Programowanie Obiektowe

Oleksander Mikhov, Yehor Vysotskyi

Radeo30001@gmail.com,

Clarn@clarncc.cc

143075, 143662

14.09.2023

https://github.com/Clarnc/ZOO

Added classes in 1-st:
AfricanElephantScreen

ChimpanzeeScreen

PolarBearScreen

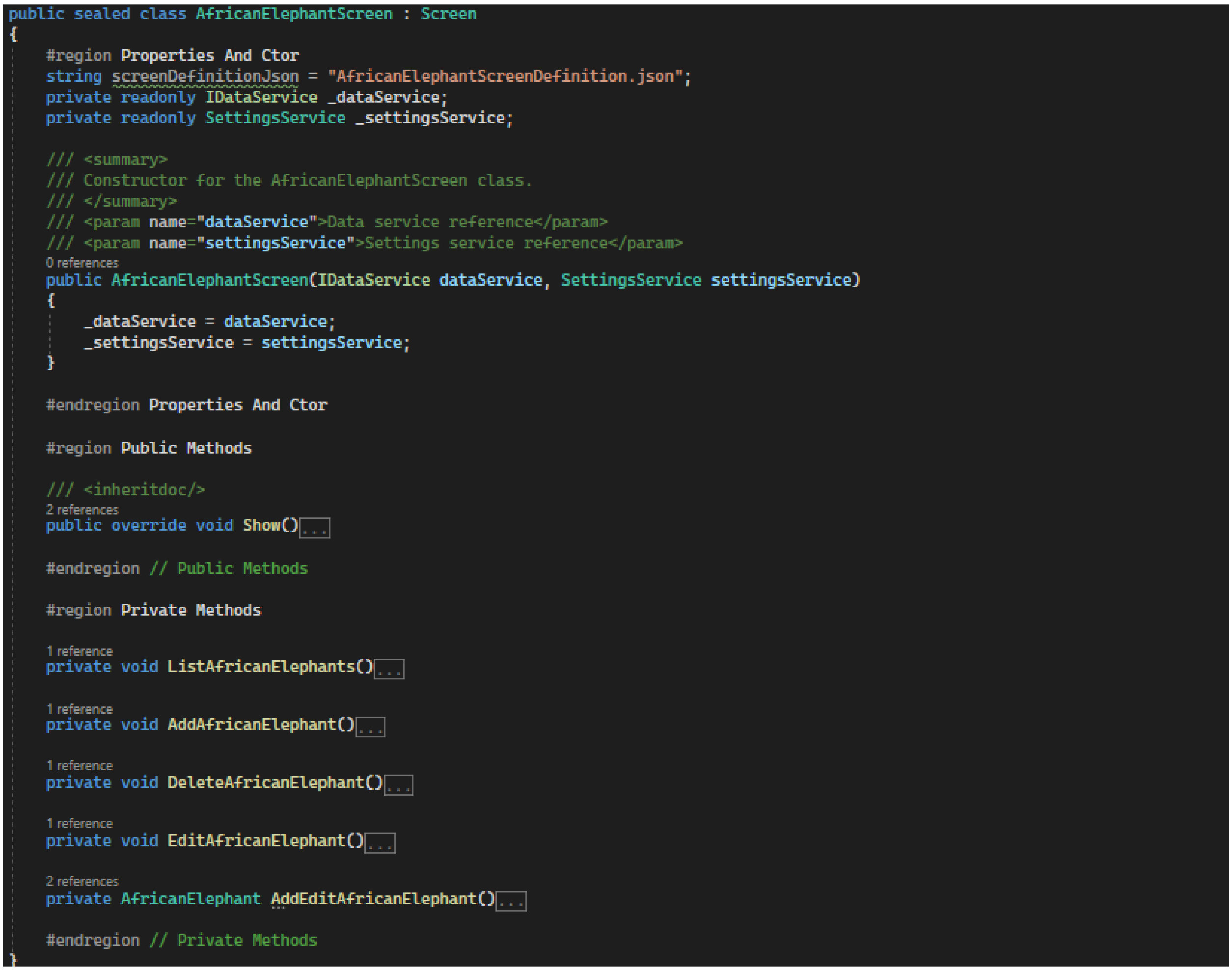
SettingsScreen

SettingsScreenChoices

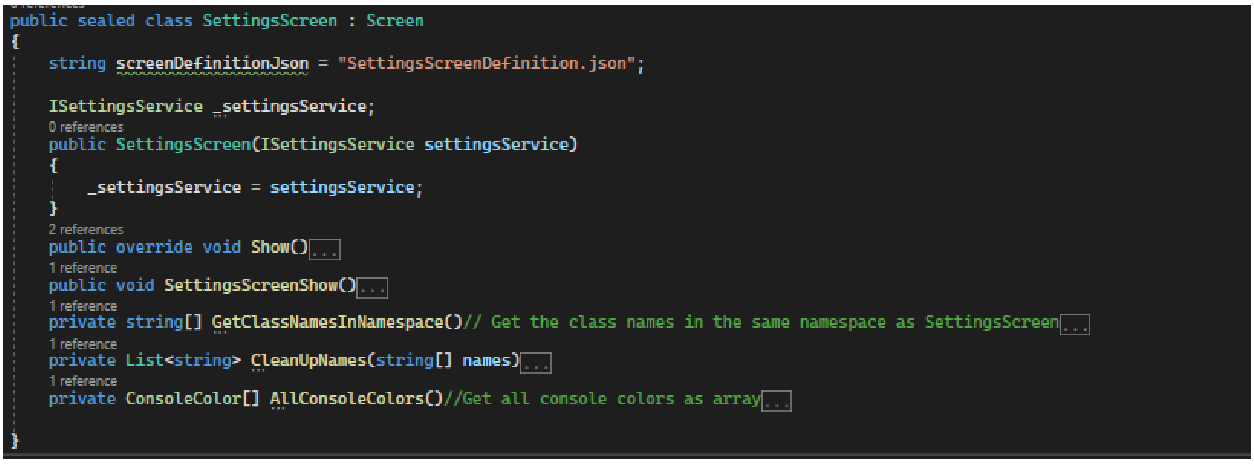
AfricanElephantScreenChoices

ChimpanzeeScreenChoices

PolarBearScreenChoices



```
public sealed class AfricanElephantScreen : Screen
{
    #region Properties And Ctor
    string screenDefinitionJson = "AfricanElephantScreenDefinition.json";
    private readonly IDataService _dataService;
    private readonly SettingsService _settingsService;

    /// <summary>
    /// Constructor for the AfricanElephantScreen class.
    /// </summary>
    /// <param name="dataService">Data service reference</param>
    /// <param name="settingsService">Settings service reference</param>
    0 references
    public AfricanElephantScreen(IDataService dataService, SettingsService settingsService)
    {
        _dataService = dataService;
        _settingsService = settingsService;
    }

    #endregion Properties And Ctor

    #region Public Methods

    /// <inheritdoc/>
    2 references
    public override void Show()...

    #endregion // Public Methods

    #region Private Methods

    1 reference
    private void ListAfricanElephants()...

    1 reference
    private void AddAfricanElephant()...

    1 reference
    private void DeleteAfricanElephant()...

    1 reference
    private void EditAfricanElephant()...

    2 references
    private AfricanElephant AddEditAfricanElephant()...

    #endregion // Private Methods
}
```



```
public sealed class SettingsScreen : Screen
{
    string screenDefinitionJson = "SettingsScreenDefinition.json";

    ISettingsService _settingsService;
    0 references
    public SettingsScreen(ISettingsService settingsService)
    {
        _settingsService = settingsService;
    }
    2 references
    public override void Show()...
    1 reference
    public void SettingsScreenShow()...
    1 reference
    private string[] GetClassNamesInNamespace()// Get the class names in the same namespace as SettingsScreen...
    1 reference
    private List<string> CleanUpNames(string[] names)...
    1 reference
    private ConsoleColor[] AllConsoleColors()//Get all console colors as array...

}
```

