УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе № 2

по предмету «Объектно-Ориентированные Технологии Программирования и Стандарты Проектирования»

Выполнил:

Руденя Д. А.

гр. 351001

Проверил:

Деменковец Д.В.

Минск 2025

Задание к лабораторной работе

Расширить пример, чтобы можно было менять поля через свойства и просто на уровне интерфейса. Cоздать новый класс без изменения уже существующих.

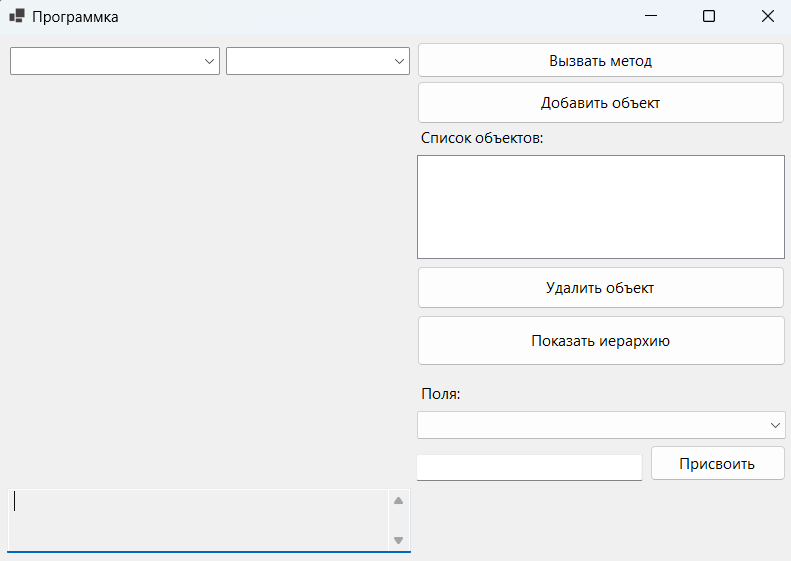


Рисунок 1. Внешний вид программы

Базовый начальный класс ElectronicDevice:

namespace Lab1;  
  
public abstract class ElectronicDevice : IDevice  
{  
 public abstract void Property();  
}

Новый класс MacBook.cs:

namespace Lab1;  
  
public class MacBook : LaptopDevice, IPowerable, IDevice  
{  
   
}

Код главной формы:

using System;  
using System.Drawing;  
using System.Linq;  
using System.Reflection;  
using System.Text;  
using System.Windows.Forms;  
using AForge.Video;  
using AForge.Video.DirectShow;  
using Emgu.CV;  
using Emgu.CV.CvEnum;  
  
namespace Lab1  
{  
 public partial class MainForm : Form  
 {  
   
 private static MainForm \_instance;  
 public static MainForm Instance => \_instance;  
  
 private ClassesContainer container = new ClassesContainer();  
 private FilterInfoCollection videoDevices;   
 private VideoCaptureDevice videoSource;   
 private Bitmap currentFrame;   
  
 public MainForm()  
 {  
 \_instance = this;  
 InitializeComponent();  
 InitializeVideoCapture();  
 var deviceTypes = Assembly.GetExecutingAssembly().GetTypes()  
 .Where(t => t.IsClass && typeof(ElectronicDevice).IsAssignableFrom(t) && t != typeof(ElectronicDevice))  
 .Select(t => t.Name)  
 .ToArray();  
  
 comboBox1.Items.AddRange(deviceTypes);  
 comboBox1.SelectedIndexChanged += ComboBox1\_SelectedIndexChanged;  
  
 button1.Click += Button1\_Click;  
   
 button2.Click += Button2\_Click;  
 }  
   
 private void InitializeVideoCapture()  
 {  
 videoDevices = new FilterInfoCollection(FilterCategory.VideoInputDevice);  
  
 if (videoDevices.Count == 0)  
 {  
 MessageBox.Show("Не найдены видеоустройства.");  
 return;  
 }  
   
 videoSource = new VideoCaptureDevice(videoDevices[1].MonikerString);  
   
 if (videoSource.VideoCapabilities.Length == 0)  
 {  
 MessageBox.Show("Нет доступных разрешений для выбранного устройства.");  
 return;  
 }  
   
 videoSource.VideoResolution = videoSource.VideoCapabilities  
 .FirstOrDefault(vc => vc.FrameSize.Width == 640 && vc.FrameSize.Height == 480)   
 ?? videoSource.VideoCapabilities[1];   
 videoSource.NewFrame += new NewFrameEventHandler(VideoSource\_NewFrame);  
   
 videoSource.Start();  
 }  
   
 private void VideoSource\_NewFrame(object sender, NewFrameEventArgs eventArgs)  
 {  
 try  
 {  
 currentFrame = (Bitmap)eventArgs.Frame.Clone();  
 }  
 catch (Exception ex)  
 {  
 MessageBox.Show($"Ошибка при захвате кадра: {ex.Message}");  
 }  
 }  
   
 private void ComboBox1\_SelectedIndexChanged(object sender, EventArgs e)  
 {  
 comboBox2.Items.Clear();  
   
 string selectedClass = comboBox1.SelectedItem.ToString();  
 Type? type = Type.GetType($"Lab1.{selectedClass}");  
  
 if (type != null)  
 {  
 var methods = type.GetMethods(BindingFlags.**Public** | BindingFlags.**Instance** | BindingFlags.**FlattenHierarchy**)  
 .Where(m => !m.IsSpecialName && m.DeclaringType != typeof(object))  
 .Select(m => m.Name)  
 .Distinct()  
 .ToArray();  
   
 comboBox2.Items.AddRange(methods);  
 comboBox2.SelectedIndex = 0;  
 }  
 }  
  
 private void Button1\_Click(object sender, EventArgs e)  
 {  
 if (listBox1.SelectedItem == null)  
 {  
 Output("Выберите объект из списка.");  
 return;  
 }  
  
 int index = listBox1.Items.IndexOf(listBox1.SelectedItem);  
   
