Министерство науки и высшего образования Российской Федерации

Новосибирский Государственный технический университет

Кафедра автоматизированных систем управления



**Отчет по лабораторной работе № 3**

**по дисциплине «Технологии и методы программирования»**

**«**реализация программных модулей многократного использования**»**

Выполнили

студенты группы АП-026

Давыдов М.С.

Косточкин М.

г. Новосибирск

2021 г

**1. ЗАДАНИЕ**

Описание класса и все модули, созданные при выполнении работы 2, оформить как DLL. Тестирующая программа должна повторить задание работы 2, используя созданную DLL и разные варианты связывания.

1) Реализовать класс, конструктор которого создаёт меню, внешний вид которого определяется данными, находящимися во внешнем текстовом файле. Аргументом конструктора будет имя файла. Меню программы рассматривается как множество деревьев (иерархий), корнями которых являются пункты главного меню (уровень иерархии 0). Следующие уровни иерархии представляют подпункты, при этом количество уровней произвольное для каждого пункта (подпункта).

Каждая запись файла имеет следующую структуру:

Нмер\_уровня\_в\_иерархииПробелНазвание\_пунктаПробелСтатус\_пунктаПробелИмяМетода

Последовательность записей соответствует последовательности пунктов. Статус пункта определяется:

0 – пункт виден и доступен;

1 – пункт виден, но не доступен;

2 – пункт не виден.

ИмяМетода – имя обработчика клика мыши на пункте меню.

Если имя обработчика не указано, это означает, что пункт имеет подменю, которое и должно появиться, после выбора этого пункта.

2) Любым текстовым редактором создать файл, описывающий структуру меню.

3) Написать тестовую программу для проверки работы класса. Программа должна построить окно, в котором поместит меню по описанию.

4) При выборе мышью пункта срабатывает обработчик пункта или появляется подменю, если имя метода не указано. Обработчик пункта должен вызвать метод с указанным именем, который просто обозначит себя.

**2. ИСХОДНЫЙ ТЕКСТ КЛАССА И ТЕСТИРУЮЩЕЙ ПРОГРАММЫ**

|  |
| --- |
| Неявное связывание |
| using System;  using System.Windows.Forms;  using System.Reflection;  using System.Linq;  using MenuStripCreator;  namespace ProgrammingLab3  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  openFileDialog1.Filter = "Text files(\*.txt)|\*.txt|All files(\*.\*)|\*.\*";  openFileDialog1.Title = "Открыть файл с структурой меню";  openFileDialog1.FileName = "";  }  private void Button1Click(object sender, EventArgs e)  {  if (openFileDialog1.ShowDialog() == DialogResult.Cancel)  return;  string filename = openFileDialog1.FileName;  try  {  MenuCreator menu = new MenuCreator(filename, menuStrip1);  }  catch (Exception ex)  {  MessageBox.Show(ex.ToString(), "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  }  } |
| Явное связывание |
| using System;  using System.Windows.Forms;  using System.Reflection;  using System.Linq;  namespace ProgrammingLab3  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  openFileDialog1.Filter = "Text files(\*.txt)|\*.txt|All files(\*.\*)|\*.\*";  openFileDialog1.Title = "Открыть файл с структурой меню";  openFileDialog1.FileName = "";  }  private void Button1Click(object sender, EventArgs e)  {  if (openFileDialog1.ShowDialog() == DialogResult.Cancel)  return;  string filename = openFileDialog1.FileName;  try  {  Assembly asm = Assembly.LoadFrom("MenuCreatorLibrary.dll");  Type type = asm.GetTypes().FirstOrDefault(x => x.Name == "MenuCreator");  object obj = Activator.CreateInstance(type, new object[] { filename, menuStrip1 });  }  catch (Exception ex)  {  MessageBox.Show(ex.ToString(), "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  }  } |