениеToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            buttonOpenImage_Click(sender, e);
        }

        private void открытьДиректориюToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            buttonOpenDirectory_Click(sender, e);
        }

        private void LoadGallery()
        {
            flowLayoutPanelSmallGallery.Controls.Clear();
            foreach (var path in imagePaths)
            {
                PictureBox pb = new PictureBox();
                pb.Image = Image.FromFile(path);
                pb.SizeMode = PictureBoxSizeMode.Zoom;
                pb.Width = 150;
                pb.Height = 150;
```

```
                pb.Tag = path;
                pb.Click += Pb_Click;
                flowLayoutPanelSmallGallery.Controls.Add(pb);
                imageCopyrighted[path] = false;
            }

            // Автозагрузка первого изображения
            if (imagePaths.Any())
            {
                currentImagePath = imagePaths[0];
                pictureBoxForChoosenImage.Image = Image.FromFile(currentImagePath);
            }
        }


        private void buttonAddCopyright_Click(object sender, EventArgs e)
        {
            if (string.IsNullOrEmpty(currentImagePath)) return;

            using (Image originalImage = Image.FromFile(currentImagePath))
            using (Bitmap bmp = ResizeIfTooSmall(new Bitmap(originalImage)))
            {
                Bitmap watermarked = AddWatermark(bmp, copyrightText);

                pictureBoxForChoosenImage.Image?.Dispose();
                pictureBoxForChoosenImage.Image = new Bitmap(watermarked);

                // Добавляем галочку
                foreach (PictureBox pbItem in flowLayoutPanelSmallGallery.Controls)
                {
                    if (pbItem.Tag.ToString() == currentImagePath)
                    {
                        string checkIconPath = Path.Combine(Application.StartupPath,
"images", "check.png");
                        using (Image original = Image.FromFile(currentImagePath))
                        using (Bitmap iconImage = new Bitmap(original))
                        {
                            Bitmap updated = AddCheckIcon(iconImage, checkIconPath);
                            pbItem.Image?.Dispose();
                            pbItem.Image = new Bitmap(updated);
                            pbItem.Invalidate();
                        }

                        break;
                    }
                }

                // Обновление таблицы
                string fileName = Path.GetFileName(currentImagePath);
                if (!IsImageAlreadyInGrid(fileName, copyrightText))
                {
                    dataGridViewImageData.Rows.Add(
```

```
                        fileName,
                        watermarked.Width,
                        watermarked.Height,
                        $"{copyrightText} [{DateTime.Now:HH:mm}]"
                    );
                }

                imageCopyrighted[currentImagePath] = true;
            }
        }




        private void buttonSaveImage_Click(object sender, EventArgs e)
        {
            SaveFileDialog sfd = new SaveFileDialog();
            sfd.Filter = "PNG Image|*.png";
            if (sfd.ShowDialog() == DialogResult.OK)
            {
                pictureBoxForChoosenImage.Image.Save(sfd.FileName,
System.Drawing.Imaging.ImageFormat.Jpeg);
            }
        }


        private void buttonBatchMode_Click(object sender, EventArgs e)
        {
            if (string.IsNullOrEmpty(exportDirectory) ||
!Directory.Exists(exportDirectory))
            {
                MessageBox.Show("Укажите папку для экспорта.", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return;
            }

            int total = flowLayoutPanelSmallGallery.Controls.Count;
            int processed = 0;

            progressBarBatch.Minimum = 0;
            progressBarBatch.Maximum = total;
            progressBarBatch.Value = 0;
            progressBarBatch.Visible = true;

            foreach (Control control in flowLayoutPanelSmallGallery.Controls)
            {
                if (control is PictureBox pb && pb.Tag is string filePath)
                {
                    try
                    {
                        if (!imageCopyrighted[filePath])
```

```
                        {
                            using (Image img = Image.FromFile(filePath))
                            using (Bitmap bitmap = ResizeIfTooSmall(new Bitmap(img)))
                            {
                                Bitmap watermarked = AddWatermark(bitmap,
copyrightText);
                                string outputName = Path.Combine(exportDirectory,
Path.GetFileNameWithoutExtension(filePath) + ".png");
                                watermarked.Save(outputName,
System.Drawing.Imaging.ImageFormat.Png);

                                // Добавляем галочку
                                foreach (PictureBox galleryPb in
flowLayoutPanelSmallGallery.Controls)
                                {
                                    if (galleryPb.Tag.ToString() == filePath)
                                    {
                                        string checkIconPath =
Path.Combine(Application.StartupPath, "images", "check.png");
                                        using (Image original =
Image.FromFile(filePath))
                                        using (Bitmap iconImage = new
Bitmap(original))
                                        {
                                            Bitmap updated = AddCheckIcon(iconImage,
checkIconPath);

                                            galleryPb.Image?.Dispose();
                                            galleryPb.Image = new Bitmap(updated);
                                            galleryPb.Invalidate();
                                        }

                                        break;
                                    }
                                }

                                // Обновляем таблицу
                                string fileName = Path.GetFileName(filePath);
                                if (!IsImageAlreadyInGrid(fileName))
                                {
                                    dataGridViewImageData.Rows.Add(
                                        fileName,
                                        watermarked.Width,
                                        watermarked.Height,
                                        $"{copyrightText} [{DateTime.Now:HH:mm}]"
                                    );
                                }

                                imageCopyrighted[filePath] = true;
                            }
                        }
                    }
                    catch (Exception ex)
```

```
                    {
                        MessageBox.Show($"Ошибка при обработке
файла:\n{filePath}\n\n{ex.Message}", "Ошибка", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    }
                }

                processed++;
                progressBarBatch.Value = processed;
                progressBarBatch.Refresh();
            }

            progressBarBatch.Visible = false;

            // Обновление выбранного изображения
            if (!string.IsNullOrEmpty(currentImagePath) &&
imageCopyrighted.ContainsKey(currentImagePath) && imageCopyrighted[currentImagePath])
            {
                string exportedPath = Path.Combine(exportDirectory,
Path.GetFileNameWithoutExtension(currentImagePath) + ".png");
                if (File.Exists(exportedPath))
                {
                    pictureBoxForChoosenImage.Image?.Dispose();
                    pictureBoxForChoosenImage.Image = Image.FromFile(exportedPath);
                }
            }

            MessageBox.Show("Водяные знаки успешно добавлены ко всем изображениям!",
"Готово", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }



        private void текстВодяногоЗнакаToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            buttonCopyrightText_Click(sender, e);
        }

        private void папкаДляЭкспортаToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            FolderBrowserDialog fbd = new FolderBrowserDialog();
            if (fbd.ShowDialog() == DialogResult.OK)
            {
                exportDirectory = fbd.SelectedPath;
            }
        }

        private void flowLayoutPanelSmallGallery_Paint(object sender, PaintEventArgs
e)
        {
```

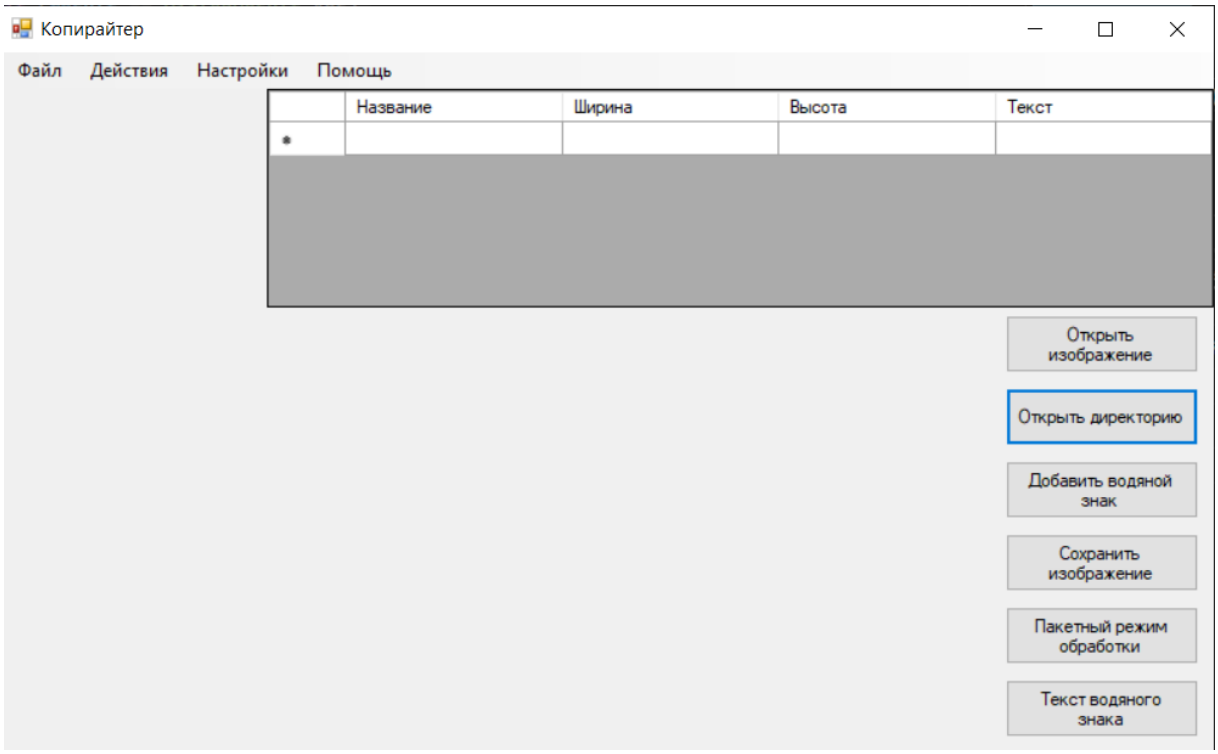
```
        }

        private Bitmap ResizeIfTooSmall(Bitmap original, int minWidth = 400, int
minHeight = 400)
        {
            if (original.Width >= minWidth && original.Height >= minHeight)
                return new Bitmap(original); // Копия без изменений

            float scaleX = (float)minWidth / original.Width;
            float scaleY = (float)minHeight / original.Height;
            float scale = Math.Max(scaleX, scaleY); // сохраняем пропорции

            int newWidth = (int)(original.Width * scale);
            int newHeight = (int)(original.Height * scale);

            Bitmap resized = new Bitmap(newWidth, newHeight);
            using (Graphics g = Graphics.FromImage(resized))
            {
                g.InterpolationMode =
System.Drawing.Drawing2D.InterpolationMode.HighQualityBicubic;
                g.DrawImage(original, 0, 0, newWidth, newHeight);
            }

            return resized;
        }


        private void Pb_Click(object sender, EventArgs e)
        {
            PictureBox pb = sender as PictureBox;
            string originalPath = pb.Tag.ToString();
            currentImagePath = originalPath;

            // Если изображение уже было обработано — подгружаем экспортированную
версию
            if (imageCopyrighted.ContainsKey(originalPath) &&
imageCopyrighted[originalPath] && !string.IsNullOrEmpty(exportDirectory))
            {
                string exportedPath = Path.Combine(exportDirectory,
Path.GetFileNameWithoutExtension(originalPath) + ".png");

                if (File.Exists(exportedPath))
                {
                    pictureBoxForChoosenImage.Image?.Dispose();
                    pictureBoxForChoosenImage.Image = Image.FromFile(exportedPath);
                    return;
                }
            }

