

Uniwersytet WSB Merito

Programowanie Obiektowe

Ćwiczenia 1 Sprawozdanie

Imiona i Nazwiska: Markin Vadym, Kachalov Yevhenii

Adresy mail: kachalovz171@gmail.com, markin4097@gmail.com

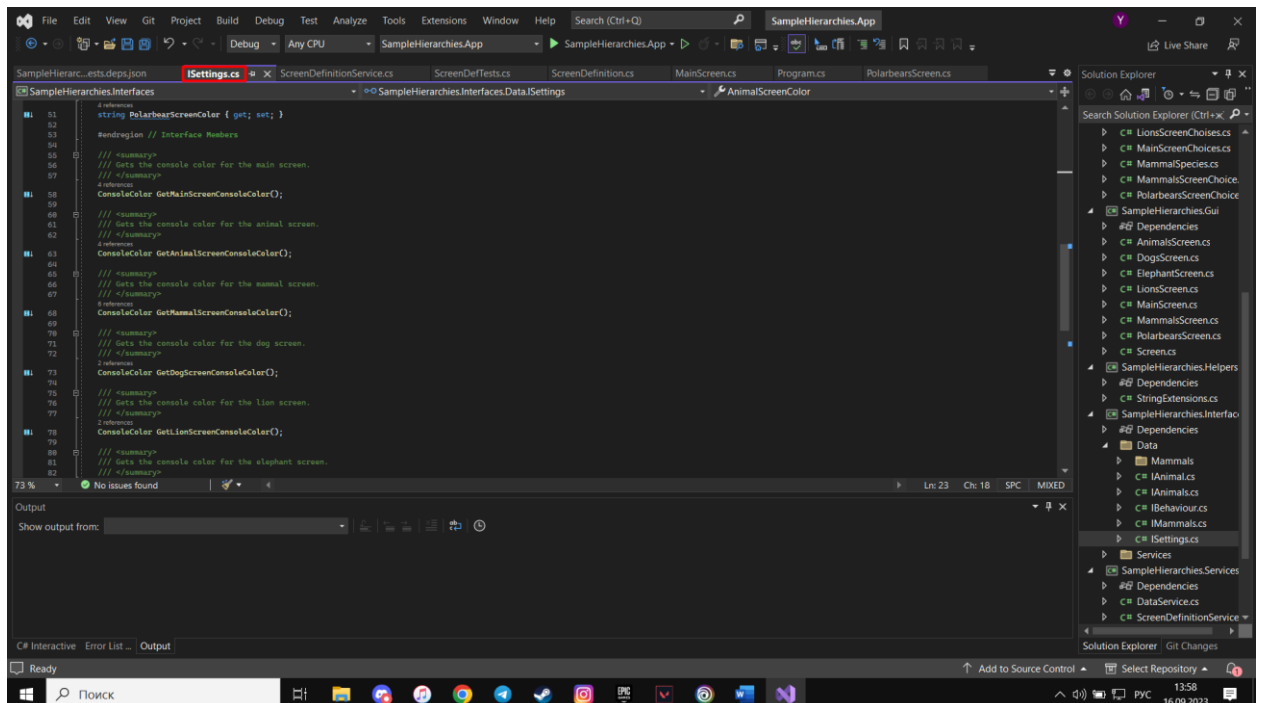
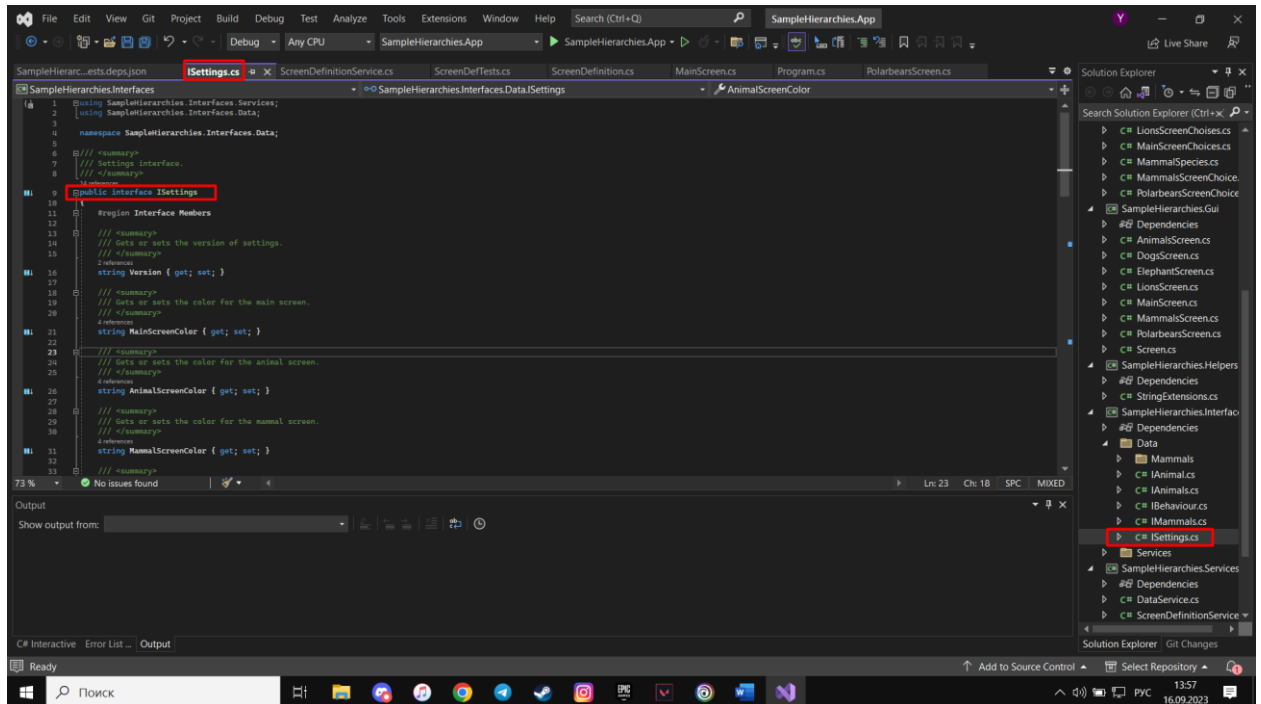
Nr.Albumów: 143069, 143071

