**Лабораторна робота №5**

ДИНАМІЧНА ІДЕНТИФІКАЦІЯ ТИПІВ

Завдання:

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.

1. Описати метод, який би виводив перелік всіх властивостей (разом з типом та значенням) об’єкта описаного класу у TreeView.

Варіант завдання - class Студент.

Хід роботи:

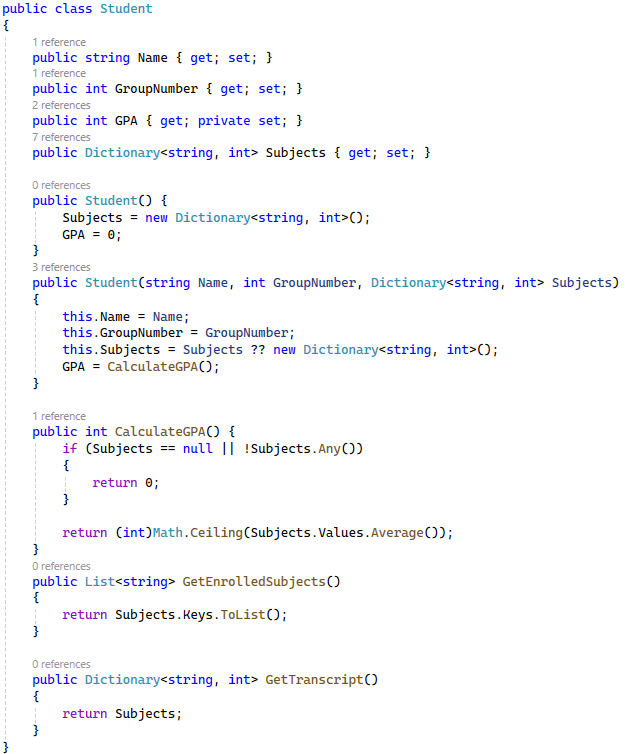
Для цього завдання спочатку створимо клас згідно варіанту. Пропишемо такі 4 властивості як Ім’я та прізвище (string), номер групи (integer), середній бал (double), курси з оцінками, на які студент записаний (dictionary<string, integer>). У цьому класі є три методи:

1. Розрахунок середнього балу.
2. Отримання курсів, на які записаний студент.
3. Отримання витягу з оцінками.

Також є два конструктори:

1. Конструктор за замовчуванням. Цей конструктор використовується, коли створюється об’єкт Student без надання початкових даних. Він ініціалізує властивість Subjects порожнім словником і встановлює GPA на 0. Цей конструктор корисний, так як нам потрібно створити об’єкт Student і пізніше призначити значення властивостям.
2. Параметризований конструктор. Цей конструктор використовується, коли ми вже хочемо створити об’єкт Student із певними значеннями для зазначених параметрів.

Код класу:



Далі пропишемо метод, який би виводив перелік всіх властивостей (разом з типом та значенням) об’єкта описаного класу у TreeView.



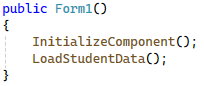


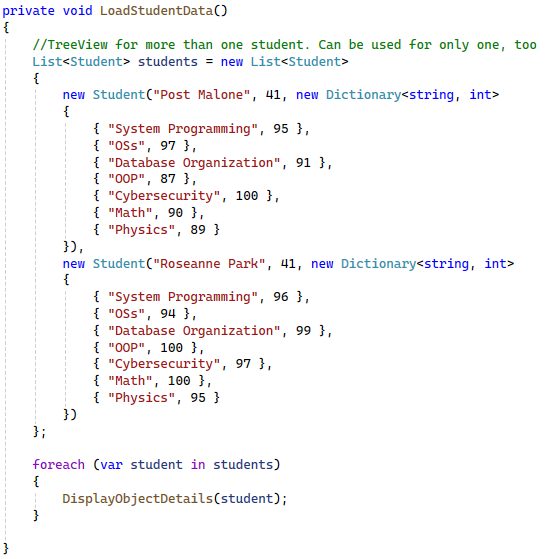
Метод DisplayObjectDetails() приймає об’єкт як параметр і динамічно відображає його властивості та методи в елементі керування TreeView. По-перше, він отримує тип об’єкта за допомогою obj.GetType() і створює кореневий вузол у TreeView, позначений повним ім’ям класу об’єкта та “Details”. Потім він перебирає властивості об’єкта за допомогою рефлексії (type.GetProperties()) і створює вузол дерева для кожної властивості. У випадку властивості словнику, створюється підвузол, що відображає ключі та значення словника. Для інших типів властивостей він додає вузол із назвою, типом і значенням властивості. Властивості згруповані у вузлі “Properties”.