 ElectronicDevice selectedDevice = listBox1.SelectedItem as ElectronicDevice;  
 string selectedMethodName = comboBox2.SelectedItem.ToString();  
  
 MethodInfo method = selectedDevice.GetType().GetMethod(selectedMethodName);  
 if (method == null)  
 {  
 MessageBox.Show("Метод не найден.");  
 return;  
 }  
  
 ParameterInfo[] paramsInfo = method.GetParameters();  
 object[] parameters = null;  
 if ((selectedDevice is SmartphoneDevice smartphone) && selectedMethodName == "TakePhoto")  
 {  
 if (currentFrame != null && currentFrame is Bitmap bitmapFrame)  
 {  
 parameters = new object[] { bitmapFrame };  
 }  
 else  
 {  
 MessageBox.Show("Invalid frame");  
 }  
 }  
 else  
 {  
 if ((selectedDevice is LaptopDevice laptop) && selectedMethodName == "TakePhoto")  
 {   
 if (currentFrame != null)  
 {  
 parameters = new object[] { currentFrame };  
 }  
 else  
 {  
 Output("Не удалось сделать фото.");  
 }  
 }  
 else  
 {  
 if (paramsInfo.Length > 0)  
 {  
 using (ParameterInputForm inputForm = new ParameterInputForm(paramsInfo))  
 {  
 if (inputForm.ShowDialog() == DialogResult.**OK**)  
 {  
 parameters = inputForm.Parameters;  
 }  
 else  
 {  
 return;  
 }  
 }  
 }  
 }  
 }  
 container.CallMethod(index, selectedMethodName, parameters);  
 }  
  
  
 private void Button2\_Click(object sender, EventArgs e)  
 {  
 string selectedClass = comboBox1.SelectedItem.ToString();  
 Type type = Type.GetType($"Lab1.{selectedClass}");  
  
 if (type != null)  
 {  
 ElectronicDevice newDevice = null;  
  
 if (type == typeof(SmartphoneDevice))  
 {  
 newDevice = new SmartphoneDevice();  
 }  
 else if (type == typeof(LaptopDevice))  
 {  
 newDevice = new LaptopDevice();  
 }  
 else  
 {  
 newDevice = Activator.CreateInstance(type) as ElectronicDevice;  
 }  
  
 if (newDevice != null)  
 {  
 container.AddElectronicDevice(newDevice);  
 listBox1.Items.Add(newDevice);  
 Output($"Создан объект: {newDevice.GetType().Name}");  
 }  
 }  
 }  
   
 public void Output(string message)  
 {  
 textBoxOutput.AppendText(message + Environment.NewLine);  
 }  
  
 private void ShowHierarchy\_Click(object sender, EventArgs e)  
 {  
 ClassHierarchyForm hierarchyForm = new ClassHierarchyForm();  
 hierarchyForm.Show();  
 }  
   
 protected override void OnFormClosing(FormClosingEventArgs e)  
 {  
 if (videoSource.IsRunning)  
 {  
 videoSource.SignalToStop();  
 videoSource.WaitForStop();  
 }  
 base.OnFormClosing(e);  
 }  
  
 private void button3\_Click(object sender, EventArgs e)  
 {  
 if (listBox1.SelectedItem == null)  
 {  
 Output("Выберите объект из списка.");  
 return;  
 }  
  
 ElectronicDevice selectedDevice = listBox1.SelectedItem as ElectronicDevice;  
 container.RemoveElectronicDevice(selectedDevice);  
 listBox1.Items.Remove(selectedDevice);  
 }  
  
 private void FieldButton\_Click(object sender, EventArgs e)  
 {  
 if (FieldsComboBox.SelectedItem == null)  
 {  
 MessageBox.Show("Вы не выбрали свойство для изменения");  
 return;  
 }  
 if (listBox1.SelectedItem == null)  
 {  
 MessageBox.Show("Вы не выбрали объект для изменения");  
 }  
 int index = listBox1.Items.IndexOf(listBox1.SelectedItem);  
 string newValue = FieldTextBox.Text;  
 string valueName = FieldsComboBox.Text;  
 container.ChangeParameters(index, valueName, newValue);  
 ErrorFieldLabel.Text = $"{FieldsComboBox.Text}: {container.GetDevices()[index].GetType().GetProperty(FieldsComboBox.Text).GetValue(container.GetDevices()[index])}";  
 }  
  
 private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)  
 {  
 FieldsComboBox.Items.Clear();  
 int index = listBox1.Items.IndexOf(listBox1.SelectedItem);  
 Type type = container.GetDevices()[index].GetType();  
 var properties = type.GetProperties();  
 var propertiesNames = properties.Select(p => p.Name).ToArray();  
 FieldsComboBox.Items.AddRange(propertiesNames);  
   
 }  
  
 private void FieldsComboBox\_SelectedIndexChanged(object sender, EventArgs e)  
 {  
 int index = listBox1.Items.IndexOf(listBox1.SelectedItem);  
 ErrorFieldLabel.Text = $"{FieldsComboBox.Text}: {container.GetDevices()[index].GetType().GetProperty(FieldsComboBox.Text).GetValue(container.GetDevices()[index])}";  
 }  
 }  
}