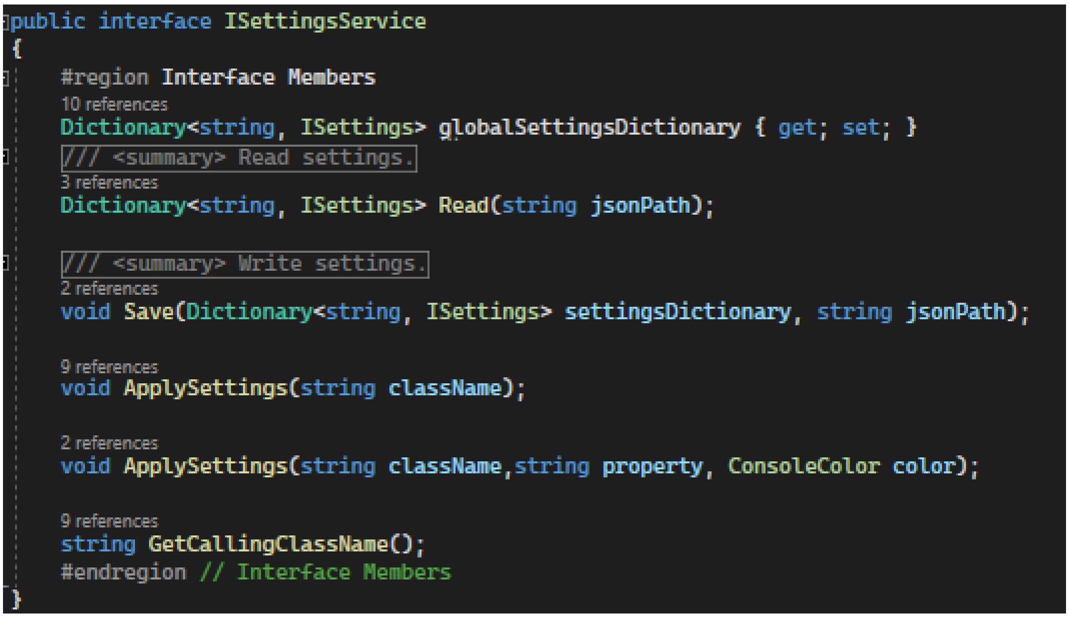
ISettings

Settings

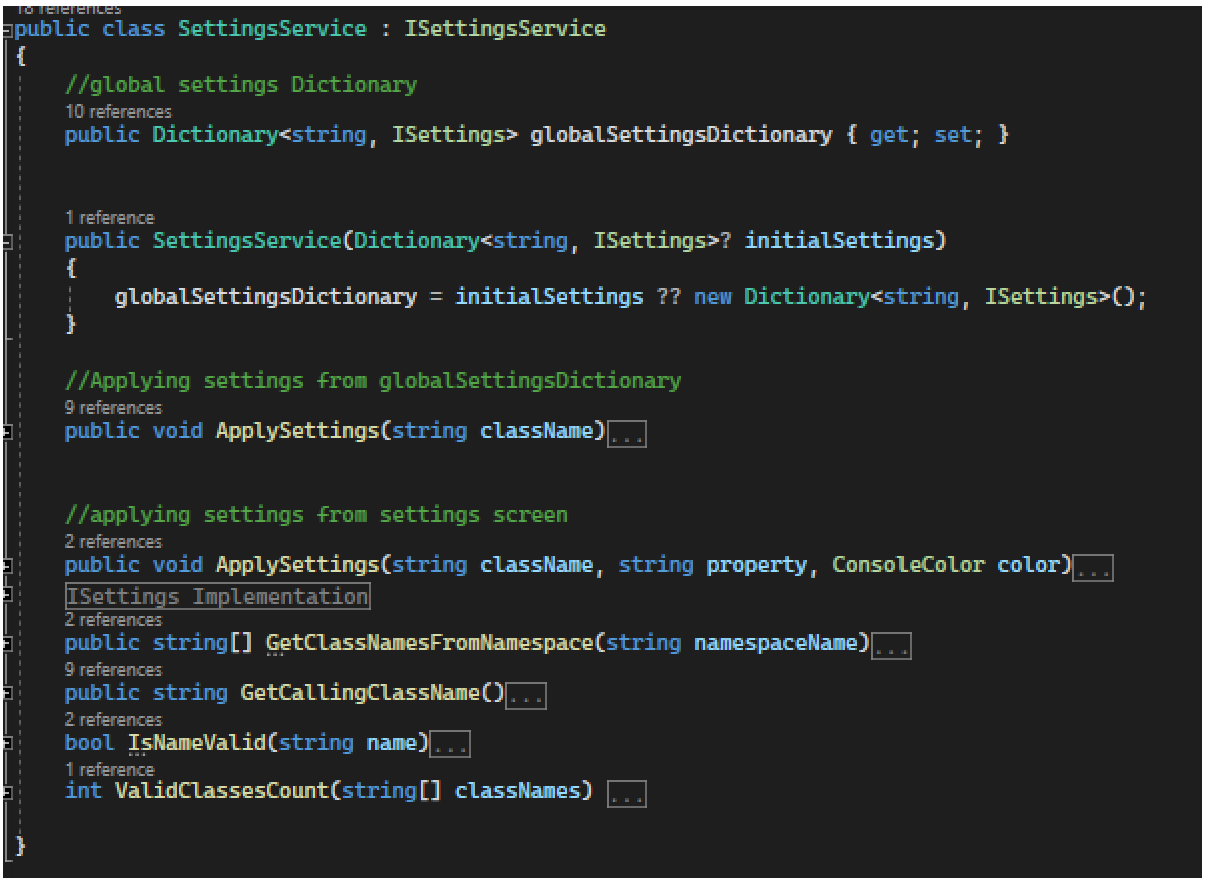
ISettingsService

SettingsService

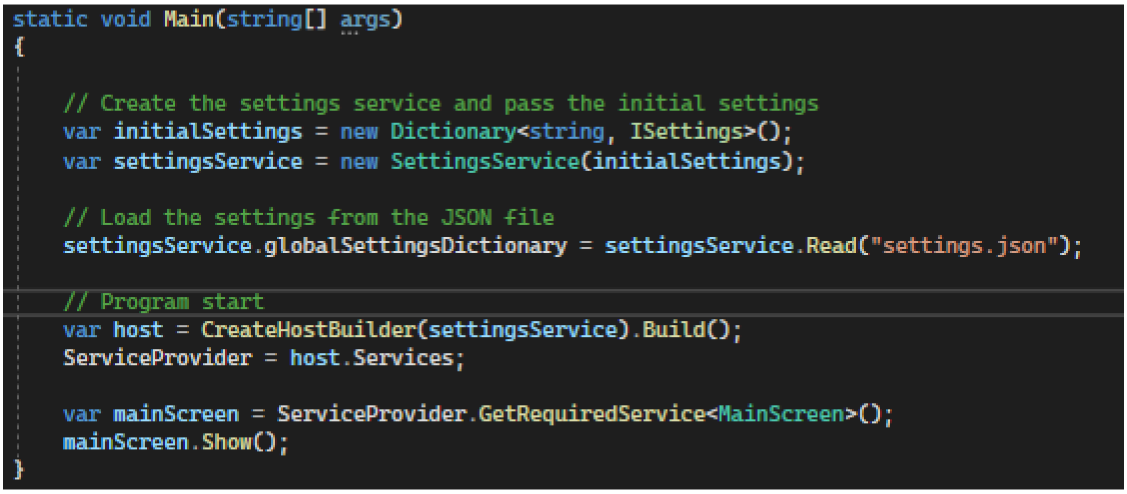
Program



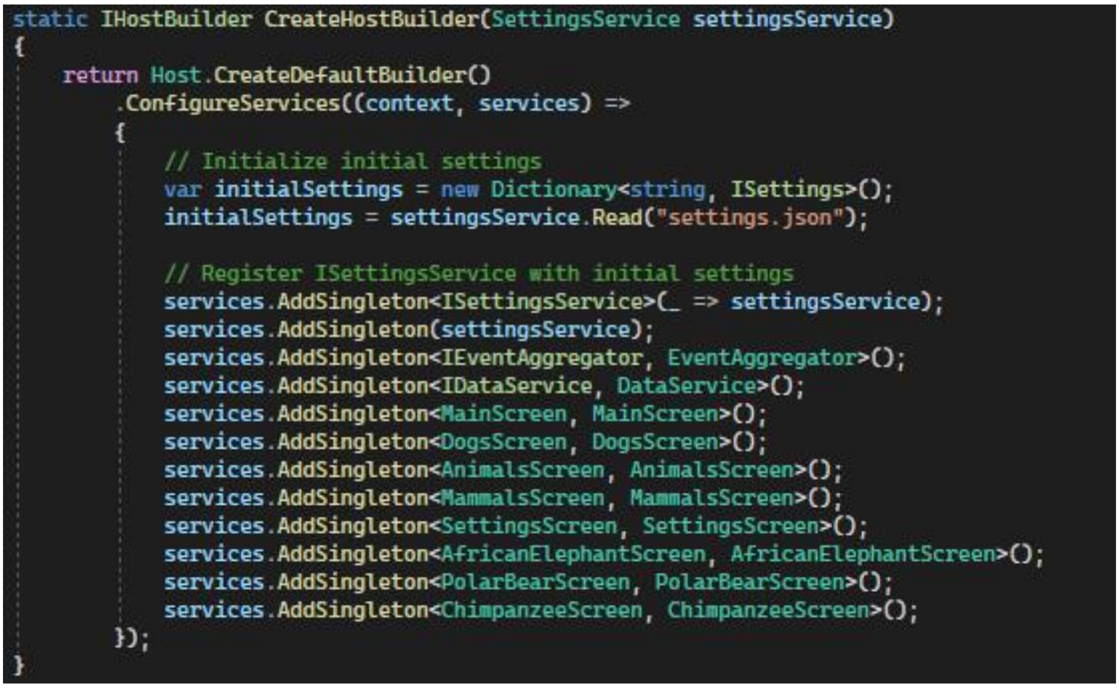
```
public interface ISettingsService
{
    #region Interface Members
    10 references
    Dictionary<string, ISettings> globalSettingsDictionary { get; set; }
    /// <summary> Read settings.
    3 references
    Dictionary<string, ISettings> Read(string jsonPath);

    /// <summary> Write settings.
    2 references
    void Save(Dictionary<string, ISettings> settingsDictionary, string jsonPath);

    9 references
    void ApplySettings(string className);

    2 references
    void ApplySettings(string className,string property, ConsoleColor color);

    9 references
    string GetCallingClassName();
    #endregion // Interface Members
}
```



```
18 references
public class SettingsService : ISettingsService
{
    //global settings Dictionary
    10 references
    public Dictionary<string, ISettings> globalSettingsDictionary { get; set; }


    1 reference
    public SettingsService(Dictionary<string, ISettings>? initialSettings)
    {
        globalSettingsDictionary = initialSettings ?? new Dictionary<string, ISettings>();