Link na git: <https://github.com/BlupiR>

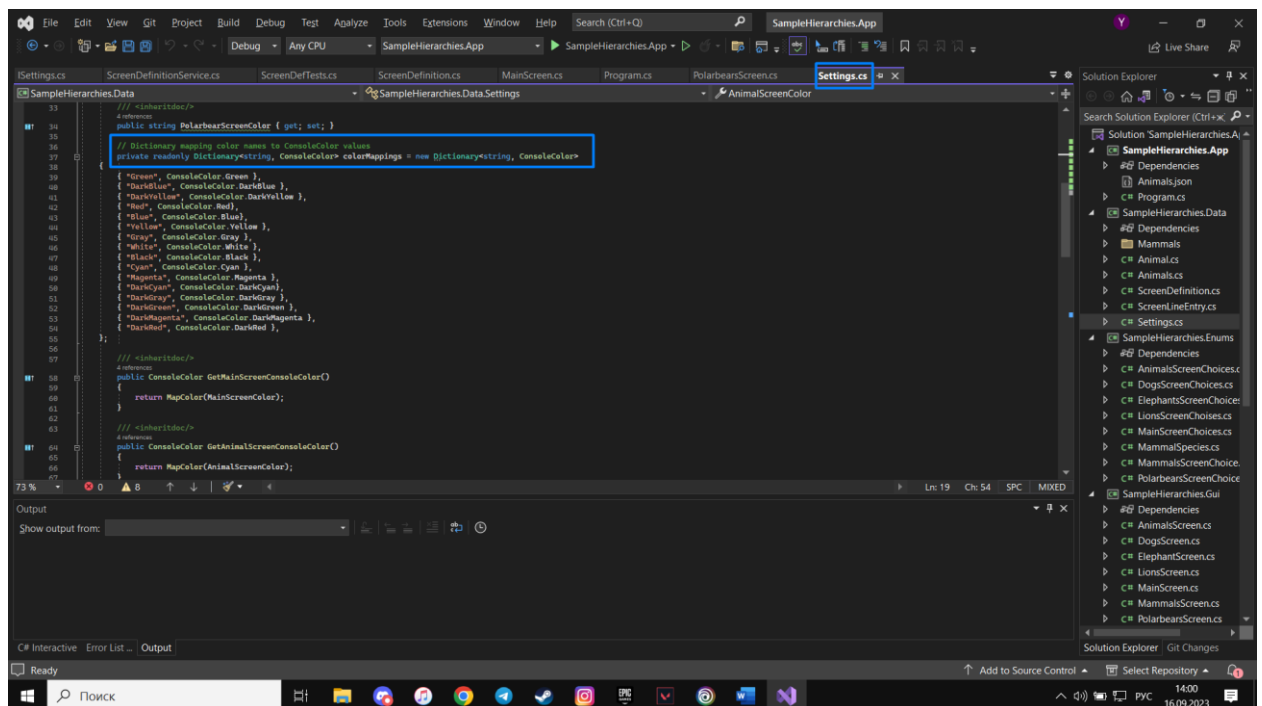
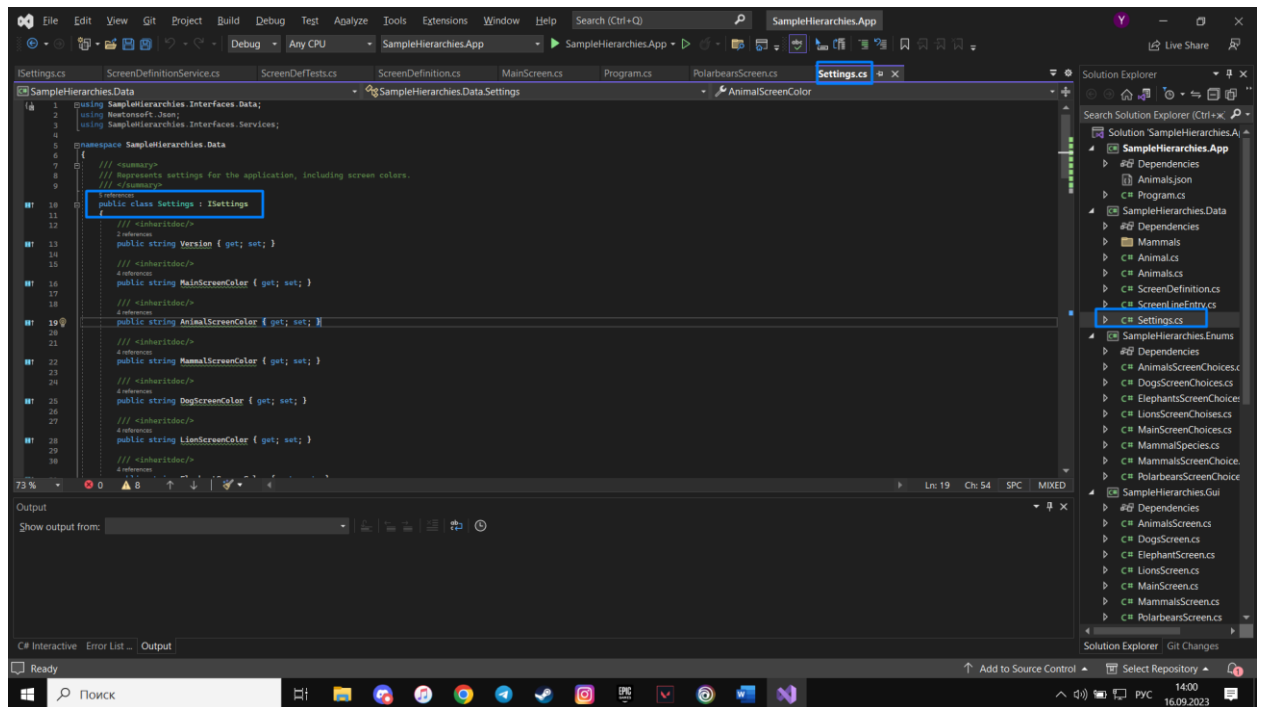
1) Dodanie ustawień dotyczących kolorystyki poszczególnych ekranów:

