***Лабараторна робота №4***

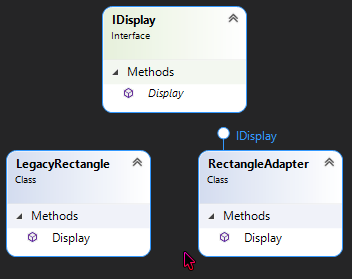
***Варіант №14***

**Завдання №1**

14. Реалізувати паттерн Адаптер для наведеної діаграми класів. Нехай існує клас Прямокутник з методом відображення, який повинен отримати параметри розмір сторін та координати розташування. Але з’явилась  необхідність, щоб клієнт передавав в метод відображення координати верхнього лівого кута та координати правого нижнього кута.

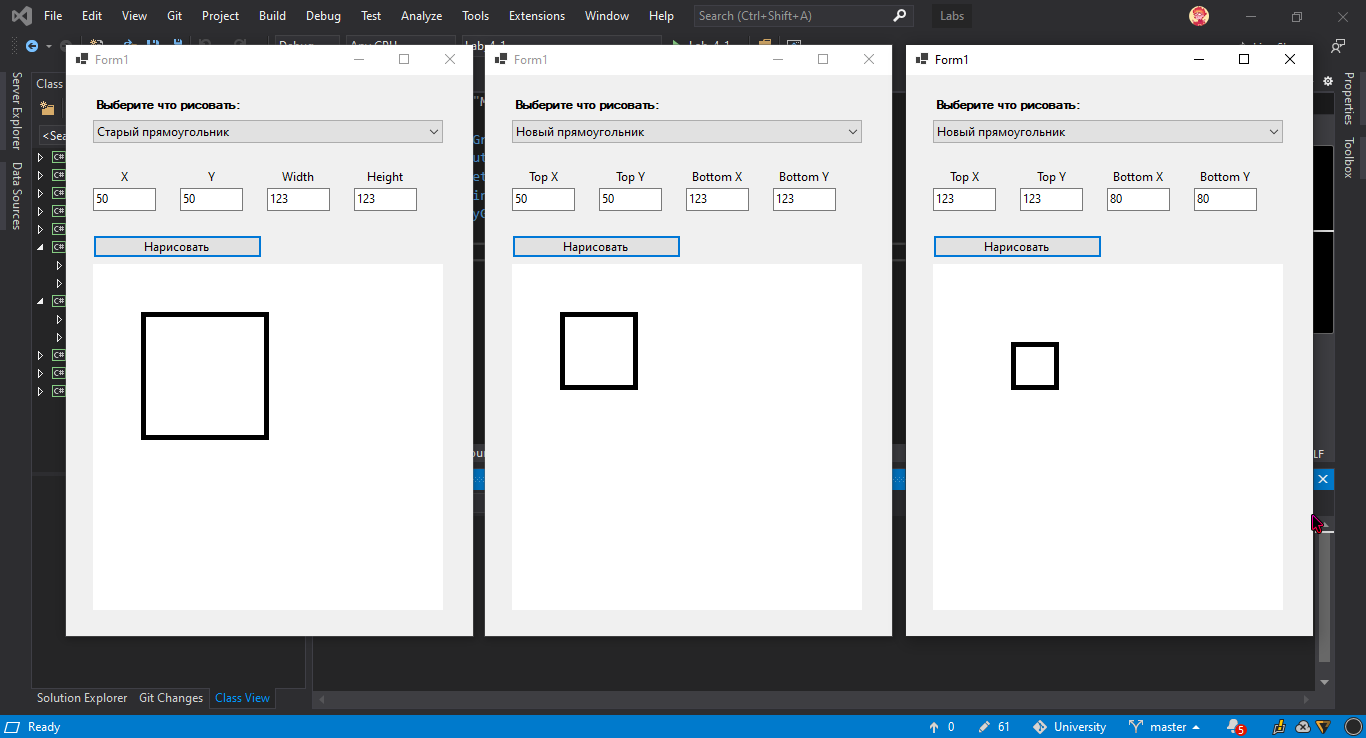
***Результати виконання завдання 1:***

*Діаграма класів додатку (рис.1):*



**Рис. 1 Діаграма класів додатку**

*Результати роботи додатку (рис.2)*



**Рис. 2 Результат виконаня додатку**

*Лістинг програмного коду:*

*\*\* MainForm.cs \*\**

using System;