            // Иначе подгружаем оригинал
            pictureBoxForChoosenImage.Image?.Dispose();
```

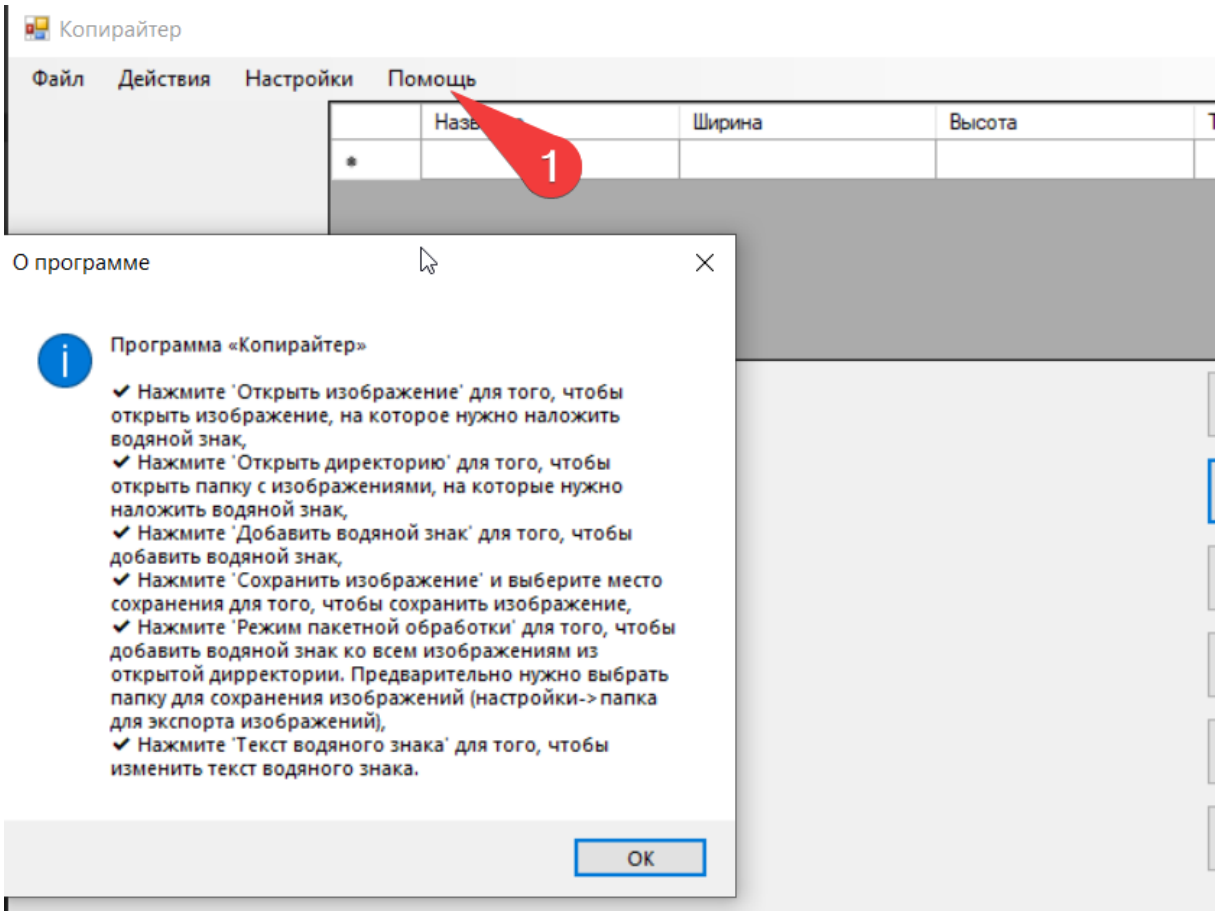
```
            pictureBoxForChoosenImage.Image = Image.FromFile(originalPath);
        }



        private void пакетныйРежимОбработкиToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            buttonBatchMode_Click(sender, e);
        }

        private void splitContainerImageViewAndbuttons_Panel2_Paint(object sender,
PaintEventArgs e)
        {

        }

        private void сохранитьИзображениеToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            buttonSaveImage_Click(sender, e);
        }

        private void добавитьВодянойЗнакToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            buttonAddCopyright_Click(sender, e);
        }

        private void button1_Click(object sender, EventArgs e)
        {

        }

        private void buttonOpenImage_Click(object sender, EventArgs e)
        {
            OpenFileDialog ofd = new OpenFileDialog();
            ofd.Filter = "Image Files|*.jpg;*.jpeg;*.png";
            if (ofd.ShowDialog() == DialogResult.OK)
            {
                imagePaths.Clear();
                imagePaths.Add(ofd.FileName);
                LoadGallery();
            }
        }

        private void buttonOpenDirectory_Click(object sender, EventArgs e)
        {
            FolderBrowserDialog fbd = new FolderBrowserDialog();
            if (fbd.ShowDialog() == DialogResult.OK)
            {
                imagePaths = Directory.GetFiles(fbd.SelectedPath, "*.*")
```

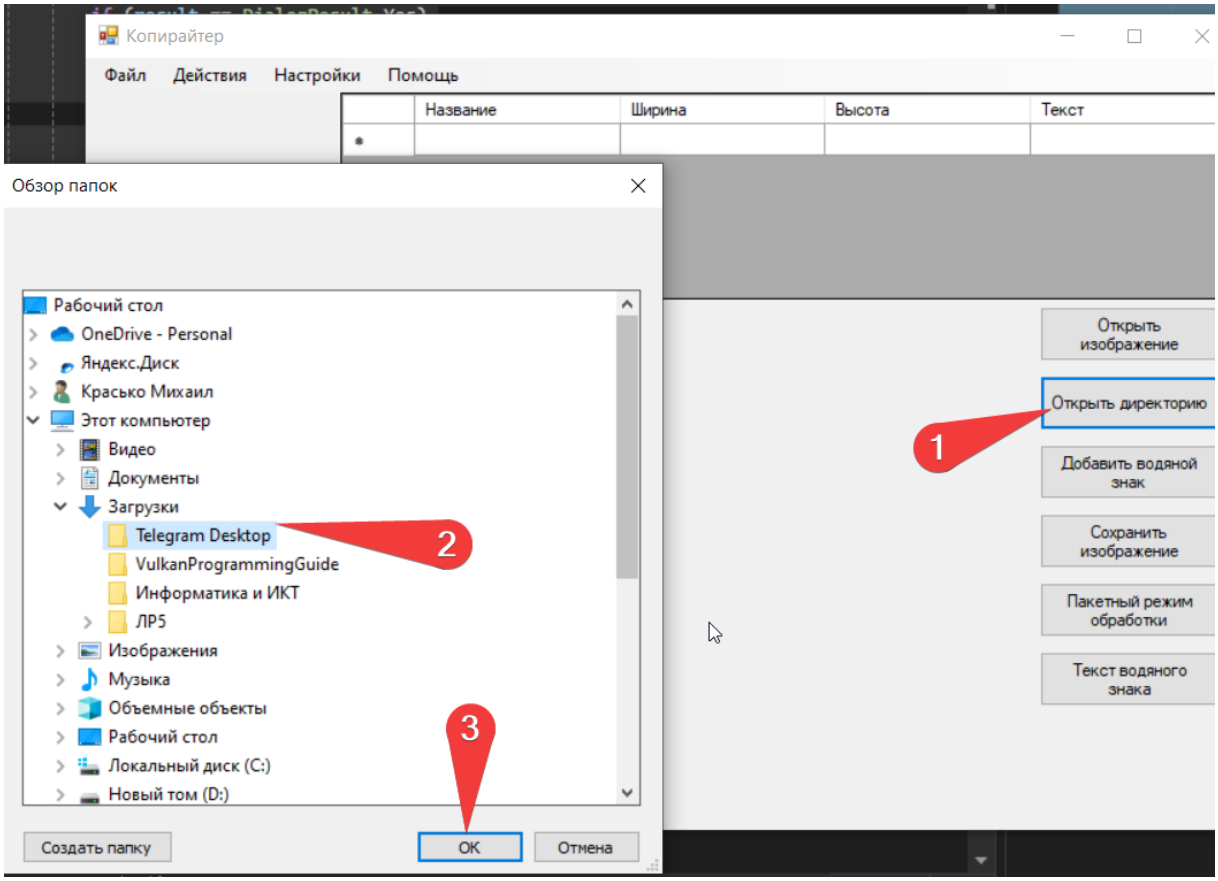
```
                                    .Where(f => f.EndsWith(".jpg") ||
f.EndsWith(".png"))
                                    .ToList();
                LoadGallery();
            }
        }

        private void splitContainerMain_SplitterMoved(object sender,
SplitterEventArgs e)
        {

        }

        private void помощьToolStripMenuItem_Click(object sender, EventArgs e)
        {
            string message =
                "Программа «Копирайтер»\n\n" +
                "✓ Нажмите 'Открыть изображение' для того, чтобы открыть изображение,
на которое нужно наложить водяной знак,\n" +
                "✓ Нажмите 'Открыть директорию' для того, чтобы открыть папку с
изображениями, на которые нужно наложить водяной знак,\n" +
                "✓ Нажмите 'Добавить водяной знак' для того, чтобы добавить водяной
знак,\n" +
                "✓ Нажмите 'Сохранить изображение' и выберите место сохранения для
того, чтобы сохранить изображение,\n" +
                "✓ Нажмите 'Режим пакетной обработки' для того, чтобы добавить
водяной знак ко всем изображениям из открытой дирректории. Предварительно нужно
выбрать папку для сохранения изображений (настройки->папка для экспорта
изображений),\n" +
                "✓ Нажмите 'Текст водяного знака' для того, чтобы изменить текст
водяного знака.";

