

Uniwersytet WSB Merito

Programowanie obiektowe. Zadanie 2

[Document subtitle]

Imiona i nazwiska: Baranovskyi Ivan, Andrii Popchuk

Adresy email: ivan.baranovskyi@gmail.com

Numery albumu: [144401](#), [144561](#)

Data: 10.09.2023

Link do repo na Github:

Rozdział 1 – Zadania do zrealizowania

Zadania podstawowe:

1. Dodać klasę ScreenLineEntry o następujących polach:

- BackgroundColor
- ForegroundColor
- Text

Klasa ta będzie definiować pojedynczą linię na ekranie.

2. Dodać klasę o nazwie ScreenDefinition o następujących polach:
 - LineEntries - typu List<ScreenLineEntry>
3. Klasy z punktów 1 oraz rozszerzyć o definicje.
4. Dodać klasę statyczną o nazwie ScreenDefinionService , która zawiera następujące metody
 - ScreenDefinition Load(string jsonFileName)
 - bool Save(ScreenDefinition screenDefinition, string jsonFileName)

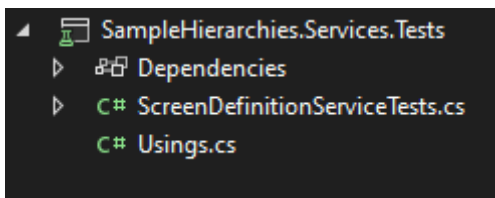
Referencja do tej klasy powinna być przekazana w taki sam sposób jak jest przekazywana referencja do SettingsService do klasy bazowej Screen.cs do konstruktora poprzez wstrzykiwanie zależności.

```
1 reference
public class ScreenLineEntry
{
    1 reference
    public ConsoleColor BackgroundColor { get; set; }
    1 reference
    public ConsoleColor ForegroundColor { get; set; }
    2 references
    public string? Text { get; set; }
}
```

```
public class ScreenDefinition
{
    public List<ScreenLineEntry>? LineEntries { get; set; }
}
```

```
99+ references
public static class ScreenDefinitionService
{
    4 references
    public static ScreenDefinition Load(string jsonFileName)...
    99+ references
    public static void ShowLine(string jsonFileName, int numberLineEntry)...
}
```

5. Dodać nowy projekt o nazwie SampleHierarchies.Services.Tests, do którego należy dodać unit testy klasy ScreenDefinitionService.cs
6. Dodać do klasy Screen.cs pole o nazwie ScreenDefinitionJson, które będzie przechowywało nazwę pliku z definicją ekranu.
7. Nadpisać tą wartość w każdej klasie potomnej.
8. Dodać pliki JSON dla każdego z ekranów z definicją ich zawartości.
9. Zastąpić obecnie istniejące definicje ekranów z wpisanym na zasadzie 'hardcoded' informacją pochodzącą z pliku JSON.



```
[TestClass]
0 references
public class ScreenDefinitionServiceTests
{
    private const string ValidJsonFileName = "valid.json";
    private const string InvalidJsonFileName = "invalid.json";
    private const string NonExistentJsonFileName = "nonexistent.json";

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

    [TestMethod]
    [ExpectedException(typeof(Exception), "Bad read from file")]
    0 references
    public void Load_InvalidJsonFile_ThrowsException()...

    [TestMethod]
    [ExpectedException(typeof(Exception), "File is not found")]
    0 references
    public void Load_NonExistentJsonFile_ThrowsException()...

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

    [TestMethod]
    [ExpectedException(typeof(Exception), "File is not found")]
    0 references
    public void ShowLine_NonExistentJsonFile_ThrowsException()...

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

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

    #region Public Methods

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

    #endregion // Public Methods
}
```

10. Wstawić renderowanie ekranu na podstawie kolorystyki wskazanej w plik JSON.
11. Dodać możliwość wyświetlenia historii wyborów poprzez zapamiętanie jej i wyświetlenie na górze ekranu. Czyli np Main Screen -> Mammals -> Dog

16 references

```
public override string screenDefinitionJson { get; set; } = "AnimalsScreenLineEntries.json";
```

```
MainScreen -> AnimalScreen -> MammalScreen -> KangaroosScreen
```

Your available choices are:

0. Exit

1. List all kangaroos

2. Create a new kangaroo

3. Delete existing kangaroo

4. Modify existing kangaroo

Please enter your choice: