**Лабораторна робота №4**

Л.Р. №5-6: ДИНАМІЧНА ІДЕНТИФІКАЦІЯ ТИПІВ

Квартюк Олексій

Варіант 12

5.6. Завдання для виконання.

1. Створити новий проект за шаблоном Windows Forms Application.

2. На форму додати елемент керування TreeView.

3. Описати клас, забезпечивши наявність:

 4-х властивостей різних типів, одна з яких мала б тип колекції;

 3-х методів;

 2-х конструкторів.

4. Описати метод, який би виводив перелік всіх властивостей (разом з типом та

значенням) об’єкта описаного класу у TreeView

12.class Принтер.

**Код:**

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;

using System.Reflection;

namespace lab5

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

}

private void MainForm\_Load(object sender, EventArgs e)

{

List<string> supportedPaperTypes = new List<string> { "A4", "A3", "Letter", "Legal", "Photo" };

Printer printer = new Printer("HP LaserJet Pro", "Laser", supportedPaperTypes, 27);

DisplayObjectProperties(printer, this.treeView1);

}

private void DisplayObjectProperties(object obj, System.Windows.Forms.TreeView treeView)

{

treeView.Nodes.Clear();

Type type = obj.GetType();

TreeNode rootNode = new TreeNode($"{type.Name} Properties");

treeView.Nodes.Add(rootNode);

PropertyInfo[] properties = type.GetProperties(BindingFlags.Public | BindingFlags.Instance);

foreach (PropertyInfo property in properties)

{

object value = property.GetValue(obj);

string valueStr;

if (value is IEnumerable<object> collection && !(value is string))

{

TreeNode propertyNode = new TreeNode($"{property.Name} (Type: {property.PropertyType.Name})");

rootNode.Nodes.Add(propertyNode);

int index = 0;

foreach (var item in (System.Collections.IEnumerable)value)

{

propertyNode.Nodes.Add(new TreeNode($"[{index}]: {item}"));

index++;

}

}

else

{

valueStr = value != null ? value.ToString() : "null";

TreeNode propertyNode = new TreeNode($"{property.Name} (Type: {property.PropertyType.Name}, Value: {valueStr})");

rootNode.Nodes.Add(propertyNode);

}

}

TreeNode methodsNode = new TreeNode("Methods");

rootNode.Nodes.Add(methodsNode);

MethodInfo[] methods = type.GetMethods(BindingFlags.Public | BindingFlags.Instance | BindingFlags.DeclaredOnly);

foreach (MethodInfo method in methods)

{

string parameters = string.Join(", ", method.GetParameters().Select(p => $"{p.ParameterType.Name} {p.Name}"));

methodsNode.Nodes.Add(new TreeNode($"{method.ReturnType.Name} {method.Name}({parameters})"));

}

treeView.ExpandAll();

}

}

public class Printer

{

public string Model { get; set; }

public string PrinterType { get; set; }

public List<string> SupportedPaperTypes { get; set; }

public int PagesPerMinute { get; set; }

public Printer()

{

Model = "Unknown";

PrinterType = "Unknown";

SupportedPaperTypes = new List<string>();

PagesPerMinute = 0;

}

public Printer(string model, string printerType, List<string> supportedPaperTypes, int pagesPerMinute)

{

Model = model;

PrinterType = printerType;

SupportedPaperTypes = supportedPaperTypes;

PagesPerMinute = pagesPerMinute;

}

public void Print(string document)

{

Console.WriteLine($"Printing document: {document}");

}

public bool AddPaper(int pages)

{

if (pages > 0)

{

Console.WriteLine($"Added {pages} pages to the printer");

return true;

}

return false;

}

public string GetStatus()