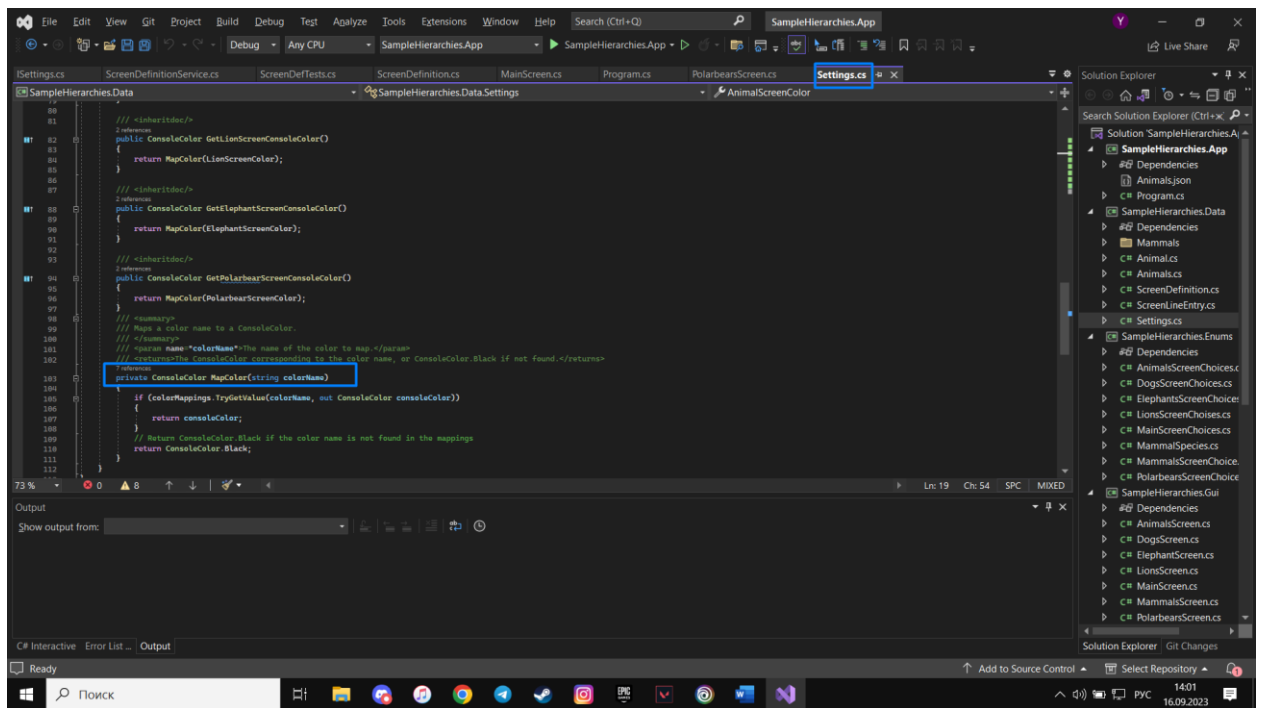
a) Zmodyfikowaliśmy interfejs **ISettings** oraz klasę **Settings**, dodaliśmy pola typu string opisujące kolory poszczególnych ekranów i opisaliśmy następujące ekrany: ekran główny, ekran zwierząt, ekran ssaków i ekran psa, a także ekran ekrany zwierząt, które sami stworzyliśmy. ✓