            MessageBox.Show(message, "О программе", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        }

        private void buttonCopyrightText_Click(object sender, EventArgs e)
        {
            string input = Microsoft.VisualBasic.Interaction.InputBox("Введите текст
водяного знака:", "Водяной знак", copyrightText);
            if (!string.IsNullOrEmpty(input))
                copyrightText = input;
        }

        private void FormMain_Load(object sender, EventArgs e)
        {

        }

        private void FormMain_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyCode == Keys.Delete && !string.IsNullOrEmpty(currentImagePath))
```

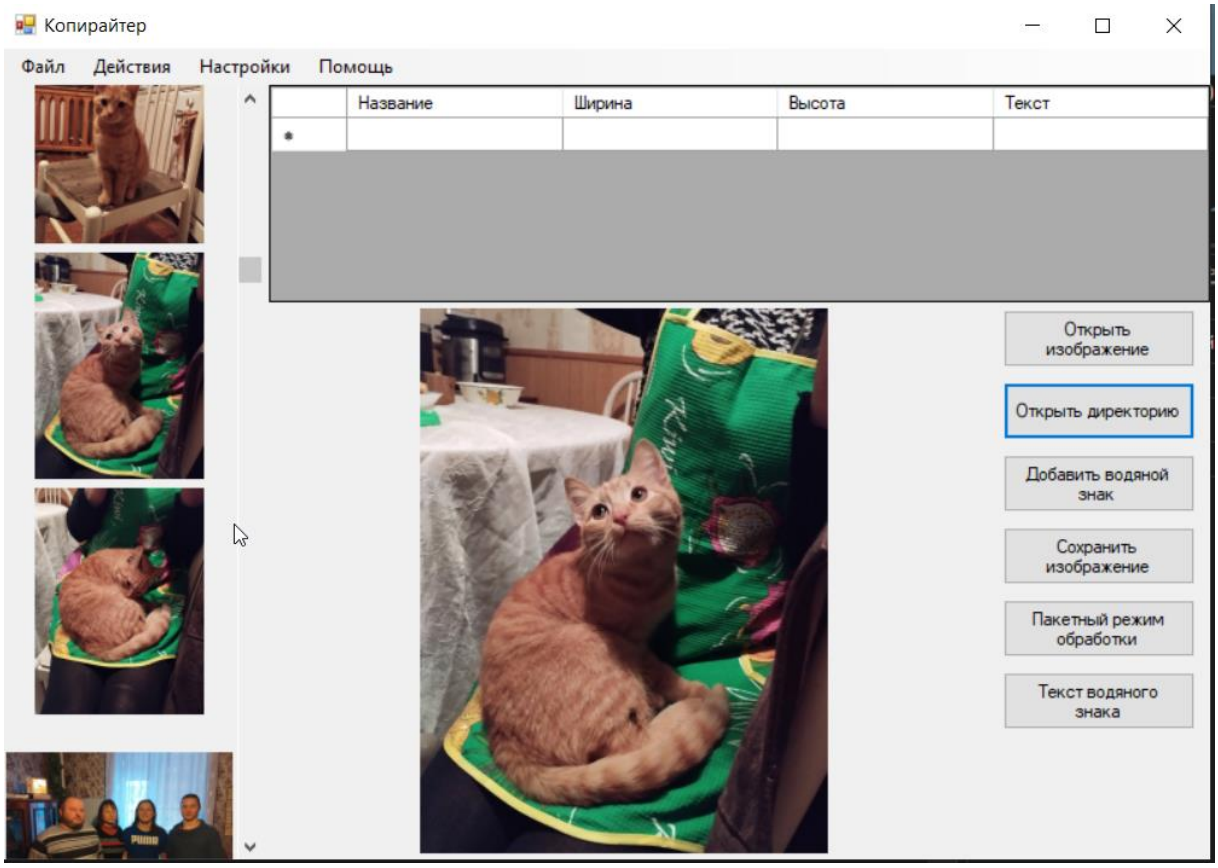
```
            {
                DialogResult result = MessageBox.Show("Удалить изображение из
галереи?", "Подтверждение", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
                if (result == DialogResult.Yes)
                {
                    // Удаление из списка путей
                    imagePaths.Remove(currentImagePath);

                    // Удаление из словаря копирайтов
                    if (imageCopyrighted.ContainsKey(currentImagePath))
                        imageCopyrighted.Remove(currentImagePath);

                    // Удаление из галереи
                    PictureBox targetPB = null;
                    foreach (PictureBox pb in flowLayoutPanelSmallGallery.Controls)
                    {
                        if (pb.Tag.ToString() == currentImagePath)
                        {
                            targetPB = pb;
                            break;
                        }
                    }
                    if (targetPB != null)
                        flowLayoutPanelSmallGallery.Controls.Remove(targetPB);

