Uniwersytet WSB Merito

Programowanie Obiektowe. Zadanie 1

Uniwersytet WSB Merito

Imiona i nazwiśka: Mykhailo Matsko, Nazar Panasiuk

Adresy email: farbi333@gmail.com, deadfoxua1337@gmail.com

Nr. albumów: 144816, 143655

Data: 01.09.2023

Link na git: https://github.com/AcidWSB/SampleHierarchiesApp_Zadanie1-2

Aby wykonać pierwsze zadanie, zmieniliśmy następujące interfejsy: ISettings, IsettingsService:

```
1 reference
public interface ISettings
{
    #region Interface Members

    /// <summary>
    /// Version of settings.
    /// </summary>
    2 references
    string Version { get; set; }

    8 references
    Dictionary<string, ConsoleColor> ScreenColors { get; set; }

    #endregion // Interface Members
}
```

```
1 reference
public interface ISettingsService
{
    #region Interface Members

    /// <summary>
    /// Read settings.
    /// </summary>
    /// <param name="jsonPath">Json path</param>
    /// <returns></returns>
    /// <summary>
    /// Write settings.
    /// </summary>
    /// <param name="settings">Settings to written</param>
    /// <param name="jsonPath">Json path</param>

    8 references
    void SetColor(string screenName);
    2 references
    void SerializeSettingsFile();
    2 references
    void EditSettingsFile();

    #endregion // Interface Members
}
```

Następnie zmieniliśmy i zaimplementowaliśmy te interfejsy w klasie SettingsService:

```
public class SettingsService : ISettingsService, ISettings
{
    #region

    /// <inheritdoc/>
    8 references
    public Dictionary<string, ConsoleColor> ScreenColors { get; set; }
    8 references
    public string JsonPath { get; set; }
    2 references
    public string Version { get; set; }

    0 references
    public SettingsService()...
    2 references
    public void SerializeSettingsFile()...

    /// <summary>
    /// Deserialize menu color settings from a JSON file.
    /// </summary>
    /// <returns></returns>
    /// <exception cref="FileNotFoundException"></exception>
    /// <exception cref="Exception"></exception>
    2 references
    private Dictionary<string, ConsoleColor> DeserializeSettingsFile()...
    /// <summary>
    /// Method, that SetColor for Screen which user want
    /// </summary>
    /// <param name="screenName"></param>
    8 references
    public void SetColor(string screenName)...
    2 references
    public void EditSettingsFile()...
    #endregion
}
```

Aby ustawić kolor ekranu, stworzyliśmy metodę SetColor:

```
public void SetColor(string screenName)
{
    Dictionary<string, ConsoleColor> colorsOfMenu = DeserializeSettingsFile();
    if (colorsOfMenu.ContainsKey(screenName))
    {
        ConsoleColor color = colorsOfMenu[screenName];
        Console.ForegroundColor = color;
    }
    else
    {
        Console.WriteLine($"Color for  '{screenName}' menu wasn't found.");
    }
}
```

Metody pomocnicze:

```
private Dictionary<string, ConsoleColor> DeserializeSettingsFile()
{
    try
    {
        string json = File.ReadAllText(JsonPath);
        var deserializedMenuColors = JsonConvert.DeserializeObject<Dictionary<string, string>>(json);
        if (deserializedMenuColors != null)
        {
            var menuColors = deserializedMenuColors.ToDictionary(kv => kv.Key, kv => (ConsoleColor)Enum.Parse(typeof(ConsoleColor), kv.Value));
            return menuColors;
        }
        else
        {
            throw new Exception("JSON deserialization resulted in null.");
        }
    }
    catch (FileNotFoundException)
    {
        Console.WriteLine($"File {JsonPath} not found.");
        throw new FileNotFoundException($"File {JsonPath} not found.");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error occurred during deserialization: {ex.Message}");
        throw new Exception($"Error occurred during deserialization: {ex.Message}", ex);
    }
}
```

```
public void SerializeSettingsFile()
{
    try
    {
        if (File.Exists(JsonPath))
        {
            string existingJson = File.ReadAllText(JsonPath);

            if (!string.IsNullOrWhiteSpace(existingJson))
            {
                var existingMenuColors = JsonConvert.DeserializeObject<Dictionary<string, string>>(existingJson);

                if (existingMenuColors != null)
                {
                    foreach (var keyValuePair in ScreenColors)
                    {
                        existingMenuColors[keyValuePair.Key] = keyValuePair.Value.ToString();
                    }
                    string updatedJson = JsonConvert.SerializeObject(existingMenuColors, Formatting.Indented);


                    File.WriteAllText(JsonPath, updatedJson);
                    Console.WriteLine("Settings successfully updated in 'ScreenColors.json'.");
                }
            }
        }
        else
        {
            var serializedMenuColors = ScreenColors.ToDictionary(kv => kv.Key, kv => kv.Value.ToString());
            string json = JsonConvert.SerializeObject(serializedMenuColors, Formatting.Indented);


            File.WriteAllText(JsonPath, json);
            Console.WriteLine("Settings successfully serialized to 'ScreenColors.json'.");
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error occurred during serialization: {ex.Message}");
    }
}
```

Metoda, która pomaga użytkownikowi zmienić kolor wybranego ekranu:

```
public void EditSettingsFile()
{
    ScreenColors = DeserializeSettingsFile();
    Console.WriteLine("Enter the name of the screen you want to change the color for: (MainScreen, AnimalScreen etc.)");
    string? screenName = Console.ReadLine();

    if (screenName != null && ScreenColors.ContainsKey(screenName))
    {
        // Display the current color for the selected screen
        Console.WriteLine($"Current color for screen '{screenName}': {ScreenColors[screenName]}");
        Console.WriteLine("Enter a new color (Red, Green, Yellow, etc.):");
        string? newColorStr = Console.ReadLine();

        // Attempt to parse the entered color into a ConsoleColor enum
        if (Enum.TryParse(typeof(ConsoleColor), newColorStr, out object? newColorObj) && newColorObj is ConsoleColor newColor)
        {
            // Set the new color for the screen
            ScreenColors[screenName] = newColor;
            Console.WriteLine($"Color for screen '{screenName}' successfully changed to {newColor}.");

            // Serialize the updated settings back to the file
            SerializeSettingsFile();
        }
        else
        {
            Console.WriteLine("Error: Invalid color entered.");
        }
    }
    else
    {
        Console.WriteLine($"Error: Screen with the name '{screenName}' not found in the settings file.");
    }
}
#endregion
```