Interfejs ISettings:



Klass Settings:





b) Dodano możliwość zapisywania i odczytywania ustawień do plików.
Zmodyfikowano interfejs `ISettingsService` oraz `SettingsService`

✓ `ISettingsService`:

```

1  using SampleHierarchies.Interfaces.Data;
2
3  namespace SampleHierarchies.Interfaces.Services;
4
5  12 references
6  public interface ISettingsService
7  {
8      #region Interface Members
9
10     /// <summary>
11     /// Save settings.
12     /// </summary>
13     /// <param name="filePath">Json path</param>
14
15     2 references
16     void SaveSettings(ISettings settings, string filePath);
17     /// <summary>
18     /// Load settings.
19     /// </summary>
20     /// <param name="filePath">Json path</param>
21
22     8 references
23     ISettings LoadSettings(string filePath);
24     /// <summary>
25     /// Change Color of Screen
26     /// </summary>
27     /// <param name="settings">Settings of Colors</param>
28     /// <param name="screenName">Name of Screen</param>
29     2 references
30     void ChangeScreenColor(ISettings settings, string screenName);
31     #endregion // Interface Members
32 }

```

SettingsService:

```
ISettings.cs* Settings.cs SettingsService.cs ISettingsService.cs*
SampleHierarchies.Services SampleHierarchies.Services.SettingsService
1 using SampleHierarchies.Data;
2 using SampleHierarchies.Interfaces.Data;
3 using SampleHierarchies.Interfaces.Services;
4 using Newtonsoft.Json;
5 using System.Diagnostics;
6 using System.Linq.Expressions;
7 namespace SampleHierarchies.Services;
8
9 /// <summary>
10 /// Settings service.
11 /// </summary>
12
13 public class SettingsService : ISettingsService
14 {
15     #region ISettings Implementation
16
17     public ISettings LoadSettings(string fileName)
18     {
19         if (File.Exists(fileName))
20         {
21             var json = File.ReadAllText(fileName);
22             return JsonConvert.DeserializeObject<Settings>(json);
23         }
24         else
25         {
26             // Jeśli plik nie istnieje, możesz zwrócić nowy obiekt ustawień z domyślnymi wartościami lub obsłużyć ten przypadek w inny sposób.
27             return new Settings();
28         }
29     }
30
31     public void SaveSettings(ISettings settings, string fileName)
32     {
33         string json = JsonConvert.SerializeObject(settings);
34         File.WriteAllText(fileName, json);
35     }
36
37     public void ChangeScreenColor(ISettings settings, string screenName)
38     {
39     }
40 }
```

```

37
38     Console.WriteLine($"Enter the new screen color for {screenName}:");
39     string newColor = Console.ReadLine();
40     Console.WriteLine($"Enter the new text color for {screenName}:");
41     string newColor2 = Console.ReadLine();
42
43     // В зависимости от имени экрана, установите соответствующее поле цвета в настройках
44     switch (screenName)
45     {
46         case "Main Screen":
47             settings.MainScreenColor = newColor;
48             settings.TextMainScreenColor = newColor2;
49             Console.BackgroundColor = settings.GetMainScreenConsoleColor();
50             Console.ForegroundColor = settings.GetTextMainScreenColor();
51             break;
52         case "Animals Screen":
53             if (screenName is not null)
54                 settings.AnimalScreenColor = newColor;
55                 settings.TextAnimalScreenColor = newColor2;
56             break;
57         case "Mammals Screen":
58             settings.MammalScreenColor = newColor;
59             settings.TextMammalScreenColor = newColor2;
60             break;
61         case "Dogs Screen":
62             settings.DogScreenColor = newColor;
63             settings.TextDogScreenColor = newColor2;
64             break;
65         case "Lions Screen":
66             settings.LionScreenColor = newColor;
67             settings.TextLionScreenColor = newColor2;
68             break;
69         case "Elephants Screen":
70             settings.ElephantScreenColor = newColor;
71             settings.TextElephantScreenColor = newColor2;
72             break;
73         case "Polarbears Screen":
74             settings.PolarbearScreenColor = newColor;
75             settings.TextPolarbearScreenColor = newColor2;
76             break;
77
78         default:
79             Console.WriteLine($"Screen '{screenName}' not found in settings.");
80             return;
81     }
82
83     // Сохраните обновленные настройки
84     ISettingsService settingsService = new SettingsService();
85     string FilePath = "settings.json";
86     settingsService.SaveSettings(settings, FilePath);
87
88     Console.WriteLine($"Screen for '{screenName}' has been updated to: {newColor}");
89     Console.WriteLine($"Text color for '{screenName}' has been updated to: {newColor2}");
90 }
91
92 }
93
94
95
96
97
98 #endregion // ISettings Implementation
99

