

Uniwersytet WSB Merito

# Programowanie obiektowe.Zadanie 1

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: [https://github.com/ivan-baranovskyi/Programowanie\\_obiektowe-1.git](https://github.com/ivan-baranovskyi/Programowanie_obiektowe-1.git)

## Rozdział 1 – Zadanie 1

Dodanie ustawień dotyczących kolorystyki poszczególnych ekranów:

- a) Należy zmodyfikować interfejs `ISettings` oraz klasę `Settings` i dodać pola typu `string` opisujących kolorystykę poszczególnych ekranów. Ekran, które powinny być opisane to ekran główny, ekran zwierząt, ekran ssaków, ekran psów. Kolory mogą się powtarzać.
- b) Dodać opcję zapisu i odczytu ustawień do plików. Zmodyfikować interfejs `ISettingsService` oraz `SettingsService`.
- c) Dodać wyświetlanie ekranów w kolorze określonym w ustawieniach.

```
public interface ISettings
{
    #region Interface Members

    /// <summary>
    /// Version of settings.
    /// </summary>
    1 reference
    string? Version { get; set; }

    1 reference
    ConsoleColor screenColor { get; set; }

    #endregion // Interface Members
}
```

```

public class Settings : ISettings
{
    public string? Version { get; set; }

    public ConsoleColor screenColor { get; set; }
}

```

```

public interface ISettingsService
{
    #region Interface Members

    /// <summary>
    /// Read settings.
    /// </summary>
    /// <param name="jsonPath">Json path</param>
    /// <returns></returns>
    1 reference
    ISettings? Read(string jsonPath);

    /// <summary>
    /// Write settings.
    /// </summary>
    /// <param name="settings">Settings to written</param>
    /// <param name="jsonPath">Json path</param>
    1 reference
    void Write(ISettings settings, string jsonPath);

    /// <summary>
    /// Get screen color from json file.
    /// </summary>
    /// <param name="screenName"></param>
    /// <returns></returns>
    2 references
    ConsoleColor GetScreenColorFromJson(string screenName);

    /// <summary>
    /// Set color for screen
    /// </summary>
    8 references
    public void SetScreenColor(string screenName);

    #endregion // Interface Members
}

```

```

17 references
public class SettingsService : ISettingsService
{
    5 references
    private string FilePath { get; set; } = "screenColorSettings.json";

    #region ISettingsService Implementation
    /// <inheritdoc/>
    1 reference
    public ISettings? Read(string jsonPath)...

    /// <inheritdoc/>
    1 reference
    public void Write(ISettings settings, string jsonPath)...

    /// <summary>
    /// Edit settings (colors) of screen
    /// </summary>
    1 reference
    public void EditSettings()...

    /// <summary>
    /// Get screen color from screenColorSettings.json
    /// </summary>
    /// <param name="screenName"></param>
    /// <returns></returns>
    2 references
    public ConsoleColor GetScreenColorFromJson(string screenName)...

    /// <summary>
    /// Set color of screen
    /// </summary>
    /// <param name="screenName"></param>
    8 references
    public void SetScreenColor(string screenName)...

    #endregion // ISettings Implementation
}

```

Main Screen

```

Your available choices are:
0. Exit
1. Animals
2. Edit settings of menus
Please enter your choice:

```

MainScreen -> AnimalScreen

```

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

```

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

```
Your available choices are:
```

- 0. Exit
- 1. Dogs
- 2. Kangaroos
- 3. Sloths
- 4. Fennecs

```
Please enter your choice:
```

```
MainScreen -> AnimalScreen -> MammalScreen -> DogsScreen
```

```
Your available choices are:
```

- 0. Exit
- 1. List all dogs
- 2. Create a new dog
- 3. Delete existing dog
- 4. Modify existing dog

```
Please enter your choice:
```

1. Dodać struktur danych, interfejsy, zmodyfikować odpowiednie serwisu oraz dodać nowy ekran opisujący jeden z poniższych rodzajów zwierząt. Wybrać jeden z gatunków. Zapisać do pliku wszystkie wskazane jednostki ze wskazanymi właściwościami. Podać w wyniku plik JSON po zapisie danych.
2. Dodać kolejny rodzaj ssaka inny od punktu 2.

Dodać kolejny rodzaj ssaka inny od punktów 2 oraz 3.

```
└─ Mammals
   ├── C# Dog.cs
   ├── C# Fennec.cs
   ├── C# Kangaroo.cs
   ├── C# MammalBase.cs
   ├── C# Mammals.cs
   └── C# Sloth.cs
```

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

```
Your available choices are:
```

- 0. Exit
- 1. Dogs
- 2. Kangaroos
- 3. Sloths
- 4. Fennecs

```
Please enter your choice:
```

Toolbox

NewAnimals.json X SampleHierarch....Services.Tests Sc

Schema: <No Schema Selected>

```
1  {
2    "Mammals": {
3      "Dogs": [
4      ],
5    },
6    "Kangaroos": [
7      {
8        "Habitat": "Savanna",
9        "FamilyGroup": "Alone",
10       "JumpHeight": 1.2,
11       "Lifespan": 10,
12       "Species": 2,
13       "Name": "Jerry",
14       "Age": 6
15     }
16   ],
17   "Sloths": [
18     {
19       "FurColor": "White",
20       "Diet": "Vegatables",
21       "ClimbingSpeed": 2.0,
22       "DailyActivityHours": 5,
23       "SleepingTime": 19,
24       "Species": 3,
25       "Name": "Carry",
26       "Age": 6
27     }
28   ],
29   "Fennecs": [
30     {
31       "EarSize": 1.1,
32       "CoatPattern": "brown",
33       "TailLength": 1.7,
34       "DietaryHabits": "sad",
35       "RunningSpeed": 12.2,
36       "Species": 4,
37       "Name": "Clark",
38       "Age": 12
39     }
40   ]
41 }
42 }
```