using System.Drawing;

using System.Windows.Forms;

namespace Lab\_4\_1

{

public partial class MainForm : Form

{

private readonly Bitmap \_bmp;

private readonly Graphics \_g;

private readonly Pen \_pen;

private readonly LegacyRectangle \_legacyRectangle = new LegacyRectangle();

private readonly RectangleAdapter \_rectangleAdapter = new RectangleAdapter();

private string xTextBoxLastText = "";

private string yTextBoxLastText = "";

private string aTextBoxLastText = "";

private string bTextBoxLastText = "";

public MainForm()

{

InitializeComponent();

\_bmp = new Bitmap(500, 500);

\_g = Graphics.FromImage(\_bmp);

\_pen = new Pen(Color.Black, 5);

pictureBox1.Image = \_bmp;

comboBox1.Items.Add("Старый прямоугольник");

comboBox1.Items.Add("Новый прямоугольник");

comboBox1.SelectedIndex = 0;

xTextBox.Text = "0";

yTextBox.Text = "0";

aTextBox.Text = "50";

bTextBox.Text = "50";

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

switch (comboBox1.SelectedIndex)

{

case 0:

xLabel.Text = "X";

yLabel.Text = "Y";

aLabel.Text = "Width";

bLabel.Text = "Height";

break;

case 1:

xLabel.Text = "Top X";

yLabel.Text = "Top Y";

aLabel.Text = "Bottom X";

bLabel.Text = "Bottom Y";

break;

}

}

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

{

\_g.Clear(Color.White);

var x = int.Parse(xTextBox.Text);

var y = int.Parse(yTextBox.Text);

var a = int.Parse(aTextBox.Text);

var b = int.Parse(bTextBox.Text);

switch (comboBox1.SelectedIndex)

{

case 0:

var rect = \_legacyRectangle.Display(x, y, a, b);

\_g.DrawRectangle(\_pen, rect);

break;

case 1:

rect = \_rectangleAdapter.Display(x, y, a, b);

\_g.DrawRectangle(\_pen, rect);

break;

}

pictureBox1.Image = \_bmp;

}

private void xTextBox\_TextChanged(object sender, EventArgs e)

{

if (int.TryParse(xTextBox.Text, out \_))

{

xTextBoxLastText = xTextBox.Text;

}

else

{

xTextBox.Text = xTextBoxLastText;

}

}

private void yTextBox\_TextChanged(object sender, EventArgs e)

{

if (int.TryParse(yTextBox.Text, out \_))

{

yTextBoxLastText = yTextBox.Text;

}

else

{

yTextBox.Text = yTextBoxLastText;

}

}

private void aTextBox\_TextChanged(object sender, EventArgs e)

{

if (int.TryParse(aTextBox.Text, out \_))

{

aTextBoxLastText = aTextBox.Text;

}

else

{

aTextBox.Text = aTextBoxLastText;

}

}

private void bTextBox\_TextChanged(object sender, EventArgs e)

{

if (int.TryParse(bTextBox.Text, out \_))

{

bTextBoxLastText = bTextBox.Text;

}

else

{

bTextBox.Text = bTextBoxLastText;

}

}

}

}

*\*\* RectangleAdapter.cs \*\**

using System;

using System.Drawing;

namespace Lab\_4\_1

{

interface IDisplay

{

public Rectangle Display(int topX, int topY, int bottomX, int bottomY);

}

class RectangleAdapter : IDisplay

{

public Rectangle Display(int topX, int topY, int bottomX, int bottomY)

{

if (bottomX < topX)

{

int tmp = topX;

topX = bottomX;

bottomX = tmp;

}

if (bottomY < topY)

{

int tmp = topY;

topY = bottomY;

bottomY = tmp;