```

c) Dodano wyświetlanie ekranów w kolorze określonym w ustawieniach ✓

```
var defaultSettings = new Settings
{
    Version = "1.0",
    MainScreenColor = "Yellow",
    AnimalScreenColor = "Cyan",
    MammalScreenColor = "Green",
    DogScreenColor = "Magenta",
    LionScreenColor = "Red",
    ElephantScreenColor = "Yellow",
    PolarbearScreenColor = "DarkGreen",
    TextMainScreenColor = "Black",
    TextAnimalScreenColor = "Red",
    TextMammalScreenColor = "Magenta",
    TextDogScreenColor = "Green",
    TextLionScreenColor = "Cyan",
    TextElephantScreenColor = "Blue",
    TextPolarbearScreenColor = "DarkMagenta"
};
```

JSON-file with default settings: settings.json

```
Your available choices are:
0. Exit
1. Animals
2. Update color settings
Please enter your choice: 1
```

```
Your available choices are:
0. Exit
1. Mammals
2. Save to file
3. Read from file
Please enter your choice: _
```

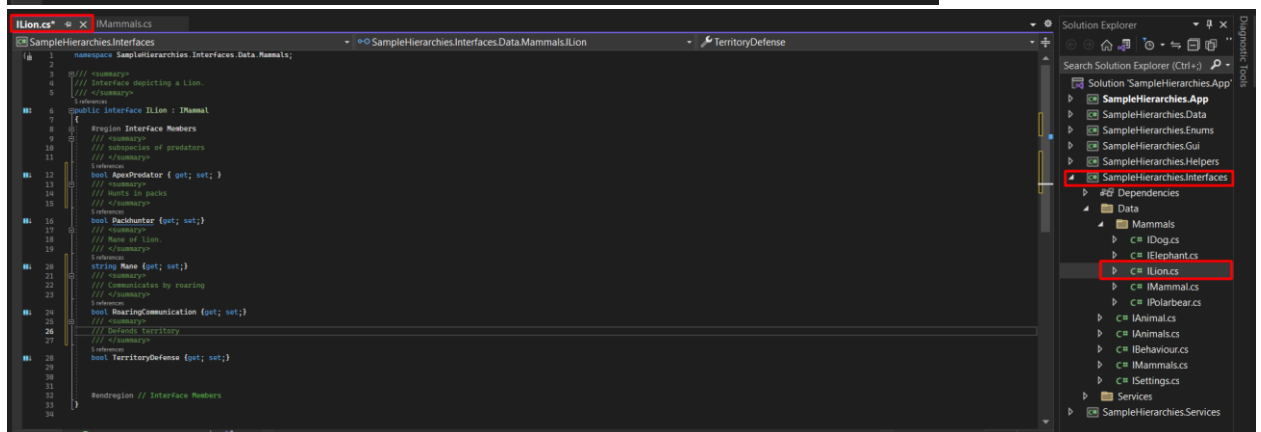
2) Dodaliśmy struktury danych, interfejsy, zmodyfikowaliśmy odpowiednią serwisu oraz dodaliśmy nowy ekran opisujący jeden z poniższych typów zwierząt (Lew). ✓