                    // Очистка изображения
                    pictureBoxForChoosenImage.Image = null;
                    currentImagePath = "";
                }
            }
        }
    }
}
```

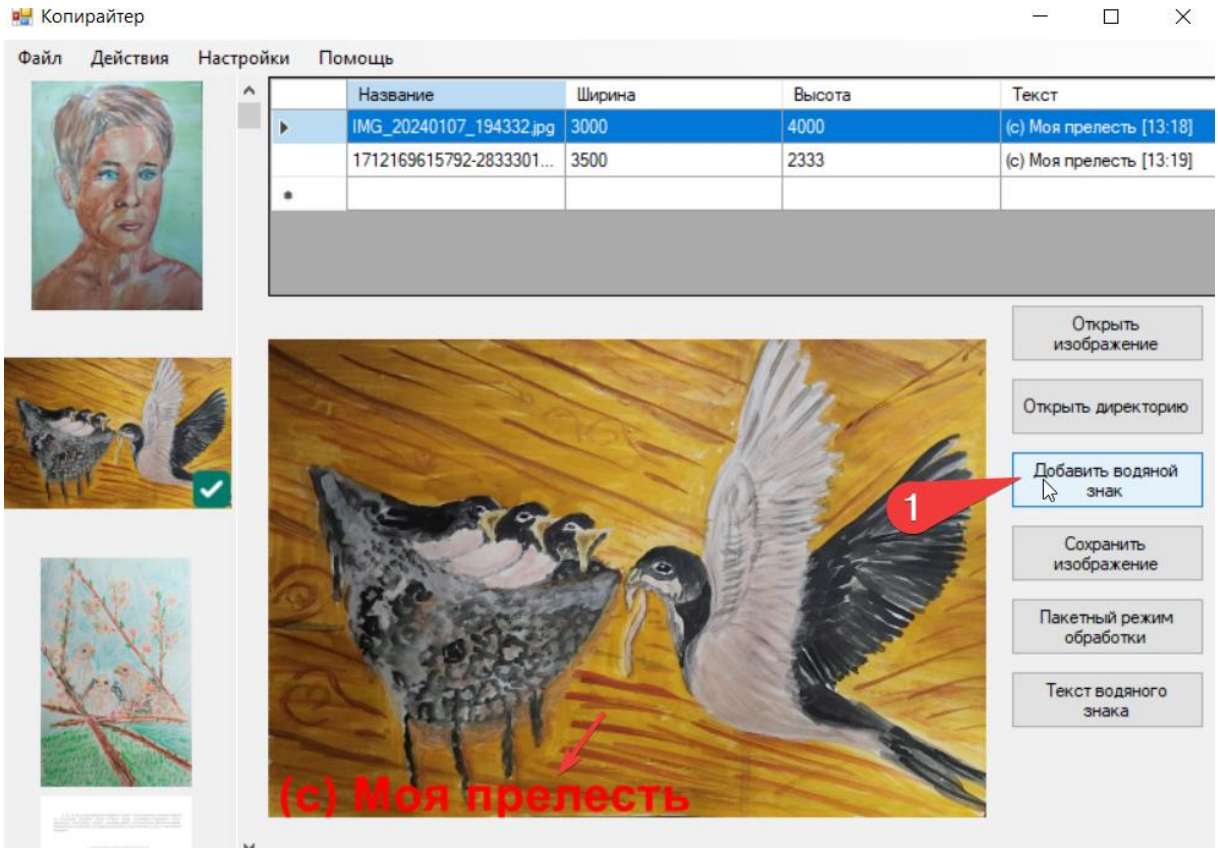
Тестирование:

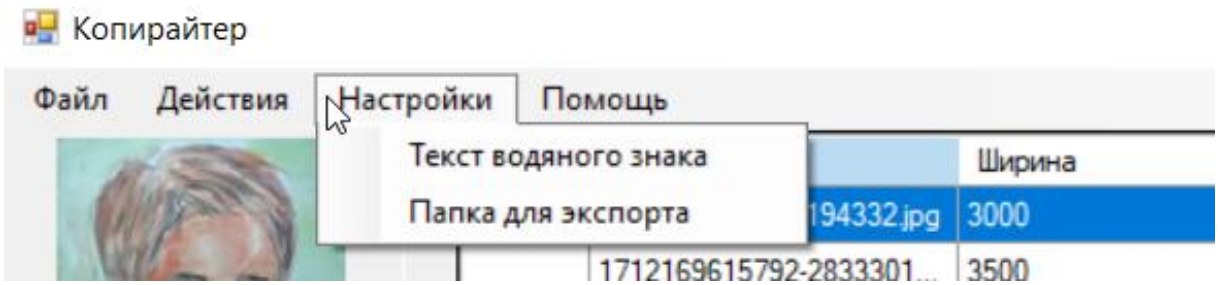


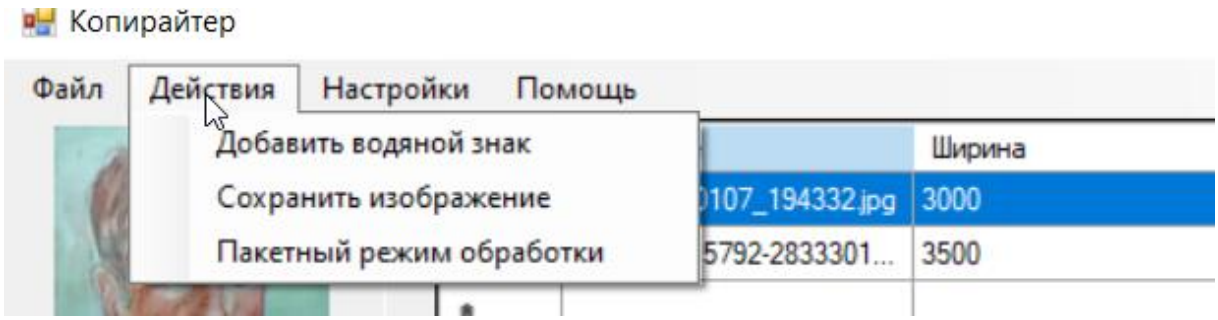


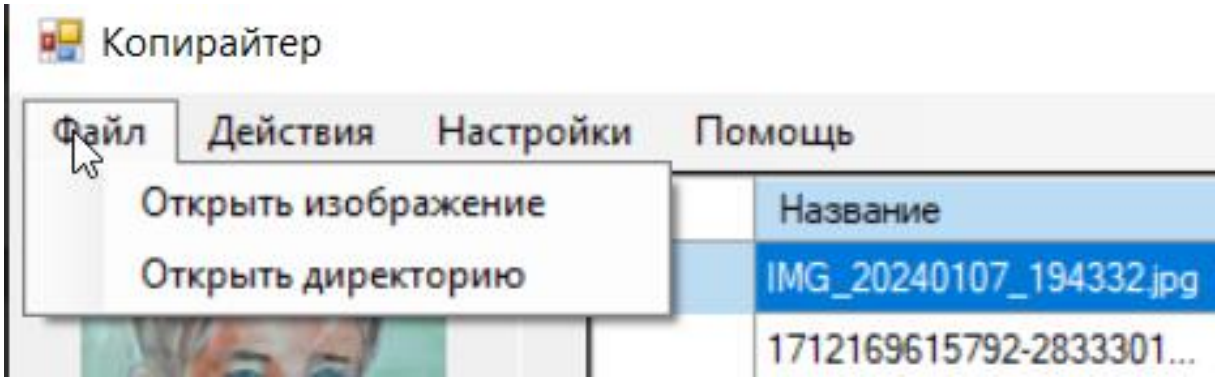

Копирайтер
Файл
Действия
Настройки
Помощь
Название
Ширина
Высота
Текст
Открыть изображение
Открыть директорию
Добавить водяной знак
Сохранить изображение
Пакетный режим обработки
Текст водяного знака
1
Обзор папок
Рабочий стол
OneDrive - Personal
Яндекс.Диск
Красько Михаил
Этот компьютер
Видео
Документы
Загрузки
Telegram Desktop
VulkanProgrammingGuide
Информатика и ИКТ
ЛР5
Изображения
Музыка
Объемные объекты
Рабочий стол
Локальный диск (C:)
Новый том (D:)
2
3
Создать папку
OK
Отмена

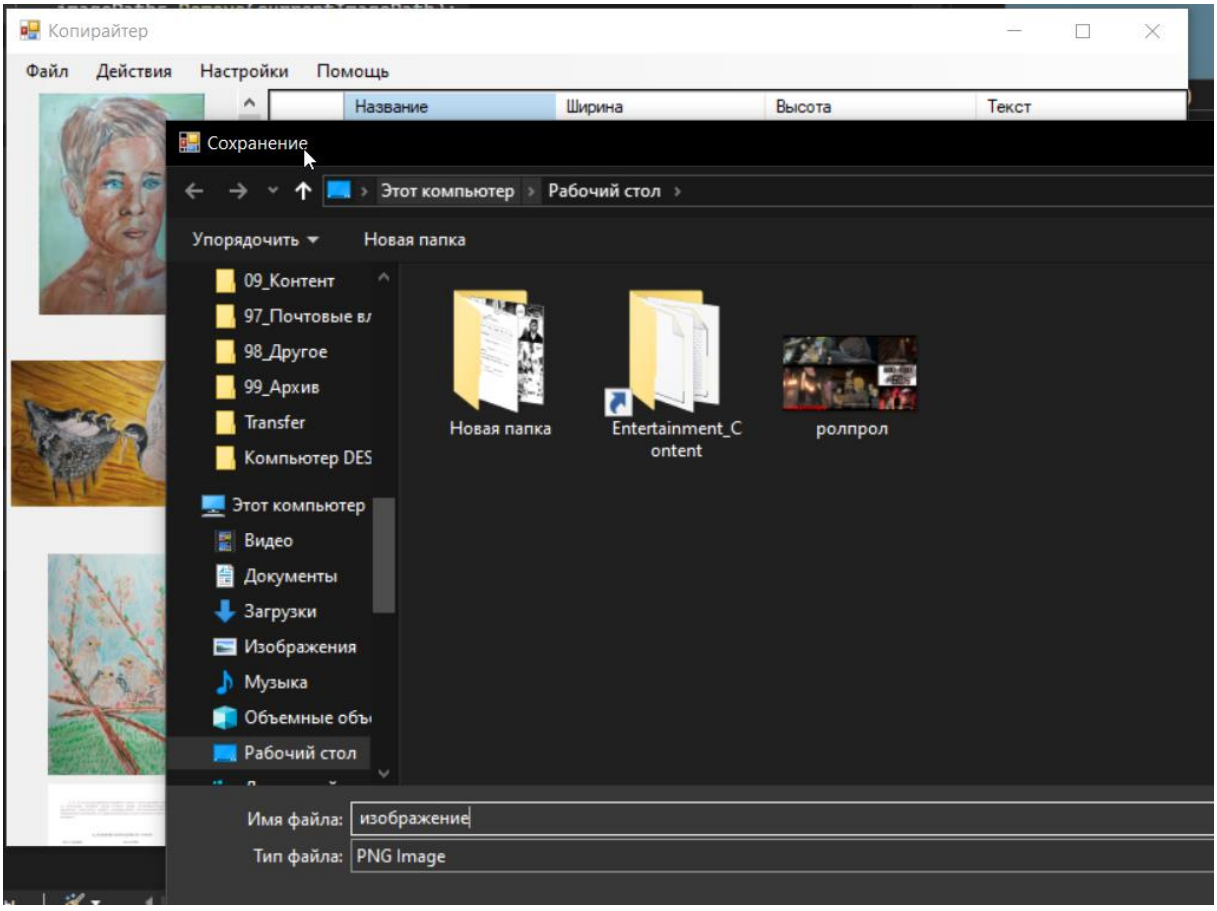

Копирайтер
Файл
Действия
Настройки
Помощь
Название
Ширина
Высота
Текст
Открыть изображение
Открыть директорию
Добавить водяной знак
Сохранить изображение
Пакетный режим обработки
Текст водяного знака

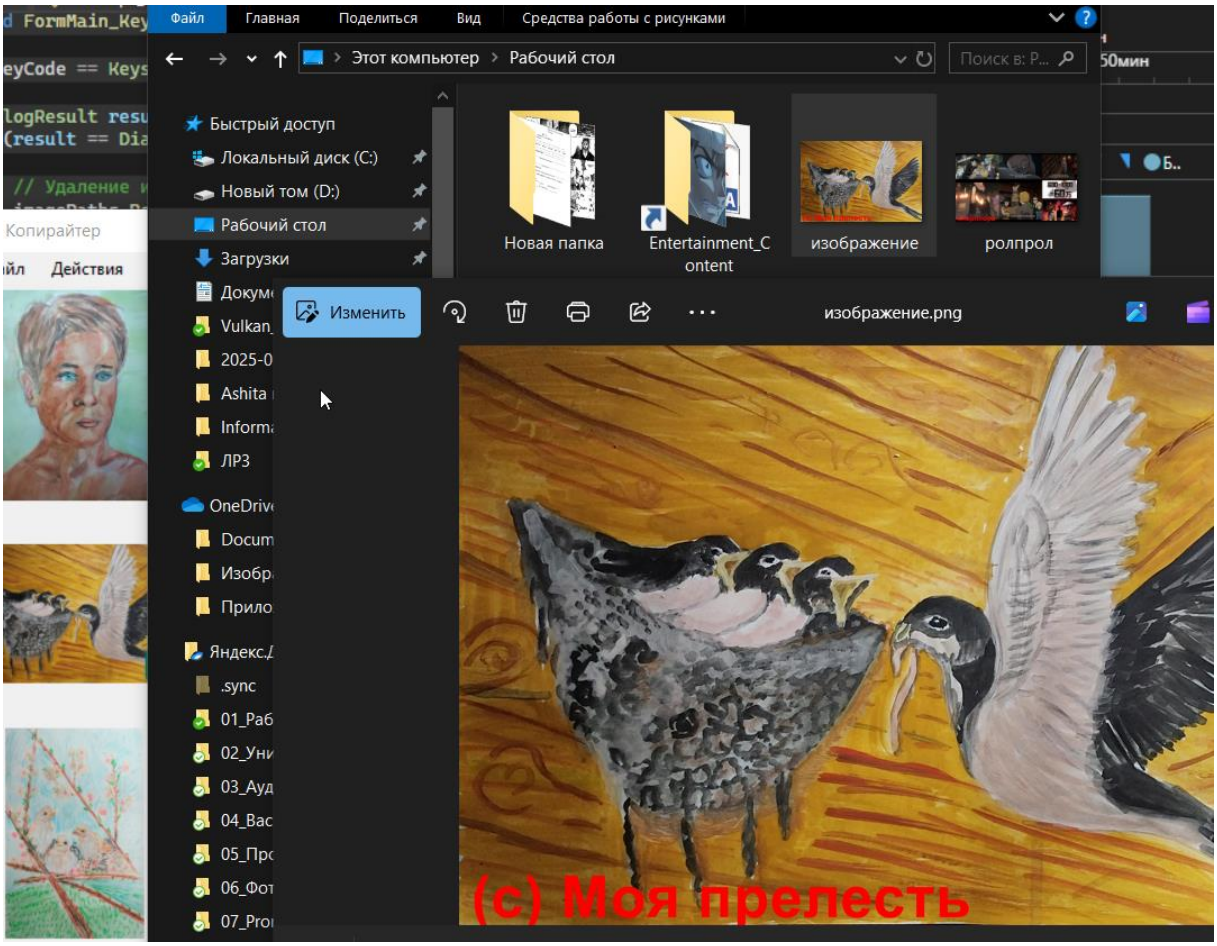

Копирайтер
Файл
Действия
Настройки
Помощь
Название
Ширина
Высота
Текст
IMG_20240107_194332.jpg
3000
4000
(c) Моя прелесть [13:18]
1712169615792-2833301...
3500
2333
(c) Моя прелесть [13:19]
Открыть изображение
Открыть директорию
Добавить водяной знак
Сохранить изображение
Пакетный режим обработки
Текст водяного знака
1
(c) Моя прелесть

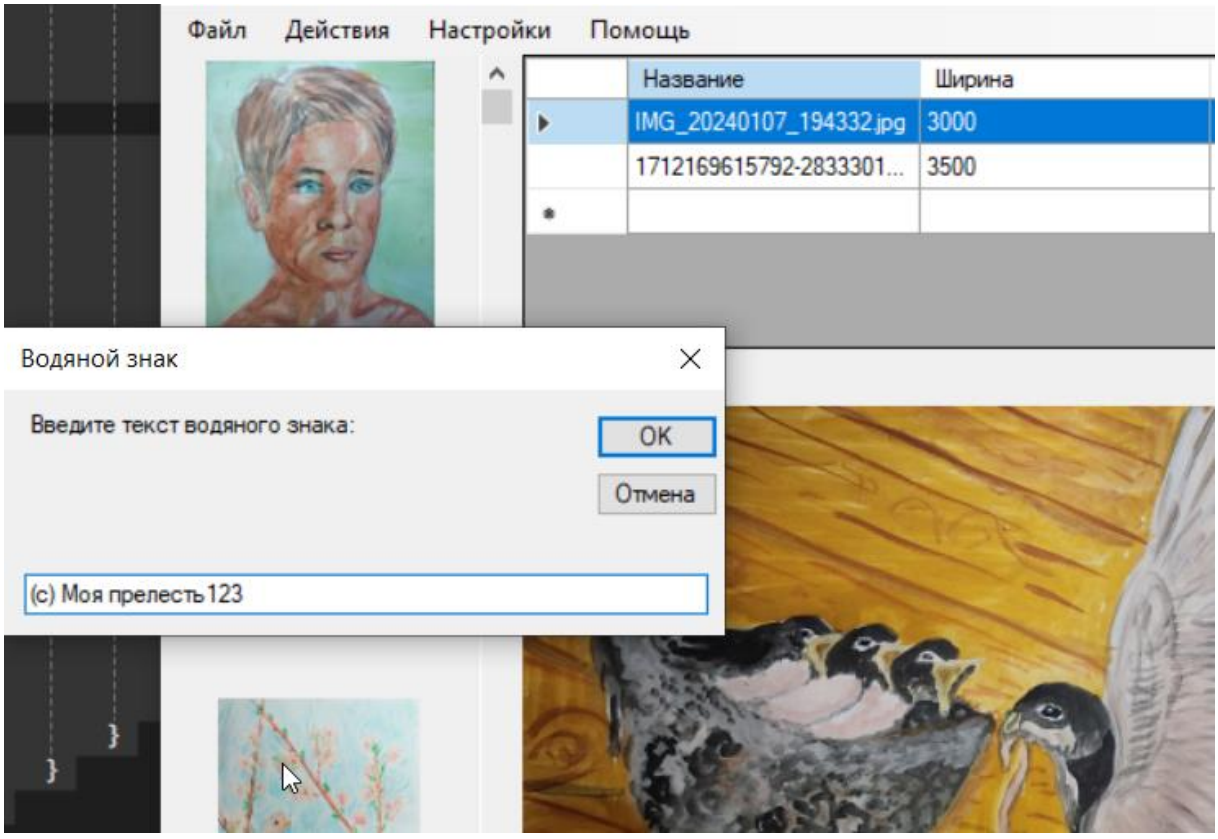

Копирайтер
Файл
Действия
Настройки
Помощь
Текст водяного знака
Папка для экспорта
Ширина
194332.jpg
3000
1712169615792-2833301...
3500

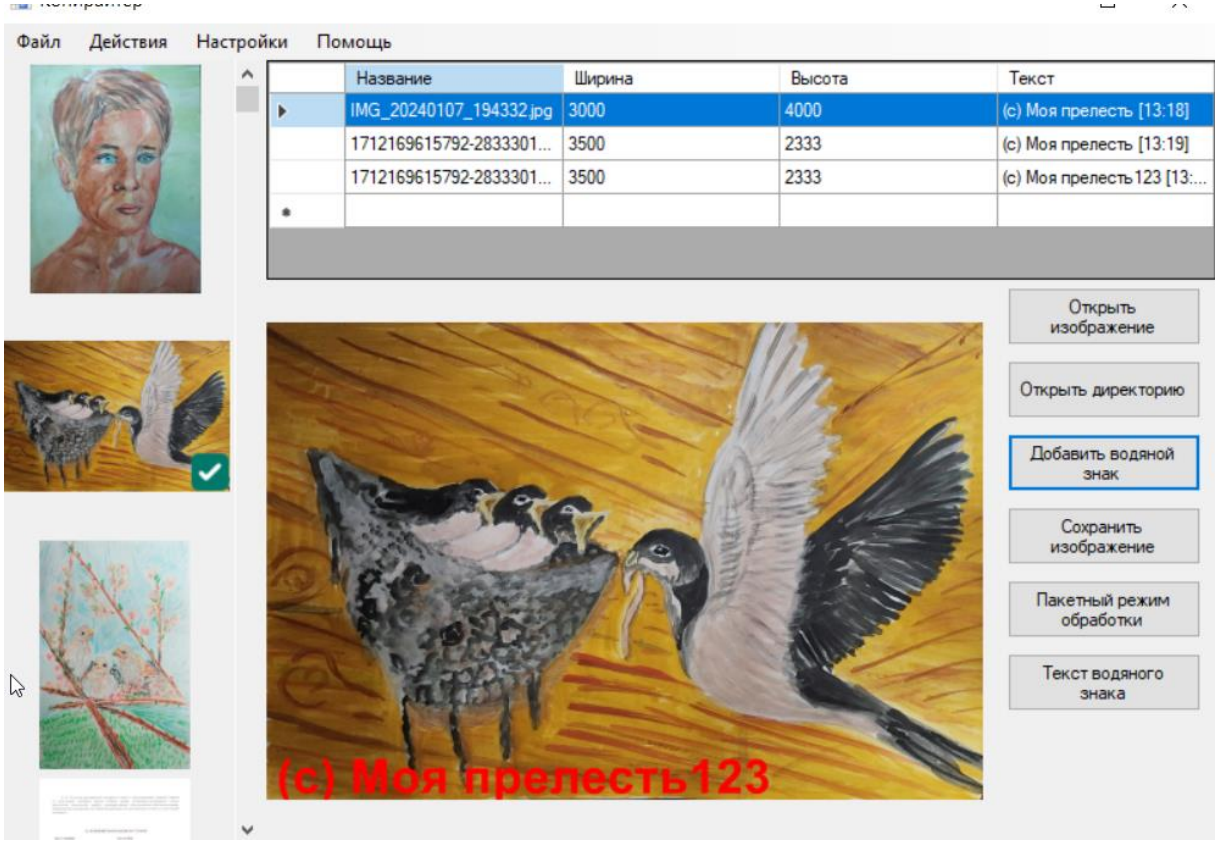

Копирайтер
Файл
Действия
Настройки
Помощь
Добавить водяной знак
Сохранить изображение
Пакетный режим обработки
Ширина
0107_194332.jpg
3000
5792-2833301...
3500

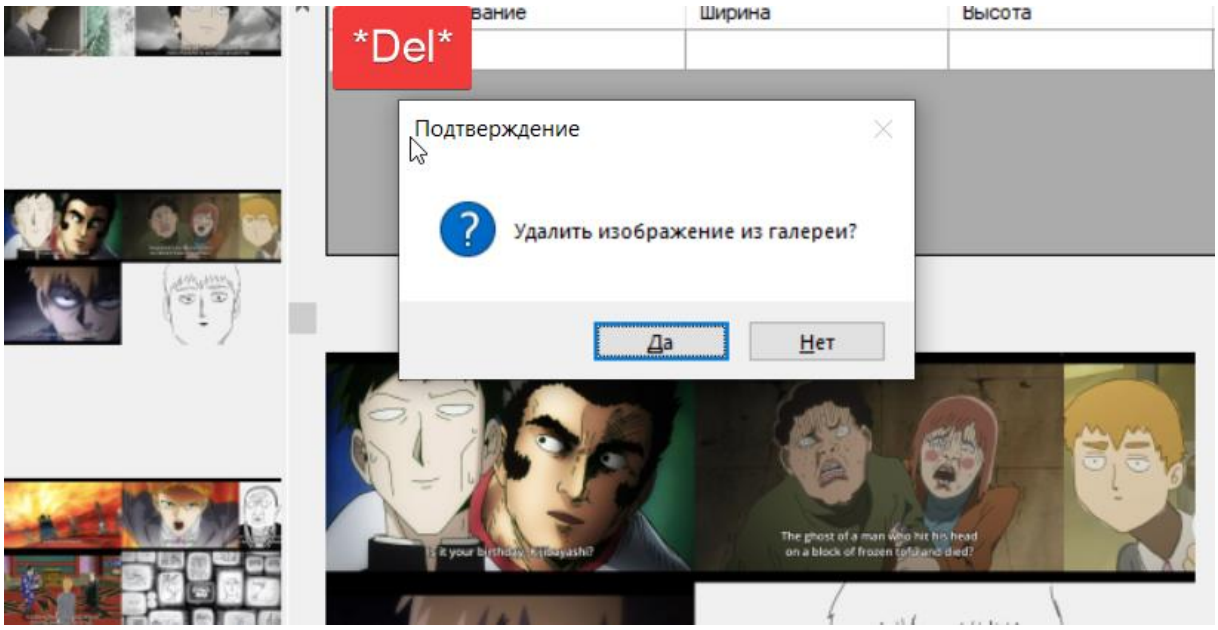

Копирайтер
Файл
Действия
Настройки
Помощь
Открыть изображение
Открыть директорию
Название
IMG_20240107_194332.jpg
1712169615792-2833301...

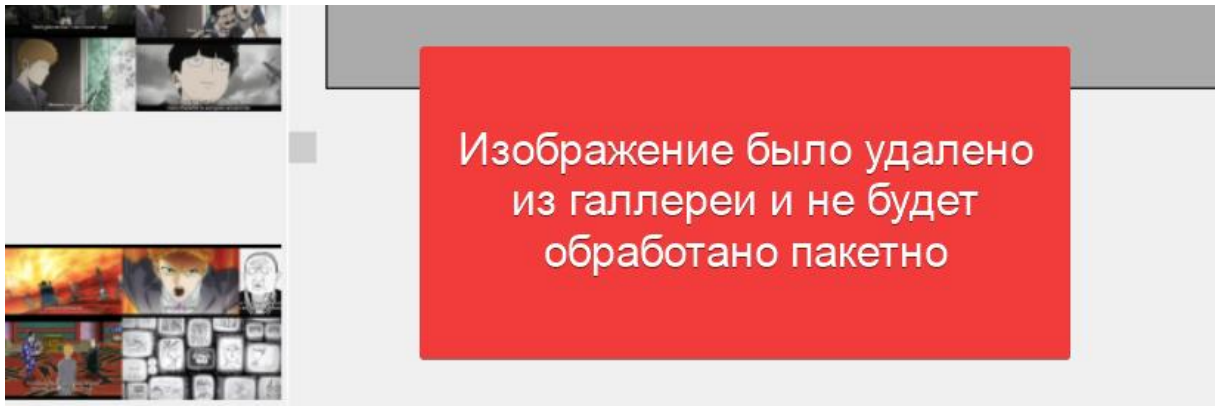

Копирайтер
Файл Действия Настройки Помощь
Название Ширина Высота Текст
Сохранение
Этот компьютер › Рабочий стол ›
Упорядочить
Новая папка
09_Контент
97_Почтовые в
98_Другое
99_Архив
Transfer
Компьютер DES
Этот компьютер
Видео
Документы
Загрузки
Изображения
Музыка
Объемные объ
Рабочий стол
Новая папка
Entertainment_Content
ролпрол
Имя файла: изображение
Тип файла: PNG Image

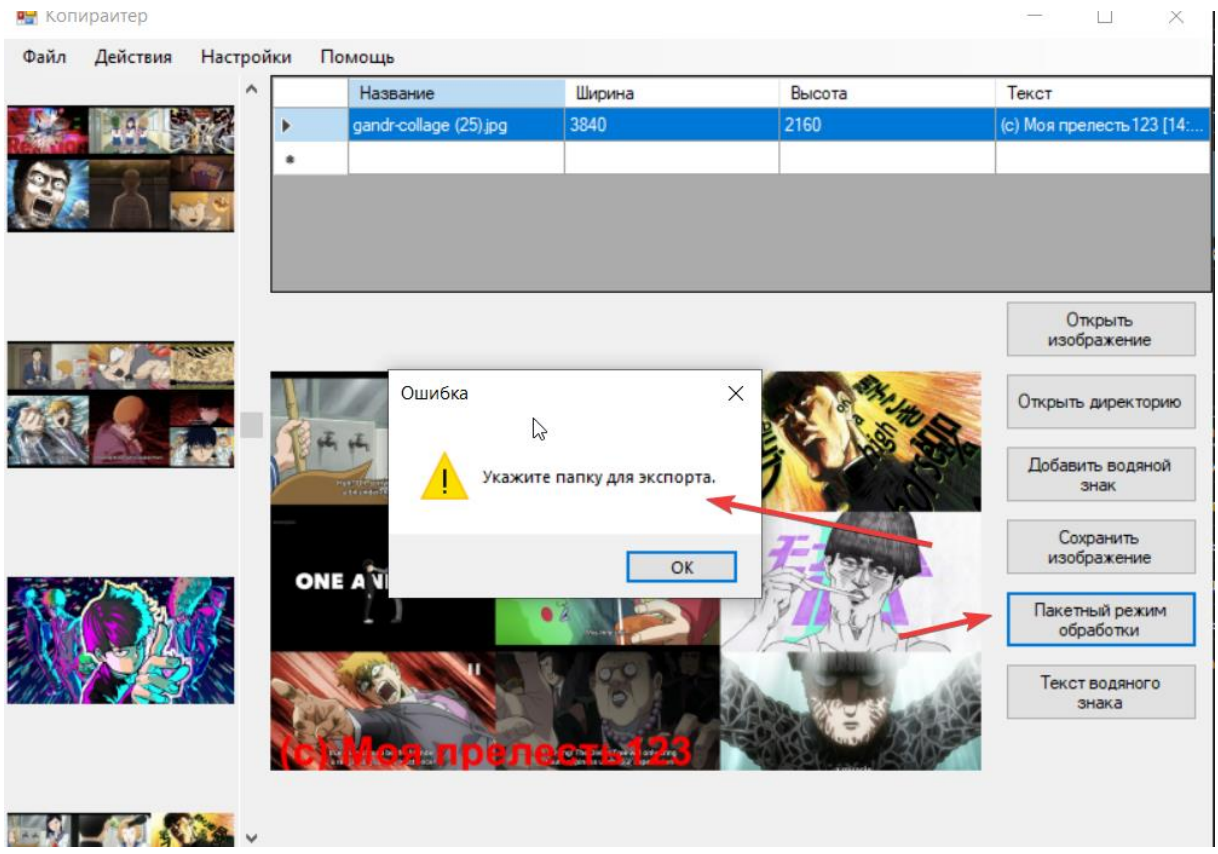

Файл Главная Поделиться Вид Средства работы с рисунками
Этот компьютер › Рабочий стол
Поиск в: Р...
Быстрый доступ
Локальный диск (C:)
Новый том (D:)
Рабочий стол
Загрузки
Изменить
Новая папка
Entertainment_Content
изображение
ролпрол
изображение.png
(c) Моя прелесть

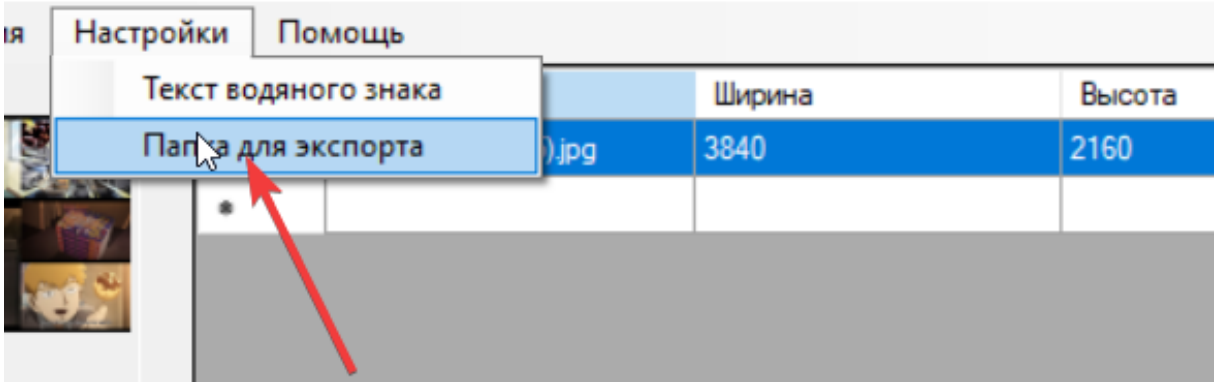

Файл
Действия
Настройки
Помощь
Название
Ширина
IMG_20240107_194332.jpg
3000
1712169615792-2833301...
3500
Водяной знак
Введите текст водяного знака:
OK
Отмена
(c) Моя прелесть123

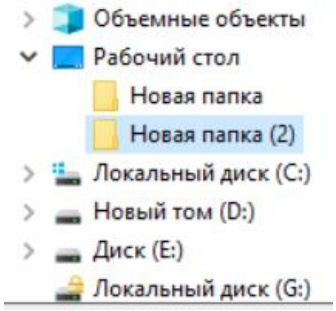

Файл
Действия
Настройки
Помощь
Название
Ширина
Высота
Текст
IMG_20240107_194332.jpg
3000
4000
(c) Моя прелесть [13:18]
1712169615792-2833301...
3500
2333
(c) Моя прелесть [13:19]
1712169615792-2833301...
3500
2333
(c) Моя прелесть123 [13:...
Открыть изображение
Открыть директорию
Добавить водяной знак
Сохранить изображение
Пакетный режим обработки
Текст водяного знака
(c) Моя прелесть123

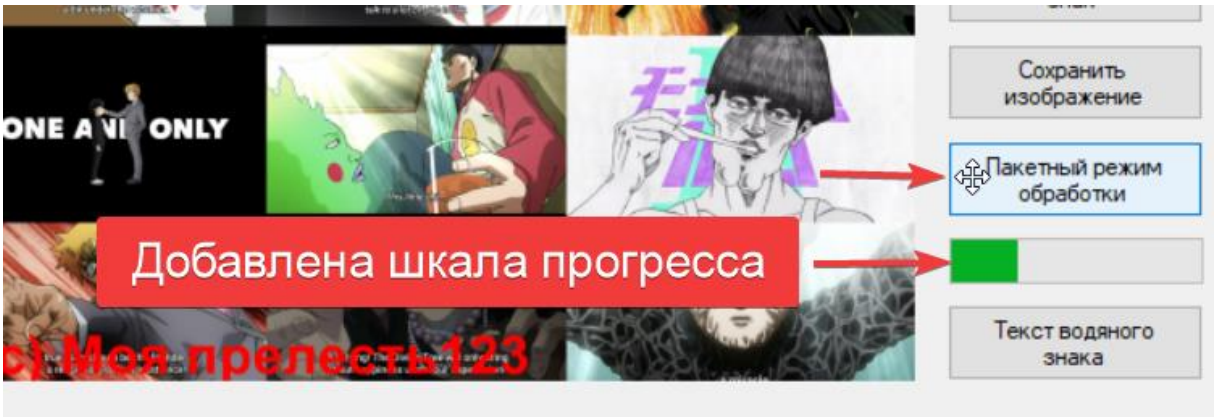

Del
Название
Ширина
Высота
Подтверждение
Удалить изображение из галереи?
Да
Нет

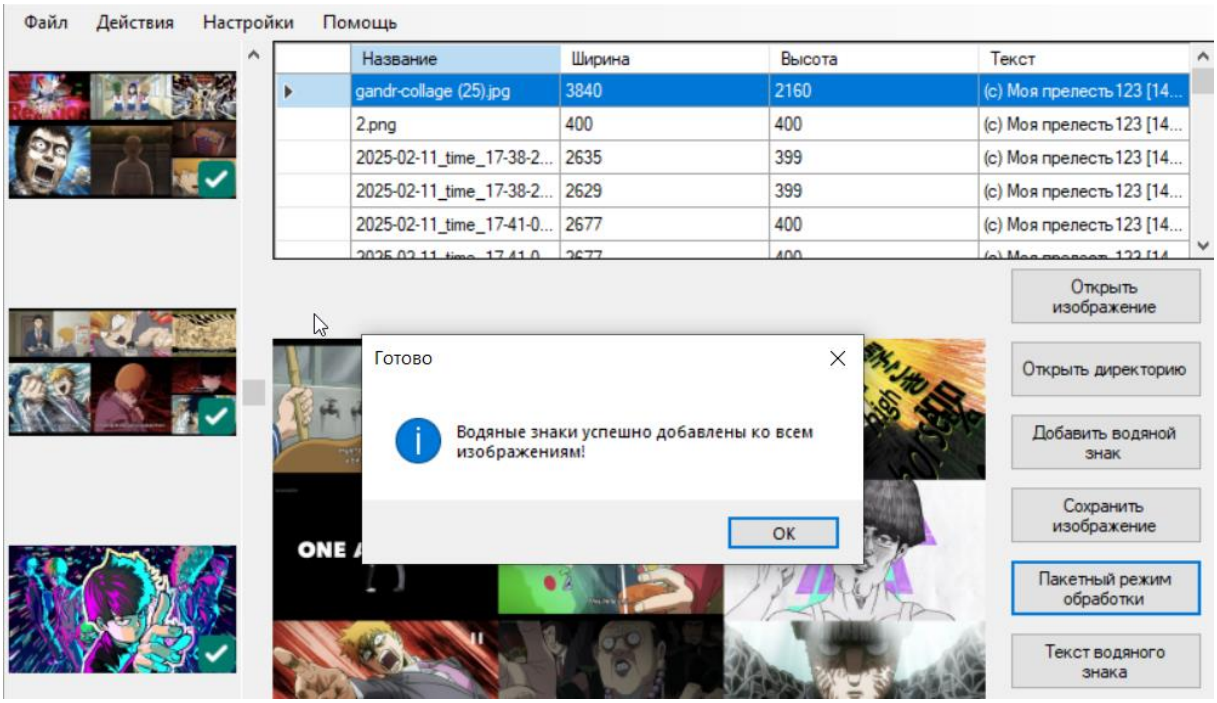

Изображение было удалено из галлереи и не будет обработано пакетно

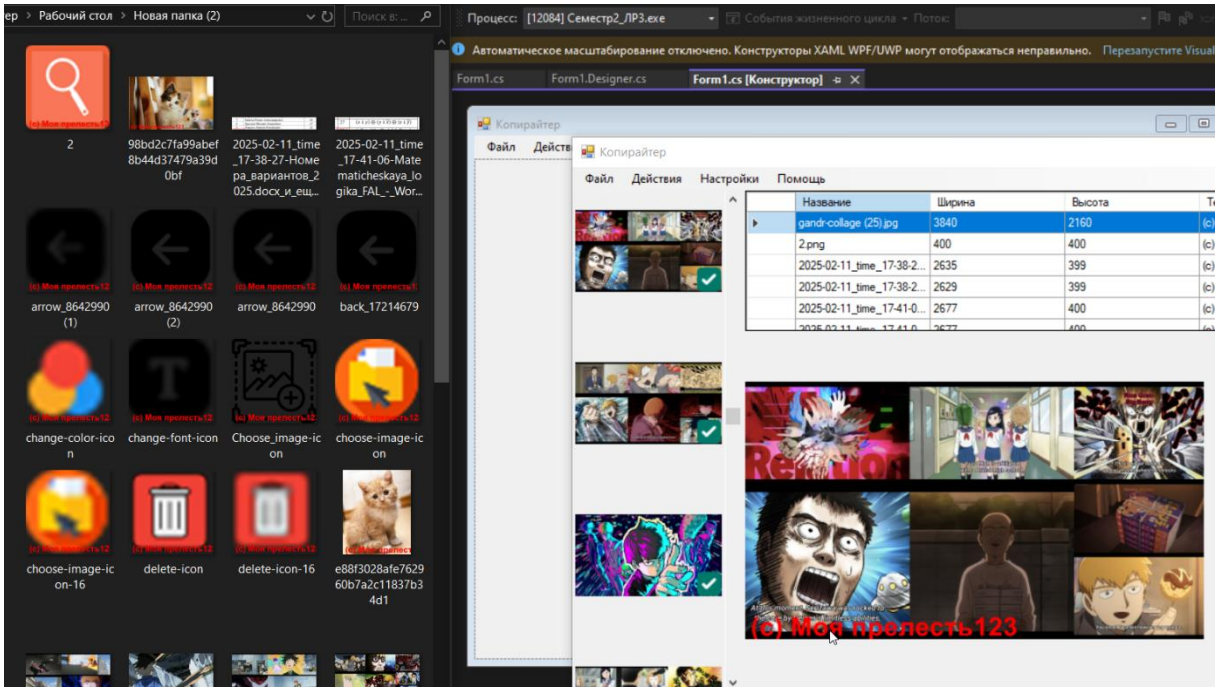

Копираитер
Файл
