**Uniwersytet WSB Merito** 

## Programowanie Obiektowe. Zadanie 2

Uniwersytet WSB Merito

Imiona i nazwiśka: Mykhailo Matsko, Nazar Panasiuk

Adresy email: <a href="mailto:farbi333@gmail.com">farbi333@gmail.com</a>, deadfoxua1337@gmail.com

Nr. albumów: 144816, 143655

Data: 01.09.2023

Link na git:

Zgodnie z zadaniem utworzyliśmy klasy ScreenLineEntry, ScreenDefinition i ScreenDefinitionService:

```
public class ScreenLineEntry
{
    public ConsoleColor ForeGroundColor { get; set; }
    public ConsoleColor BackGroundColor { get; set; }
    public string? Text { get; set; }
}
```

```
99+ references
public static class ScreenDefinitionService

{
    8 references
    public static ScreenDefinition? screenDefinition { get; set; }
    3 references
    public static ScreenDefinition Load(string jsonFileName)...

99+ references
public static void DisplayLineFromFile(string jsonFileName, int lineNumber)...

0 references
public static void Serialize(string jsonFileName)...

7 references
public static List<ScreenLineEntry> ReadMenu(string jsonFileName)...

1 reference
public static bool IsMenuItem(string text)...

7 references
public static void DisplayMenu(List<ScreenLineEntry> menuEntries, int selectedOption)...
}
```

Następnie, zgodnie z zadaniem, dodaliśmy SampleHierachies.Services.Tests do projektu i dodaliśmy testy jednostkowe dla ScreenDefinitionService

```
public class ScreenDefinitionServiceTests
{
    private const string TestJsonFileName = "test.json";

    [TestInitialize]
    0 references
    public void Initialize()...

[TestMethod]
    0 references
    public void Load_ExistingFile_ReturnsScreenDefinition()...

[TestMethod]
    0 references
    public void Load_NonExistingFile_ThrowsException()...

[TestMethod]
    0 references
    public void DisplayLineFromFile_ValidLineNumber_DisplayText()...

[TestCleanup]
    0 references
    public void Cleanup()...
}
```

Dodaliśmy również publiczny wirtualny ciąg znaków screenDefinitionJson w klasie Screen, a następnie w każdej klasie, która dziedziczy po niej, przypisaliśmy własną wartość do screenDefinitionJson:

```
7 references

public abstract class Screen

{
    #region Public Methods

    /// <summary>
    /// Show the screen.
    /// </summary>
    14 references
    public virtual void Show()
    {
        Console.WriteLine("Showing screen");
    }

    99+ references
    public virtual string? screenDefinitionJson { get; set; }

#endregion // Public Methods

}
```

Dodano również: wyjście każdej linii z jej kolorami, ruch ze strzałkami,

historia wyboru użytkownika:

```
MainScreen -> AnimalScreen -> MammalScreen -> CatsScreen

0. Exit

1. List all cat

2. Create a new cat

3. Delete existing cat

4. Modify existing cat
```