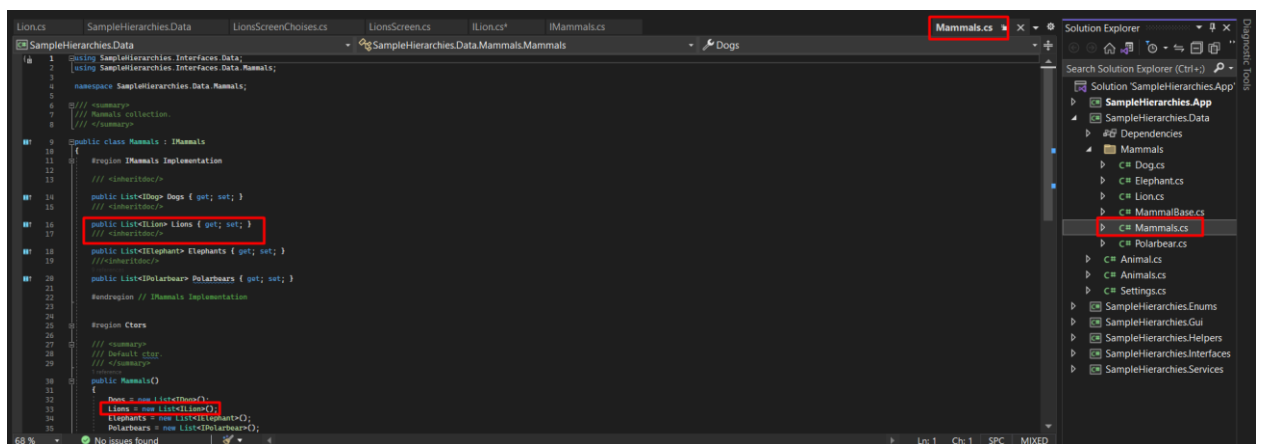
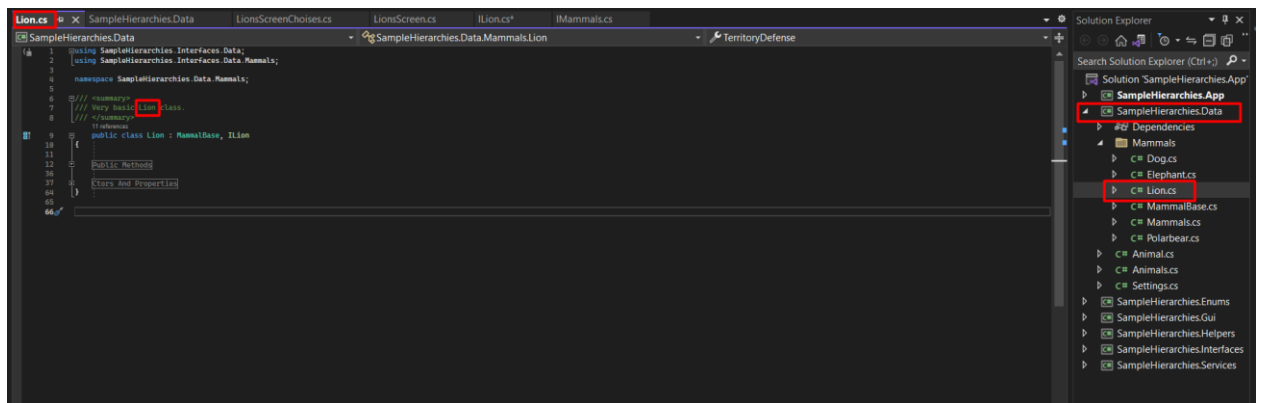
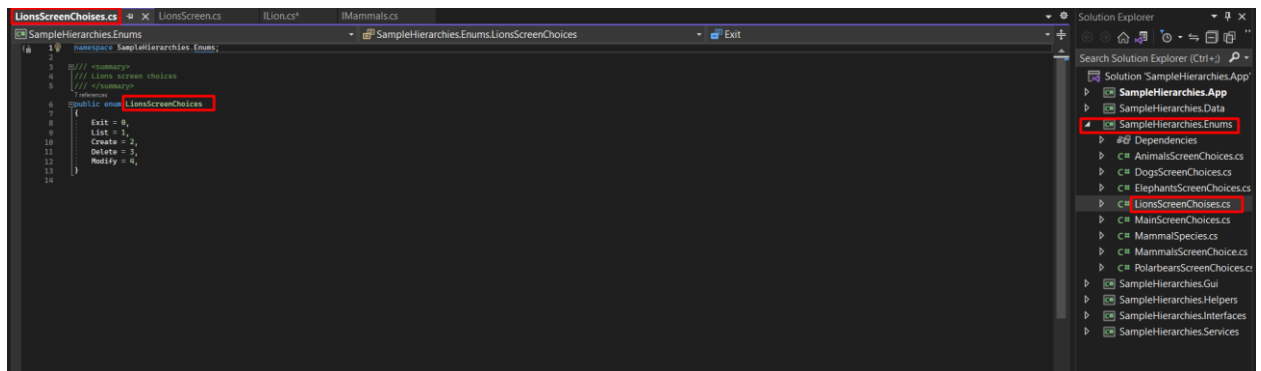
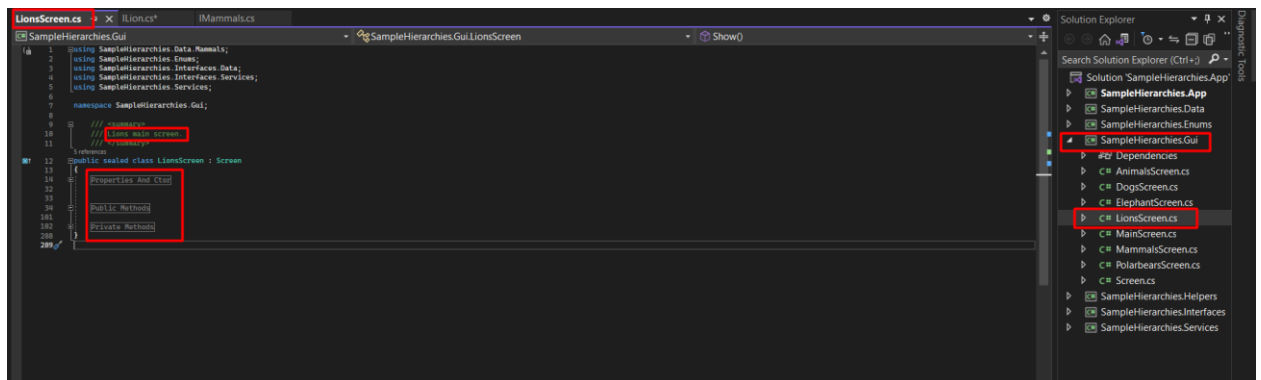
Wszystkie określone obiekty o określonych właściwościach zostały zapisane do pliku. W rezultacie po zapisaniu danych udostępniono plik JSON. ✓