Действия
Настройки
Помощь
Название
Ширина
Высота
Текст
gandr-collage (25).jpg
3840
2160
(c) Моя прелесть123 [14:...
Открыть изображение
Открыть директорию
Добавить водяной знак
Сохранить изображение
Пакетный режим обработки
Текст водяного знака
Ошибка
Укажите папку для экспорта.
OK
(c) Моя прелесть123


Настройки
Помощь
Текст водяного знака
Папка для экспорта
Ширина
Высота
3840
2160


Объемные объекты
Рабочий стол
Новая папка
Новая папка (2)
Локальный диск (C:)
Новый том (D:)
Диск (E:)
Локальный диск (G:)

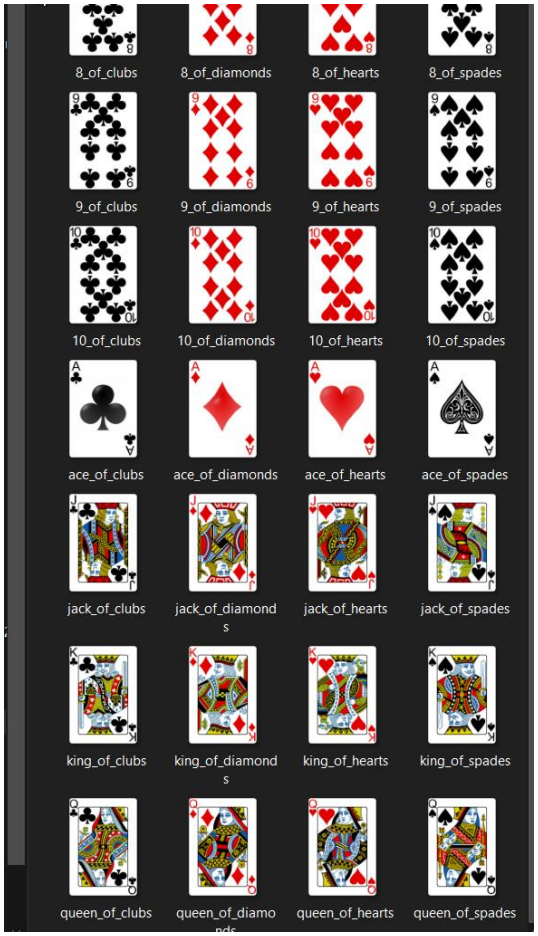

Добавлена шкала прогресса
Сохранить изображение
Пакетный режим обработки
Текст водяного знака

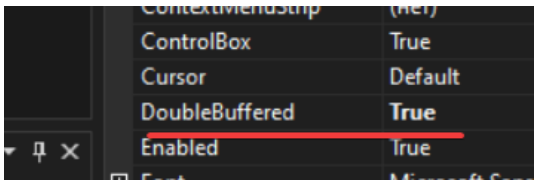

Файл
Действия
Настройки
Помощь
Название
Ширина
Высота
Текст
gandr-collage (25).jpg
3840
2160
(c) Моя прелесть 123 [14...
2.png
400
400
2025-02-11_time_17-38-2...
2635
399
2025-02-11_time_17-38-2...
2629
399
2025-02-11_time_17-41-0...
2677
400
Открыть изображение
Открыть директорию
Добавить водяной знак
Сохранить изображение
Пакетный режим обработки
Текст водяного знака
Готово
Водяные знаки успешно добавлены ко всем изображениям!
OK

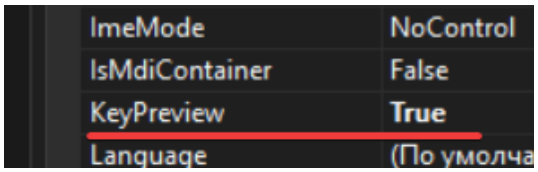


2) Разработать приложение, эмулирующее игровой стол с картами. Вывести колоду из 36 карт рубашкой вверх. Программа должна предоставлять возможность снять карту с верхушки колоды и перетащить в любое место игрового стола. При отпускании карты она должна падать на стол под случайным углом (рис.2). Заливка фона стола должна быть градиентной. При нажатии на клавишу F2 все карты должны обратно «собраться» в колоду. Реализовать двойную буферизацию при отрисовке перетаскиваемой карты.

Изображения карт были взяты с этого сайта: https://github.com/ntkhang03/poker-cards/tree/main.







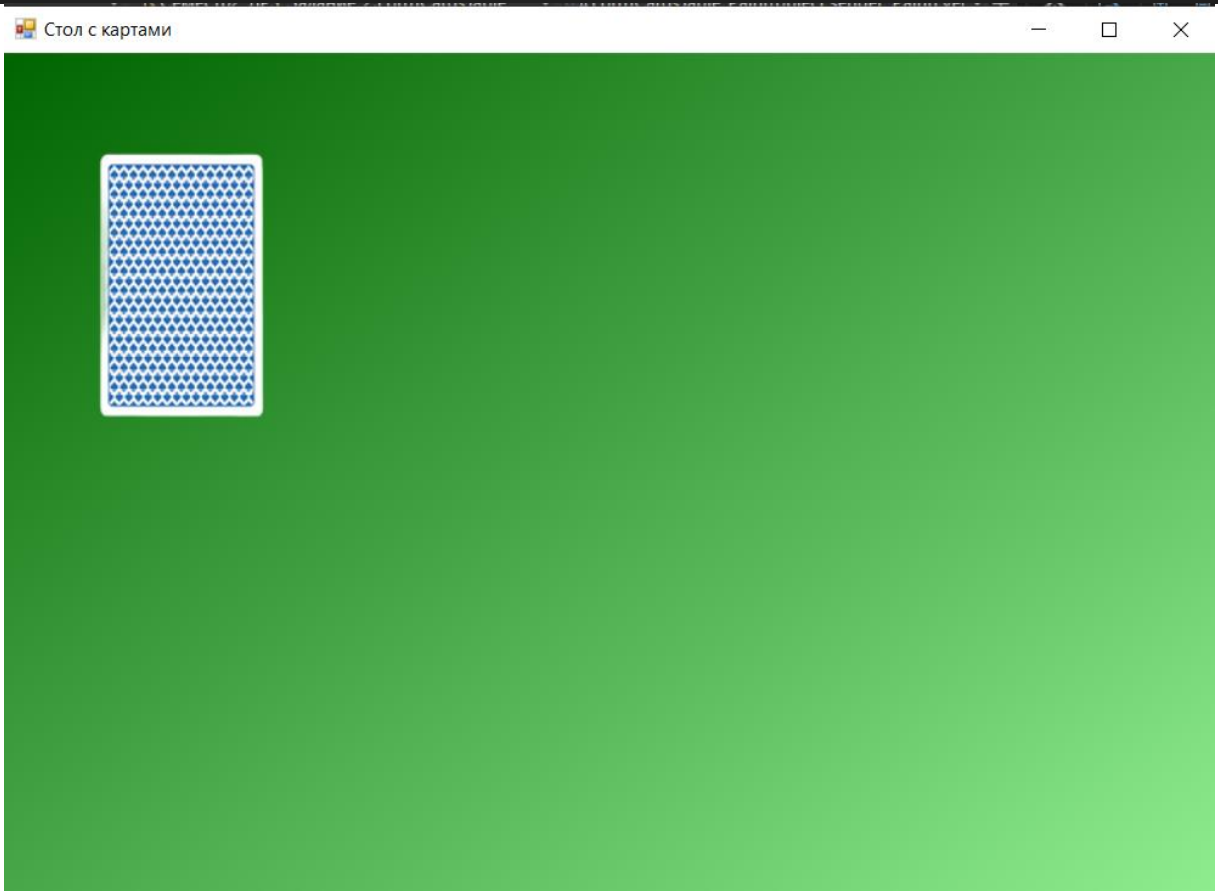

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq; // Для OrderBy
using System.Windows.Forms;
using System.IO;
using System.Drawing.Drawing2D; // Для LinearGradientBrush

namespace Семестр2_ЛР3_Задание_2
{
    public partial class FormCardsTable : Form
    {
        private List<Card> tableCards = new List<Card>();
        private Stack<Image> deck = new Stack<Image>();
        private Image backImage;

        private Card draggingCard = null;
```

```
        private Point dragOffset;
        private Random random = new Random();

        public FormCardsTable()
        {
            InitializeComponent();

            LoadCards();
        }

        private void LoadCards()
        {
            var files = Directory.GetFiles("cards", "*.png");

            foreach (var file in files)
            {
                var img = Image.FromFile(file);
                var resizedImg = ResizeImage(img, 140, 210); // Уменьшаем размер

                if (Path.GetFileName(file).ToLower() == "back.png")
                {
                    backImage = resizedImg;
                    continue;
                }

                deck.Push(resizedImg);
            }

            ShuffleDeck();
        }

        // Метод для изменения размера изображения
        private Image ResizeImage(Image img, int width, int height)
        {
            Bitmap bmp = new Bitmap(width, height);
            using (Graphics g = Graphics.FromImage(bmp))
            {
                g.InterpolationMode =
System.Drawing.Drawing2D.InterpolationMode.HighQualityBicubic;
                g.DrawImage(img, 0, 0, width, height);
            }
            return bmp;
        }


        private void ShuffleDeck()
        {
            var shuffledList = deck.ToList().OrderBy(x => random.Next()).ToList();
            deck = new Stack<Image>(shuffledList);
        }

        private void MainForm_Paint(object sender, PaintEventArgs e)
```

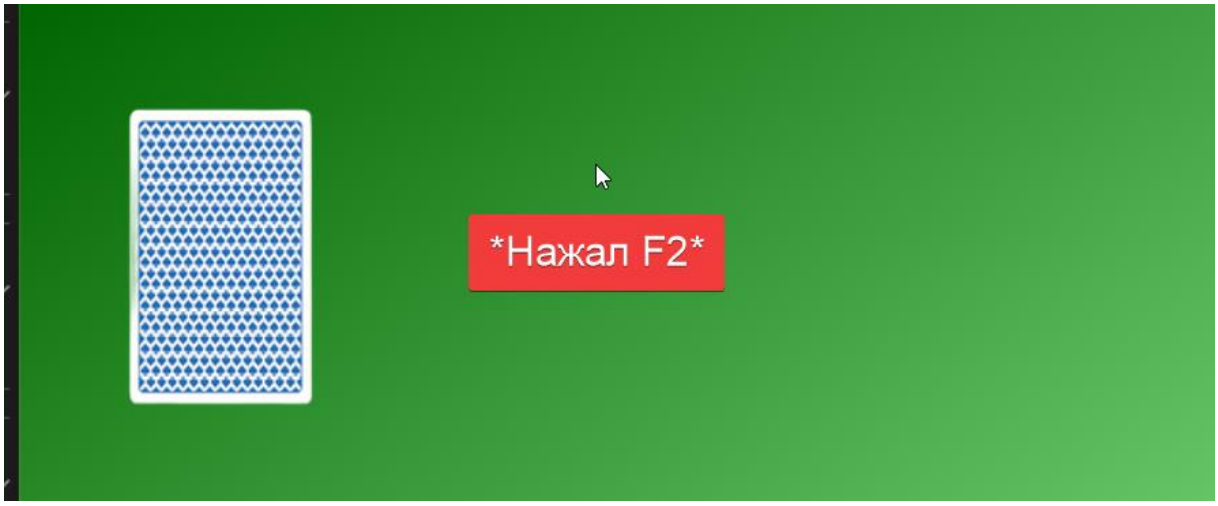
```
        {

        }

        private void DrawCard(Graphics g, Card card, bool isDragging = false)
        {
            g.TranslateTransform(card.Position.X, card.Position.Y);
            g.RotateTransform(card.Rotation);

            var image = card.Image;
            g.DrawImage(image, -image.Width / 2, -image.Height / 2, image.Width,
image.Height);

            g.ResetTransform();
        }

        private void FormCardsTable_MouseDown(object sender, MouseEventArgs e)
        {
            if (deck.Count > 0 && new Rectangle(50, 50, backImage.Width,
backImage.Height).Contains(e.Location))
            {
                var cardImage = deck.Pop();

                draggingCard = new Card
                {
                    Image = cardImage,
                    Position = e.Location,
                    Rotation = 0
                };

                dragOffset = new Point(cardImage.Width / 2, cardImage.Height / 2);
            }

            Invalidate();
        }

        private void FormCardsTable_MouseMove(object sender, MouseEventArgs e)
        {
            if (draggingCard != null)
            {
                draggingCard.Position = e.Location;
                Invalidate();
            }
        }

        private void FormCardsTable_MouseUp(object sender, MouseEventArgs e)
        {
            if (draggingCard != null)
            {
                draggingCard.Position = e.Location;
                draggingCard.Rotation = random.Next(-20, 20);
                tableCards.Add(draggingCard);
```

```
                draggingCard = null;
                Invalidate();
            }
        }

        private void FormCardsTable_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyCode == Keys.F2)
            {
                foreach (var card in tableCards)
                {
                    deck.Push(card.Image);
                }

                tableCards.Clear();
                ShuffleDeck();
                Invalidate();
            }
        }

        private void FormCardsTable_Paint(object sender, PaintEventArgs e)
        {
            Graphics g = e.Graphics;

            // Градиентный фон
            using (LinearGradientBrush brush = new
LinearGradientBrush(this.ClientRectangle,
                Color.DarkGreen, Color.LightGreen,
LinearGradientMode.ForwardDiagonal))
            {
                g.FillRectangle(brush, this.ClientRectangle);
            }

            // Отрисовка карт на столе
            foreach (var card in tableCards)
            {
                DrawCard(g, card);
            }

            // Колода
            if (deck.Count > 0)
            {
                g.DrawImage(backImage, 50, 50, backImage.Width, backImage.Height);
            }

            // Перетаскиваемая карта
            if (draggingCard != null)
            {
                DrawCard(g, draggingCard, true);
            }
        }
```

