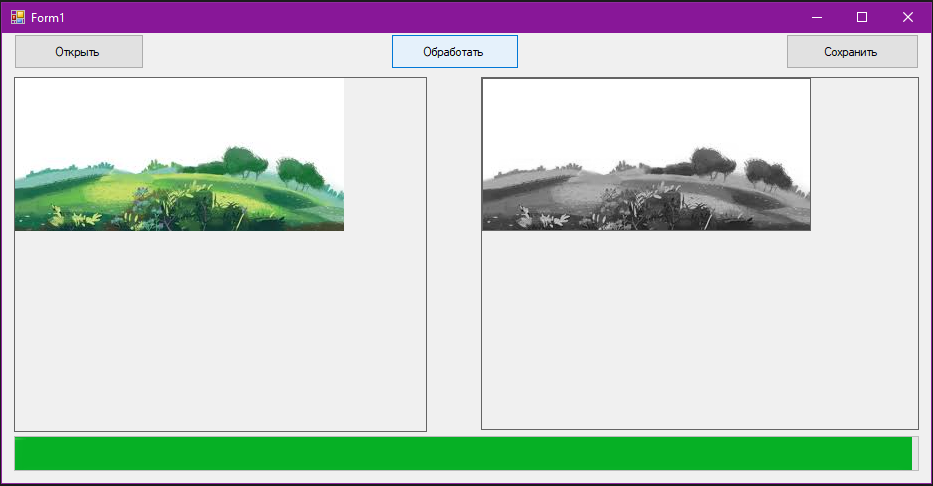
**Моргунов А. ПИ-18б**

**Лабораторная работа №2**

«Создание и использование динамически загружаемой библиотеки для работы с изображениями»

Задание

Применить произвольный фильтр к изображению(ч/б фильтр)



Код программы

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyLib

{

//Указатель на функцию обработки прогресса выполнения задачи

public delegate void ProgressDelegate(double percent);

public interface ImageHandler

{

//получение осмысленного имени обработчика

string HandlerName { get; }

//Инициализация параметров обработчика

void init(SortedList<string, object> parameters);

//Установка изображения-источника

Bitmap Source { set; }

//Получение изображения-результата

Bitmap Result { get; }

//Запуск обработки

void startHandle(ProgressDelegate progress);

}

public class ImageWorker : ImageHandler

{

public string HandlerName { set; get; }

public Bitmap Source { set; get; }

public Bitmap Result { set; get; }

public void init(SortedList<string, object> parameters)

{

}

public void startHandle(ProgressDelegate progress)//K-Mean method

{

Result = new Bitmap(Source.Width, Source.Height);

for (int j = 0; j < Source.Height; j++)//перебор пикселей изображения

{

for (int i = 0; i < Source.Width; i++)

{

UInt32 pixel = (UInt32)(Source.GetPixel(i, j).ToArgb());

double R = (double)((pixel & 0x00FF0000) >> 16);

double G = (double)((pixel & 0x0000FF00) >> 8);

double B = (double)((pixel & 0x000000FF));

double c = (R+G+B)/ 3;

UInt32 newPixel = 0xFF000000 | ((UInt32)c << 16) | ((UInt32)c << 8) | ((UInt32)c);

Result.SetPixel(i, j, Color.FromArgb((int)newPixel));

}

progress(j);

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

using MyLib;

namespace VVP\_lab\_2

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace VVP\_lab\_2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

OpenFileDialog ofd = new OpenFileDialog();

ofd.Filter = "Image Files(\*.BMP;\*.JPG;\*.PNG;)|\*.BMP;\*.JPG;\*.PNG|All files (\*.\*)|\*.\*";

ofd.Title = "Выбрать картинку для обработки";

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

{

try

{

pictureBox1.Image = new Bitmap(ofd.FileName);

pictureBox1.Bounds = new Rectangle(new Point(0,0),new Size(pictureBox1.Image.Width, pictureBox1.Image.Height));

}

catch

{

MessageBox.Show("Невозможно открыть выбранный файл","Ошибка",MessageBoxButtons.OK,MessageBoxIcon.Error);

}

}

}

private void progressBar1\_Click(object sender, EventArgs e)

{

}

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

{

if(pictureBox2.Image != null)

{

SaveFileDialog sfd = new SaveFileDialog();

sfd.Title = "Сохранить картинку как...";

sfd.OverwritePrompt = true;

sfd.CheckPathExists = true;

sfd.Filter = "Image Files(\*.BMP)|\*.BMP|Image Files(\*.JPG)|\*.JPG|Image Files(\*.PNG)|\*.PNG|All Files(\*.\*)|\*.\*";

sfd.ShowHelp = true;

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

{

try

{

pictureBox2.Image.Save(sfd.FileName);

}

catch

{

MessageBox.Show("Невозможно сохранить изображение", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

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

{

Console.WriteLine("------------------------------");

count = 0;

if (pictureBox1.Image != null)

{

progressBar1.Value = 0;

MyLib.ImageWorker imageWorker = new MyLib.ImageWorker();

MyLib.ProgressDelegate progress;

progress = method;

imageWorker.Source= new Bitmap(pictureBox1.Image);

progressBar1.Maximum = pictureBox1.Image.Height;

imageWorker.startHandle(progress);

pictureBox2.Image = imageWorker.Result;

pictureBox2.Bounds = new Rectangle(new Point(0, 0), new Size(pictureBox1.Image.Width, pictureBox1.Image.Height));

}

}

int count = 0;

public void method(double yy)

{

progressBar1.Value = (int)yy;

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

}

}

}