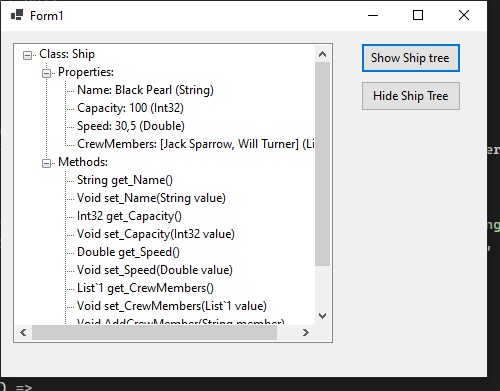
**ЛАБОРАТОРНА РОБОТА № 5-6. ЗВІТ. ДИНАМІЧНА ІДЕНТИФІКАЦІЯ ТИПІВ**

**Завдання (варіант 5 class Корабель):**

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.

**Хід роботи**

Виконання:



Деякі пояснення до коду:

Рефлексія в цьому коді використовується для отримання інформації про **властивості** та **методи** об'єкта **без явного знання його типу** на етапі компіляції.

private void DisplayShipProperties(object obj)

{

treeViewShips.Nodes.Clear();

Type type = obj.GetType();

var rootNode = new TreeNode($"Class: {type.Name}");

// Додаємо властивості

rootNode.Nodes.Add(new TreeNode("Properties:", type.**GetProperties()**

.Select(prop => new TreeNode($"{prop.Name}: {FormatValue(**prop.GetValue(obj)**)} ({prop.PropertyType.Name})"))

.ToArray()));

// Додаємо методи

rootNode.Nodes.Add(new TreeNode("Methods:", type.**GetMethods**(BindingFlags.DeclaredOnly | BindingFlags.Public | BindingFlags.Instance)

.Select(method => new TreeNode($"{method.ReturnType.Name} {method.Name}({string.Join(", ", method.**GetParameters**().Select(p => $"{p.ParameterType.Name} {p.Name}"))})"))

.ToArray()));

treeViewShips.Nodes.Add(rootNode);

treeViewShips.ExpandAll();

}

*type.GetProperties()* // GetProperties() для отримання всіх публічних властивостей класу.

*prop.GetValue(obj)* // отримує значення властивості

Отримання методів класу: *type.****GetMethods****(BindingFlags.DeclaredOnly | BindingFlags.Public | BindingFlags.Instance)*

// GetMethods(...) для отримання всіх публічних методів, визначених безпосередньо в класі (DeclaredOnly).

// method.GetParameters() дозволяє отримати список параметрів кожного методу.

Висновки: У ході виконання цієї лабораторної роботи я реалізувала клас відповідно до свого варіанту та забезпечила рефлексійне виведення його властивостей та методів.