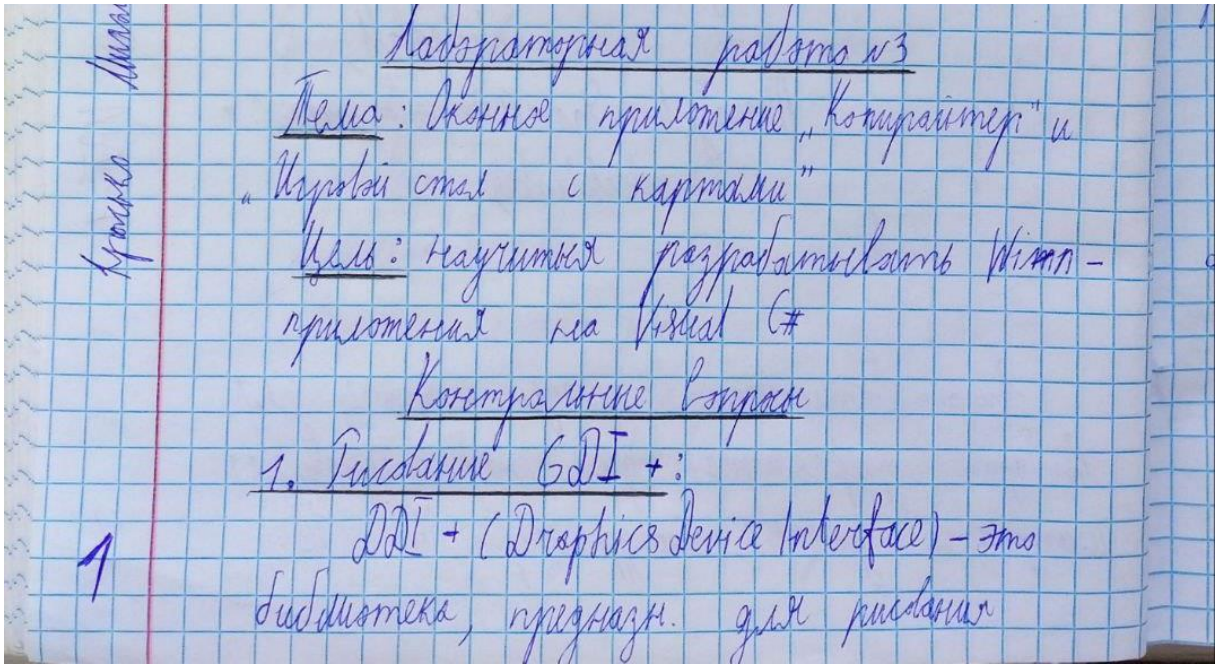
```
        private void FormCardsTable_Resize(object sender, EventArgs e)
        {
            this.Invalidate(); // Отрисовываем форму снова. Это нужно для
градиентного заднего фона
        }
    }

    public class Card
    {
        public Image Image;
        public PointF Position;
        public float Rotation;
    }
}
```




