МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»

Навчально науковий інститут інформаційних технологій і робототехніки
Кафедра комп'ютерних та інформаційних технологій і систем

і

Лабораторна робота № 1

з навчальної дисципліни
«Технології на платформі NET»

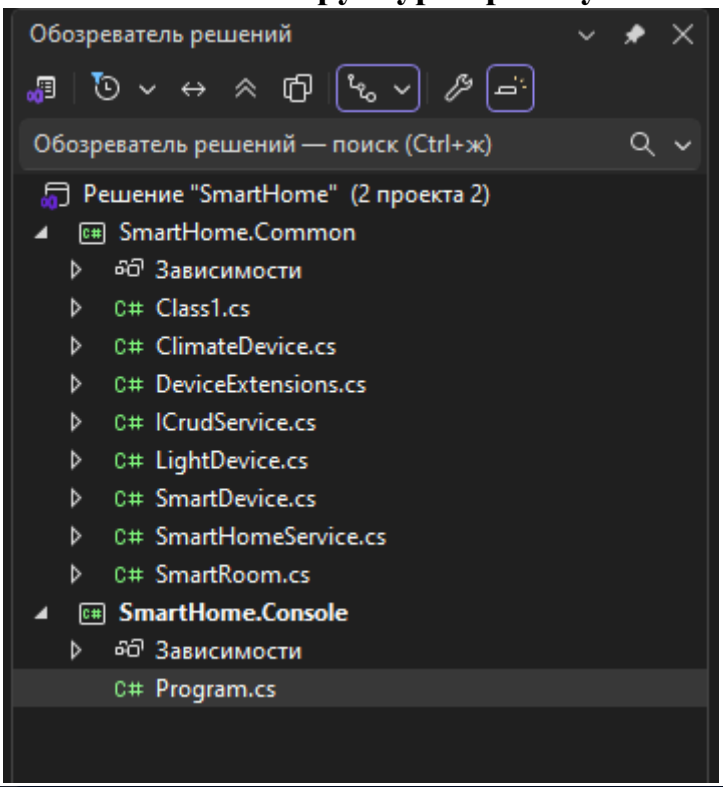
Виконав:
Студент групи 403-ТН
Солонецький Роман Миколайович