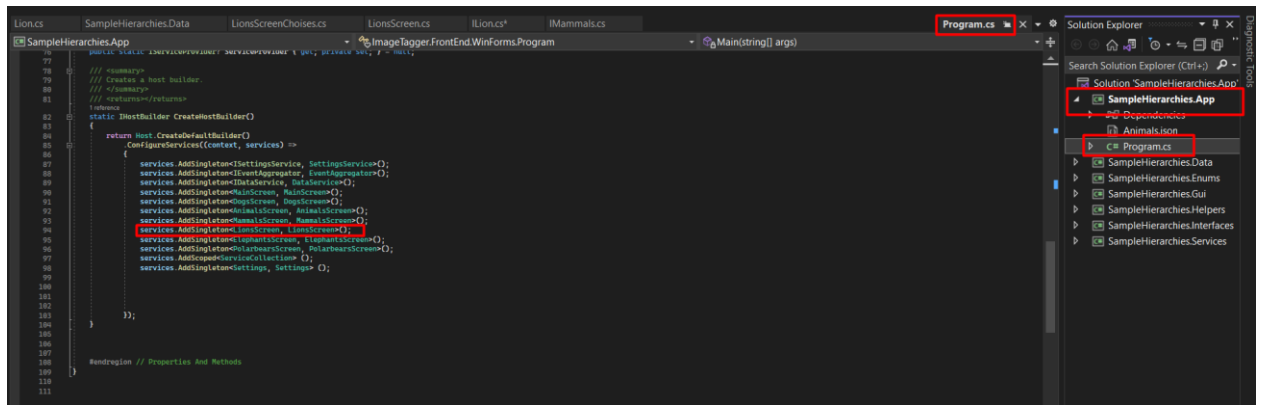
```
1 {
2   "Mammals": [
3     {
4       "Name": "Lion",
5       "Age": 10,
6       "Height": 1.2,
7       "Weight": 150,
8       "Species": "Lion",
9       "Age": 10,
10      "Height": 1.2,
11      "Weight": 150,
12      "Species": "Lion",
13      "Age": 10,
14      "Height": 1.2,
15      "Weight": 150,
16      "Species": "Lion",
17      "Age": 10,
18      "Height": 1.2,
19      "Weight": 150,
20      "Species": "Lion",
21      "Age": 10,
22      "Height": 1.2,
23      "Weight": 150,
24      "Species": "Lion",
25      "Age": 10,
26      "Height": 1.2,
27      "Weight": 150,
28      "Species": "Lion",
29      "Age": 10,
30      "Height": 1.2,
31      "Weight": 150,
32      "Species": "Lion",
33      "Age": 10,
34      "Height": 1.2,
35      "Weight": 150,
36      "Species": "Lion",
37      "Age": 10,
38      "Height": 1.2,
39      "Weight": 150,
40      "Species": "Lion",
41      "Age": 10,
42      "Height": 1.2,
43      "Weight": 150,
44      "Species": "Lion",
45      "Age": 10,
46      "Height": 1.2,
47      "Weight": 150,
48      "Species": "Lion",
49      "Age": 10,
50      "Height": 1.2,
51      "Weight": 150,
52      "Species": "Lion",
53      "Age": 10,
54      "Height": 1.2,
55      "Weight": 150,
56      "Species": "Lion",
57      "Age": 10,
58      "Height": 1.2,
59      "Weight": 150,
60      "Species": "Lion",
61      "Age": 10,
62      "Height": 1.2,
63      "Weight": 150,
64      "Species": "Lion",
65      "Age": 10,
66      "Height": 1.2,
67      "Weight": 150,
68      "Species": "Lion",
69      "Age": 10,
70      "Height": 1.2,
71      "Weight": 150,
72      "Species": "Lion",
73      "Age": 10,
74      "Height": 1.2,
75      "Weight": 150,
76      "Species": "Lion",
77      "Age": 10,
78      "Height": 1.2,
79      "Weight": 150,
80      "Species": "Lion",
81      "Age": 10,
82      "Height": 1.2,
83      "Weight": 150,
84      "Species": "Lion",
85      "Age": 10,
86      "Height": 1.2,
87      "Weight": 150,
88      "Species": "Lion",
89      "Age": 10,
90      "Height": 1.2,
91      "Weight": 150,
92      "Species": "Lion",
93      "Age": 10,
94      "Height": 1.2,
95      "Weight": 150,
96      "Species": "Lion",
97      "Age": 10,
98      "Height": 1.2,
99      "Weight": 150,
100     }
101   ]
102 }
```