Стол с картами


Нажал F2

Контрольные вопросы



Лабораторная работа №3

Тема: Оконное приложение „Конструктор" и „Игровой стол с картами"

Цель: научиться разрабатывать Windows-приложения на Visual C#

Контрольные вопросы

1. Рисование GDI+:

GDI+ (Graphics Device Interface) – это библиотека, предназн. для рисования

графики в Windows-приложениях.
Она позволяет работать с векторной графикой, текстом, изобр. и многим другим. В .NET Framework GDI+ реализ. через простр. имен System.Drawing.
Осн. операции, к-е можно выполнить с помощью GDI+:
- создание граф. объектов через Graphics (напр. для рисования на формах).
- Работа с цветами через класс Color.
- рисование примитивов (линий, прямоугольников, кругов) с исп. методов DrawLine, DrawRectangle, DrawEllipse.
- Рисование текста через DrawString.
- рисование изобр. с помощью DrawImage.

Пример:
protected override void OnPaint(PaintEventArgs e)

Курсовой минимум ВИРТ-4

2

```
{
    base.OnPaint(e);
    Graphics g = e.Graphics;
    Pen pen = new Pen(Color.Black);
    g.DrawLine(pen, 10, 10, 100, 100);
    g.DrawRectangle(pen, 150, 150, 200, 100);
}
```

2. Двойная буферизация

Двойная буферизация исп. для уменьшения мерцания и улучшения произв. при рисовании на экране. В случае прямой отрисовки на экране каждый кадр может быть виден пользователю, что вызывает мерцание. Чтобы этого избежать, исп. два буфера:

- осн. буфер (экранный буфер) - тот, что отобр. на экране;
- вспом. буфер - временный буфер, в

к-м выполняется рисование.
В Windows Forms можно вкл. двойную буферизацию с помощью установки флага:

this.DoubleBuffered = true;

3. Элемент упр. DataGridView:

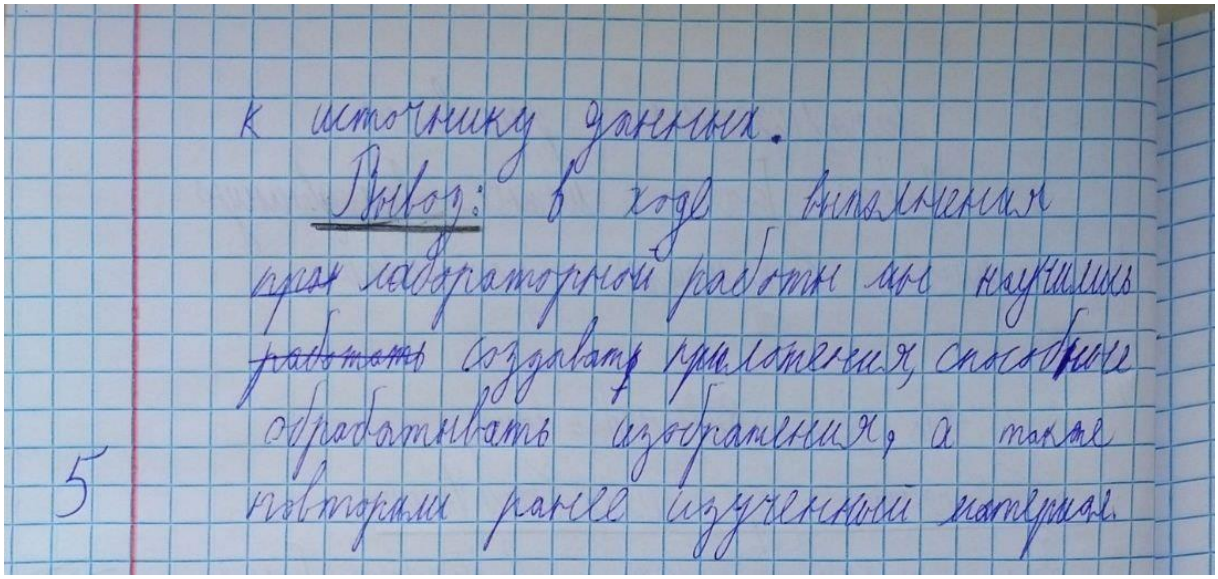
Это элемент упр. в Windows Forms, к-й исп. для отобр. табличных данных. Он предоставляет интерфейс для работы с данными в строках и столбцах. Его возможности:

- редакт. данных;
- сортировка;
- фильтрация;
- группировка и агрегация: можно сгруп. данные и выполнить вычисл.;
- настройка внешнего вида.

Также можно привязать DataGridView

Кулько Михаил ИВТ-4

4



Вывод: в ходу выполнения лабораторной работы мы научились создавать приложения, способные обрабатывать изображения, а также повторили ранее изученный материал.