}

int width = bottomX - topX;

int height = bottomY - topY;

return new LegacyRectangle().Display(topX, topY, width, height);

}

}

class LegacyRectangle

{

public Rectangle Display(int x, int y, int width, int height)

{

return new Rectangle(x, y, width, height);

}

}

}

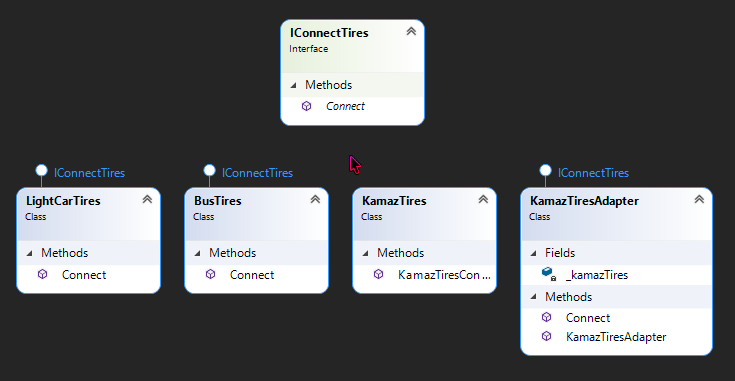
**Завдання №2**

Застосувати патерн Адаптер для заданої предметної області

(14. Шиномонтаж)

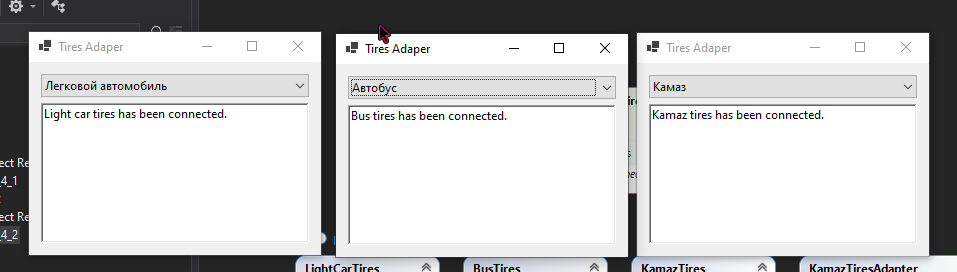
***Результати виконання завдання 2:***

*Діаграма класів додатку (рис.3):*



**Рис. 3 Діаграма класів додатку**

*Результати роботи додатку (рис.4)*



**Рис. 4 Результат виконаня додатку**

*Лістинг програмного коду:*

*\*\* MainForm.cs \*\**

using System;

using System.Windows.Forms;

namespace Lab\_4\_2

{

public partial class MainForm : Form

{

private IConnectTires \_tires;

public MainForm()

{

InitializeComponent();

comboBox1.Items.Add("Легковой автомобиль");

comboBox1.Items.Add("Автобус");

comboBox1.Items.Add("Камаз");

comboBox1.SelectedIndex = 0;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

switch (comboBox1.SelectedIndex)

{

case 0:

\_tires = new LightCarTires();

richTextBox1.Text = \_tires.Connect();

break;

case 1:

\_tires = new BusTires();

richTextBox1.Text = \_tires.Connect();

break;

case 2:

\_tires = new KamazTiresAdapter(new KamazTires());

richTextBox1.Text = \_tires.Connect();

break;

}

}

}

}

*\*\* TiresAdapter.cs \*\**

namespace Lab\_4\_2

{

public interface IConnectTires

{

string Connect();

}

class LightCarTires : IConnectTires

{

public string Connect()

{

return "Light car tires has been connected.";

}

}

class BusTires : IConnectTires

{

public string Connect()

{

return "Bus tires has been connected.";

}

}

class KamazTires

{

public string KamazTiresConnect()

{

return "Kamaz tires has been connected.";

}

}

class KamazTiresAdapter : IConnectTires

{

private readonly KamazTires \_kamazTires;

public KamazTiresAdapter(KamazTires kamazTires)

{

\_kamazTires = kamazTires;

}

public string Connect()

{

return \_kamazTires.KamazTiresConnect();

}

}

}