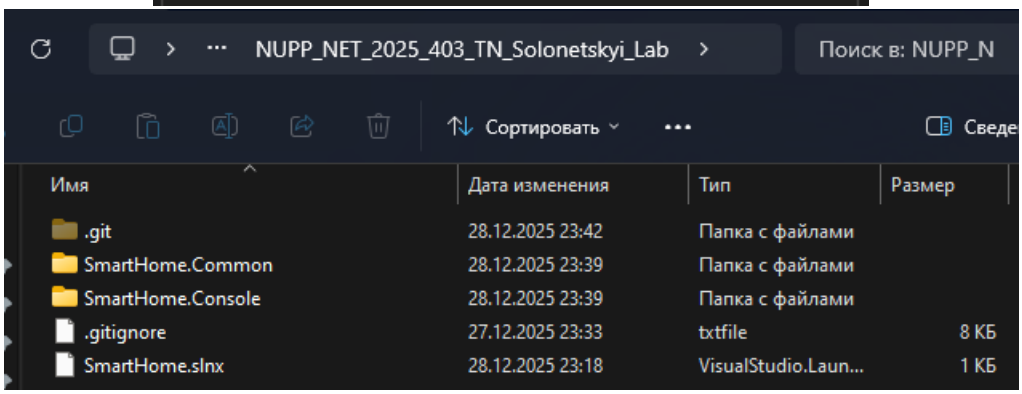
Перевірив:
Тютюнник Петро Богданович

Полтава 2025

Тема: Основи програмування у мові C#. ООП у мові C#. Базові конструкції та Типи

Файлова структура проєкту





Реалізація базових конструкцій C#

LightDevice.cs | SmartHomeService.cs | Program.cs

SmartHome.Common | SmartHome.Common.LightDevice

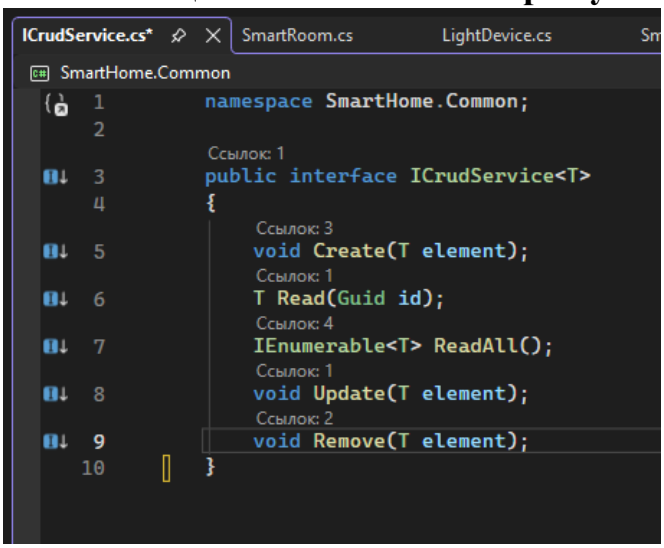
```
1   namespace SmartHome.Common;
2
    Ссылок: 4
3   public class LightDevice : SmartDevice
4   {
        Ссылок: 1
5       public int Brightness { get; set; }
        Ссылок: 1
6       public string Color { get; set; }
        Ссылок: 1
7       public string BulbType { get; set; }
8
        Ссылок: 0
9       public LightDevice() : base() { }
10
        Ссылок: 1
11      public LightDevice(string name, int brightness = 50) : base(name)
12      {
13          Brightness = brightness;
14          Color = "White";
15          BulbType = "LED";
16      }
17  }
```

SmartRoom.cs | LightDevice.cs | SmartHomeService.cs | Program.cs

SmartHome.Common | SmartHome.Common.SmartRo

```
1   namespace SmartHome.Common;
2
    Ссылок: 1
3   public class SmartRoom
4   {
        Ссылок: 0
5       public Guid Id { get; set; } = Guid.NewGuid();
        Ссылок: 1
6       public string RoomName { get; set; }
        Ссылок: 1
7       public int Floor { get; set; }
        Ссылок: 1
8       public double Area { get; set; }
9
        Ссылок: 0
10      public SmartRoom(string name, int floor, double area)
11      {
12          RoomName = name;
13          Floor = floor;
14          Area = area;
15      }
16  }
```

Реалізація Generic CRUD сервісу




```
Ссылок: 1
public class SmartHomeService<T> : ICrudService<T> where T : class
{
    private List<T> _items = new List<T>();

    Ссылок: 3
    public void Create(T element) => _items.Add(element);
    Ссылок: 4
    public IEnumerable<T> ReadAll() => _items;

    Ссылок: 1
    public T Read(Guid id)
    {
        return _items.FirstOrDefault(x => (Guid)x.GetType().GetProperty("Id")?.GetValue(x) == id);
    }

    Ссылок: 1
    public void Update(T element) { }
    Ссылок: 2
    public void Remove(T element) => _items.Remove(element);
```

Реалізовано сервіс, що працює з вбудованою колекцією `List<T>` та забезпечує повний цикл роботи з даними

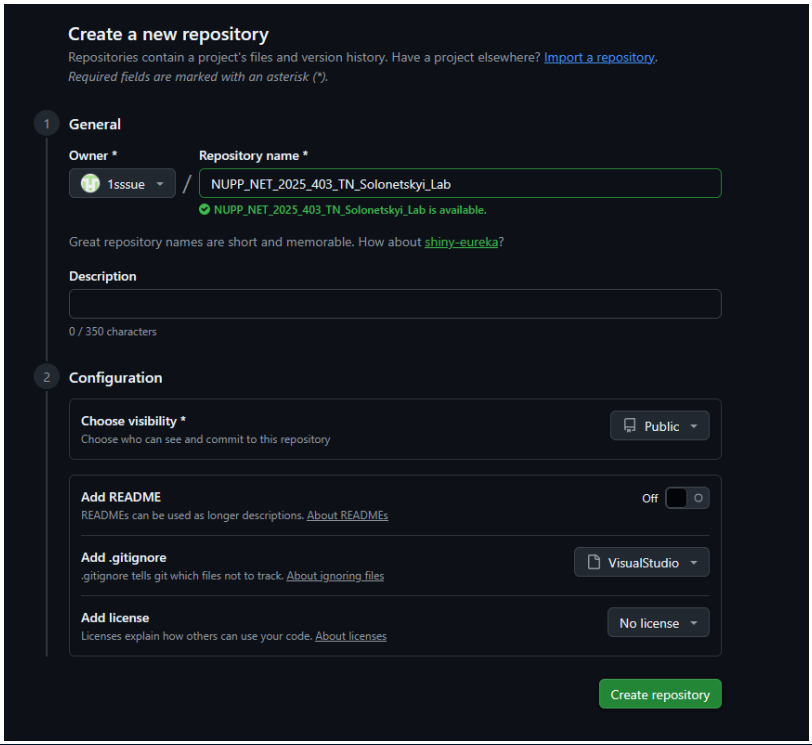
Результати виконання додаткового завдання