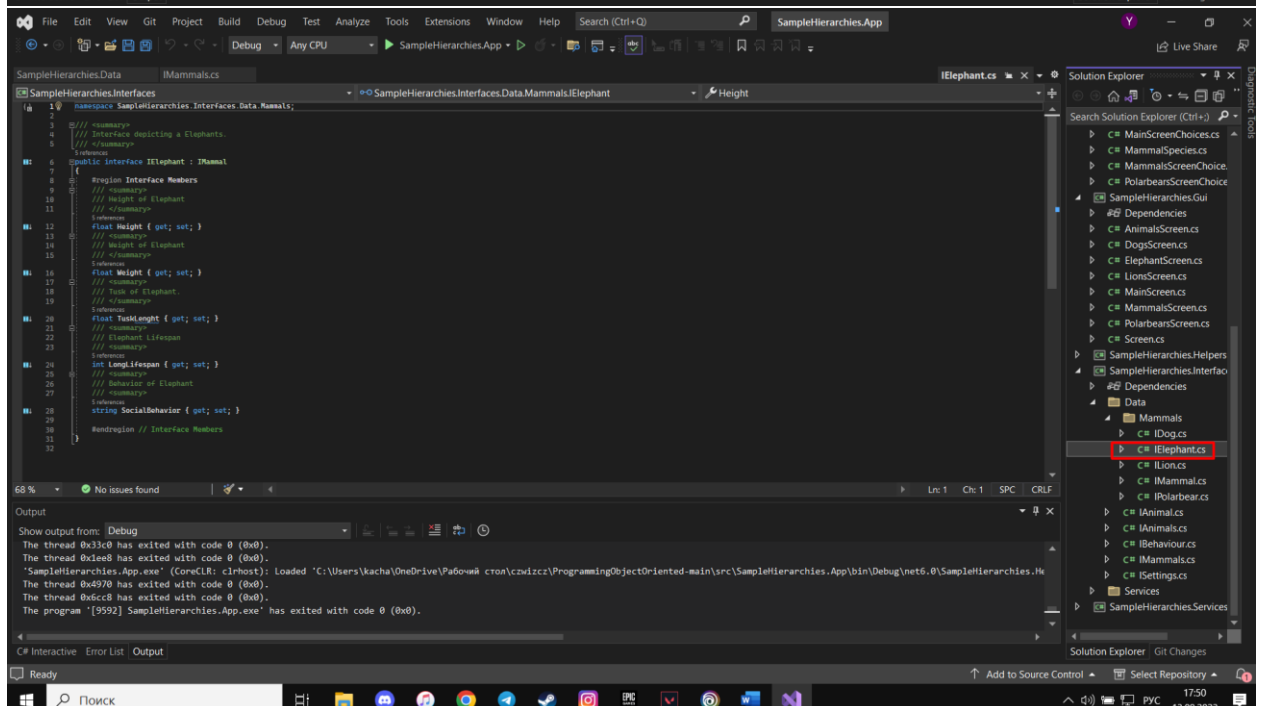
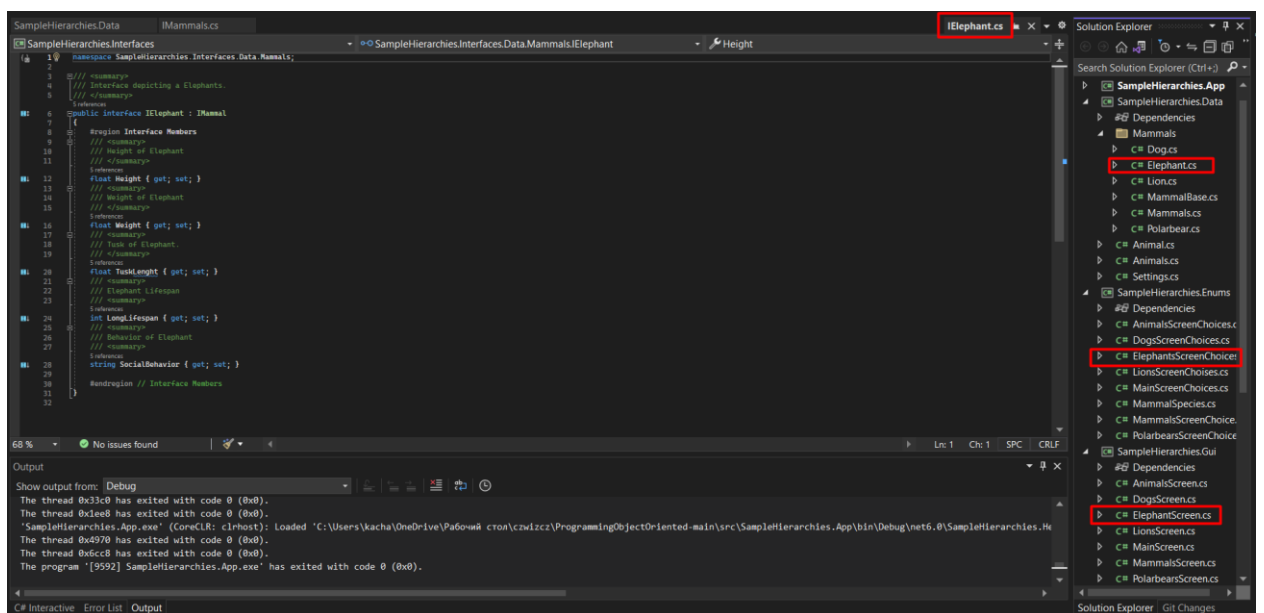
```
ILion.cs* | Mammals.cs | X
C# SampleHierarchies.Interfaces
1 using SampleHierarchies.Interfaces.Data.Mammals;
2
3 namespace SampleHierarchies.Interfaces.Data;
4
5 /// <summary>
6 /// Mammals collection.
7 /// </summary>
8 public interface IMammals
9 {
10     #region Interface Members
11
12     /// <summary>
13     /// Dogs collection.
14     /// </summary>
15     List<IDog> Dogs { get; set; }
16
17     /// <summary>
18     /// Lions collection.
19     List<ILion> Lions { get; set; }
20
21     /// <summary>
22     /// Elephants collection.
23     List<IElephant> Elephants { get; set; }
24
25     /// <summary>
26     /// Polarbears collection
27     List<IPolarbear> Polarbears { get; set; }
28
29     #endregion // Interface Members
30 }
31
```







3) Dodać kolejny rodzaj ssaka inny od punktu 2. ✓



4) Dodać kolejny rodzaj ssaka inny od punktów 2 oraz 3.