    }

    //Applying settings from globalSettingsDictionary
    9 references
    public void ApplySettings(string className)...


    //applying settings from settings screen
    2 references
    public void ApplySettings(string className, string property, ConsoleColor color)...
    ISettings Implementation
    2 references
    public string[] GetClassNamesFromNamespace(string namespaceName)...
    9 references
    public string GetCallingClassName()...
    2 references
    bool IsNameValid(string name)...
    1 reference
    int ValidClassesCount(string[] classNames) ...

}
```



```
static void Main(string[] args)
{

    // Create the settings service and pass the initial settings
    var initialSettings = new Dictionary<string, ISettings>();
    var settingsService = new SettingsService(initialSettings);

    // Load the settings from the JSON file
    settingsService.globalSettingsDictionary = settingsService.Read("settings.json");

    // Program start
    var host = CreateHostBuilder(settingsService).Build();
    ServiceProvider = host.Services;

    var mainScreen = ServiceProvider.GetRequiredService<MainScreen>();
    mainScreen.Show();
}
```



```
static IHostBuilder CreateHostBuilder(SettingsService settingsService)
{
    return Host.CreateDefaultBuilder()
        .ConfigureServices((context, services) =>
        {
            // Initialize initial settings
            var initialSettings = new Dictionary<string, ISettings>();
            initialSettings = settingsService.Read("settings.json");

            // Register ISettingsService with initial settings
            services.AddSingleton<ISettingsService>(_ => settingsService);
            services.AddSingleton(settingsService);
            services.AddSingleton<IEventAggregator, EventAggregator>();
            services.AddSingleton<IDataService, DataService>();
            services.AddSingleton<MainScreen, MainScreen>();
            services.AddSingleton<DogsScreen, DogsScreen>();
            services.AddSingleton<AnimalsScreen, AnimalsScreen>();
            services.AddSingleton<MammalsScreen, MammalsScreen>();
            services.AddSingleton<SettingsScreen, SettingsScreen>();
            services.AddSingleton<AfricanElephantScreen, AfricanElephantScreen>();
            services.AddSingleton<PolarBearScreen, PolarBearScreen>();
            services.AddSingleton<ChimpanzeeScreen, ChimpanzeeScreen>();
        });
}
```