Наступним метод обробляє публічні методи екземпляра класу за допомогою type.GetMethods(). Для кожного методу додається вузол із назвою методу, а потім ще один вузол із переліком параметрів методу та їх типів, якщо такі є. Ці вузли згруповані під вузлом “Methods”.

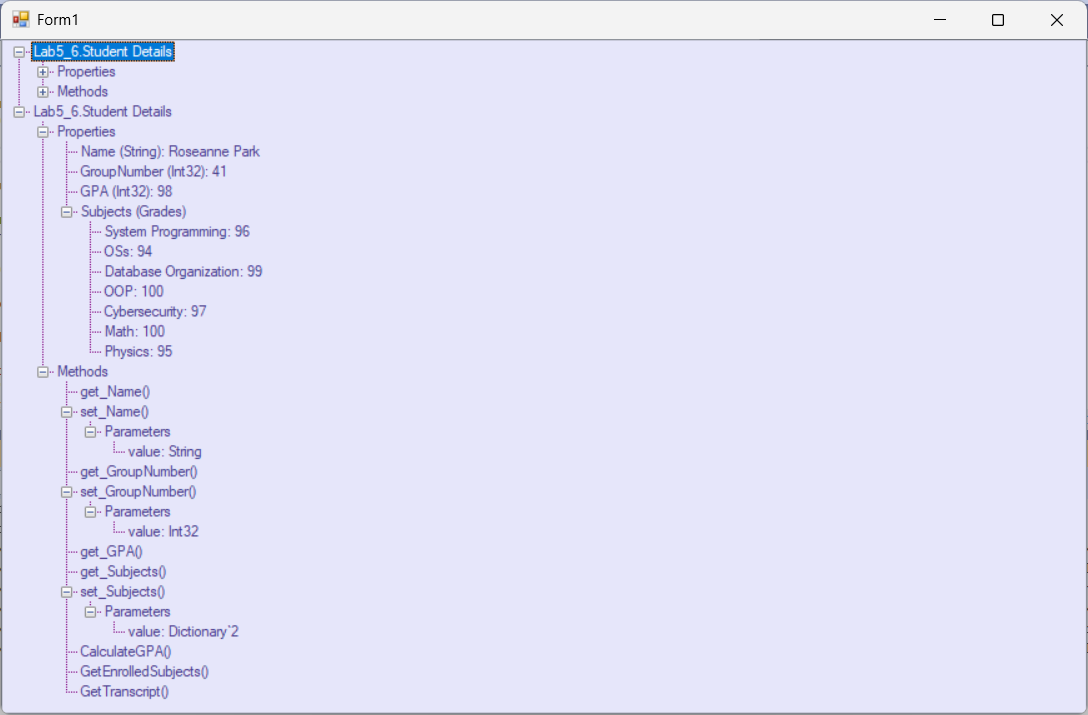
У кінці, вузли “Properties” та “Methods” додаються до кореневого вузла, який потім додається до TreeView. Команда treeView.ExpandAll() забезпечує розгортання всіх вузлів для зручного перегляду даних.

Також пропишемо метод, в якому ініціалізуємо створення об’єктів класу Student і додамо його до форми:





Після запуску програми отримаємо форму:



**Висновки**: у ході лабораторної роботи було розглянуто поняття рефлексії. Було досліджено застосування рефлексії. Були отримані практичні навички з отримання відомостей про методи та виклику методів за допомогою рефлексії.