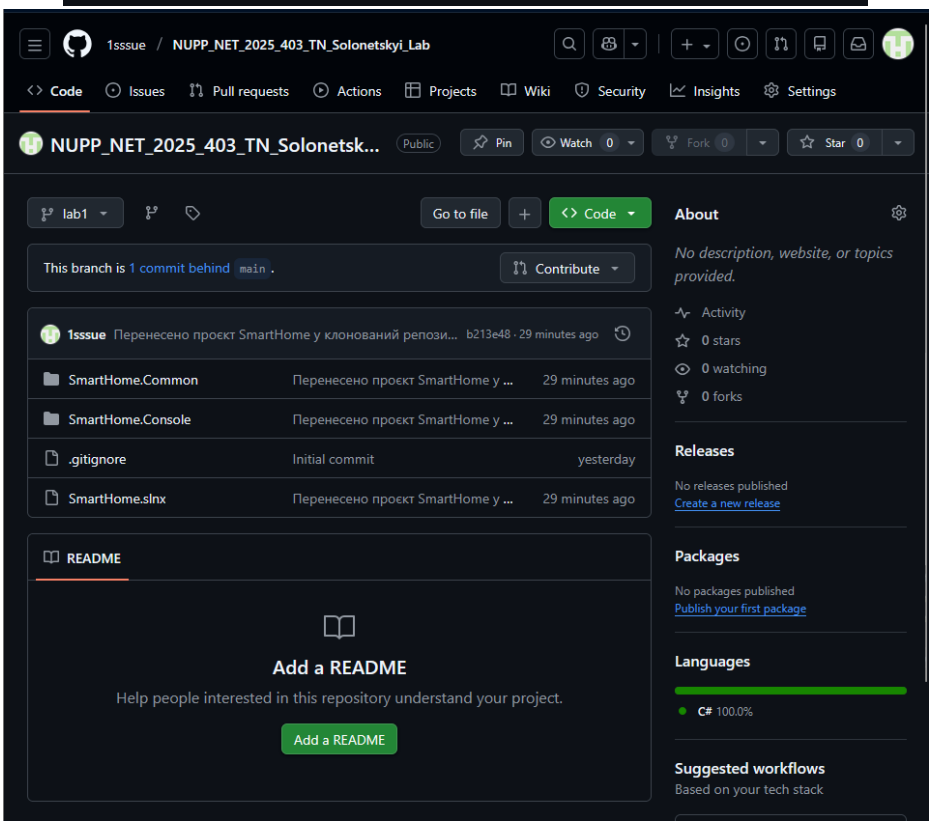
```
// Додаткове завдання: Збереження у файл
Ссылок: 1
public void Save(string filePath)
{
    string json = JsonSerializer.Serialize(_items);
    File.WriteAllText(filePath, json);
}

// Додаткове завдання: Завантаження з файлу
Ссылок: 1
public void Load(string filePath)
{
    if (File.Exists(filePath))
    {
        string json = File.ReadAllText(filePath);
        // Додаємо обробку випадку, коли файл пустий
        _items = JsonSerializer.Deserialize<List<T>>(json) ?? new List<T>();
        Console.WriteLine("Дані успішно завантажені.");
    }
}
```

Реалізовано збереження даних у форматі JSON та їх завантаження з файлу для забезпечення постійного зберігання стану системи.

Робота з Git та GitHub





Контрольні питання

1. Що таке класи та структури? Яка різниця між класом та структурою? Яка різниця між value та reference типами

Класи — це посилальні типи (reference types), що зберігаються в купі; структури — типи значень (value types), що зберігаються в стеку.

2. Що таке абстрактний клас? Яка різниця між звичайним та абстрактним класом?

Це клас-шаблон, на основі якого не можна створити об'єкт; він може містити методи без реалізації для обов'язкового перевизначення нащадками.

3. Що таке інтерфейс? Яка різниця між абстрактним класом та інтерфейсом?

Інтерфейс — це «контракт» поведінки без стану, а клас може реалізувати багато інтерфейсів. Наслідування можливе лише від одного абстрактного класу.

4. Що таке наслідування? Які є модифікатори доступу у мові C#?

Наслідування - це механізм ООП, що дозволяє створювати новий клас на основі вже існуючого.

public, private, protected, internal та їх комбінації, що регулюють видимість елементів коду.

5. Що таке статичні поля та методи? Яка різниця між звичайними та статичними полями та методами?

Це елементи, що належать самому класу, а не його екземплярам; вони існують в єдиному екземплярі для всіх об'єктів класу.

6. Що таке делегати? Які види делегатів існують?

Це типи, що описують посилання на методи з певною сигнатурою; існують одиночні, групові та вбудовані (Action, Func, Predicate) делегати.