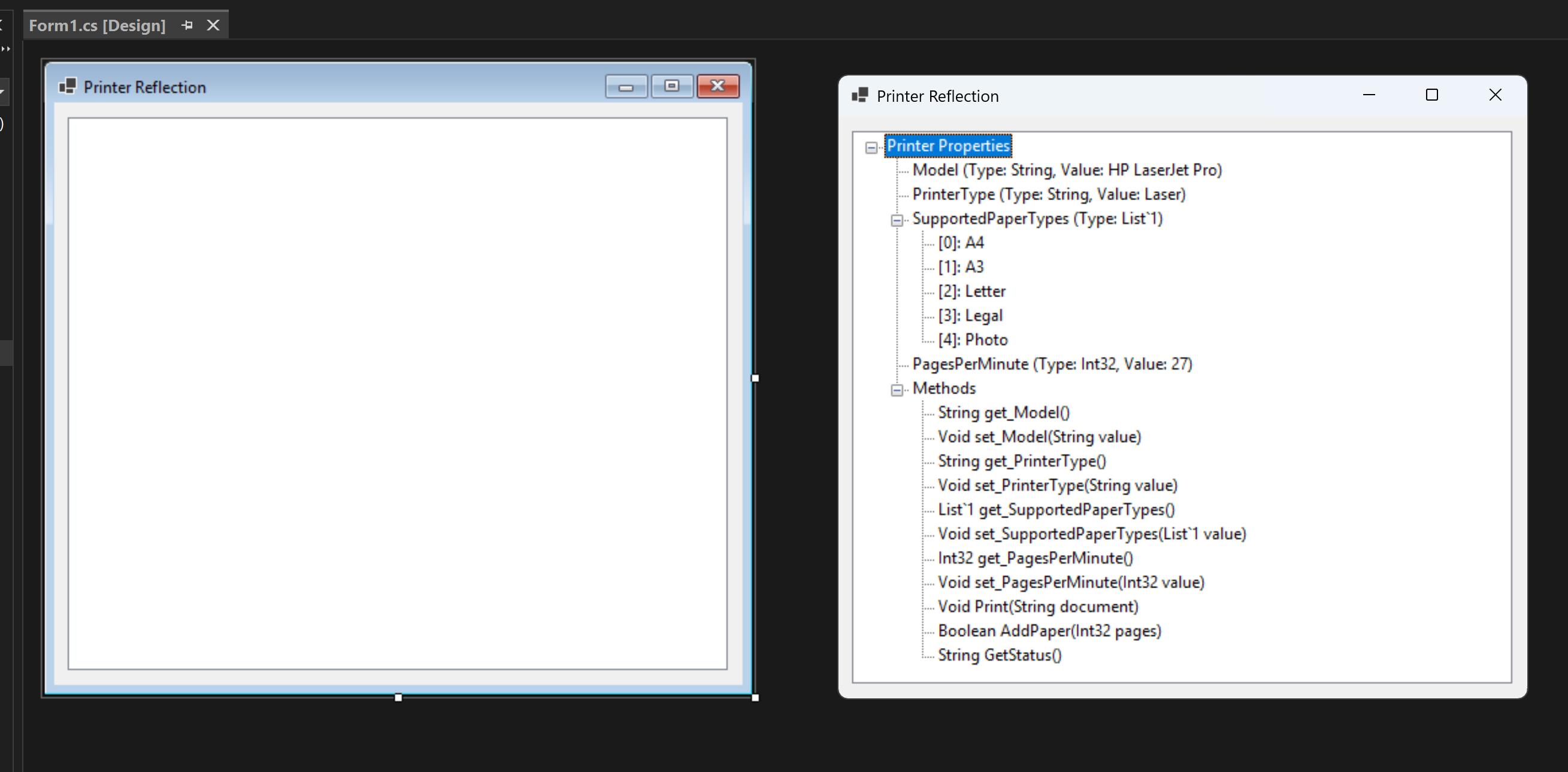
{

return $"Printer {Model} is ready. Pages per minute: {PagesPerMinute}";

}

}

}



**Висновок:** У рамках завдання було створено Windows Forms застосунок із використанням елемента керування TreeView. Розроблено клас із чотирма властивостями різних типів, включно з колекцією, двома конструкторами та трьома методами. Реалізовано метод для виведення всіх властивостей об’єкта (разом із їхніми типами та значеннями) у TreeView, що забезпечує наочне представлення структури даних.