Wyższa Szkoła Bankowa

Programowanie Obiektowe

Ćwiczenia 1 - zadania

Imiona I nazwiska: Artem Pushkarov, Maksym Cherniakov
Adresy mail: furyX66@gmail.com, mchernyako2003@gmail.com
Id's: 144819, 144806
Link na git:
https://github.com/furyX66/ProgramowanieObiektowe_Zadanie1_2023

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. Ekrany, 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.

Do interfejsa ISettings dodano właściwość ConsolColor typu string i metodę ReadFromJson.

```
public interface ISettings
{
    #region Interface Members

    /// <summary>
    /// Version of settings.
    /// </summary>
    1 reference
    public string Version { get; set; }
    1 reference
    public string ConsoleColor { get; set; }
    8 references
    public void ReadFromJson(string menu);
    #endregion // Interface Members
}
```

To pokazuje implementację interfejca Isettings.

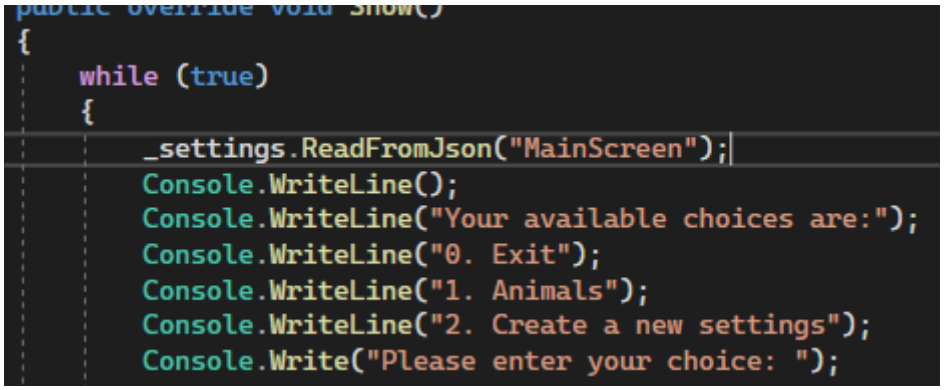
```
public class Settings : ISettings
{
    1 reference
    public string? Version { get; set; }
    1 reference
    public string? ConsoleColor { get; set; }
    3 references
    public string? FileName { get; set; }
    //ctor
    0 references
    public Settings()
    {
        FileName = "ColorSettings.json";
    }
    //Deserialize Method
    8 references
    public void ReadFromJson(string menu)//Read settings from json file and changes console color
    {
        if (File.Exists(FileName))
        {
            string content = File.ReadAllText(FileName);
            dynamic deserializedObj = JsonConvert.DeserializeObject(content);
            if (deserializedObj[menu] != null)
            {
                Console.ForegroundColor = deserializedObj[menu];
            }
        }
    }

}
```

Tak to wygląda ColorSettings.json.

To określa kolory dla wszystkich ekranów w mojej aplikacji

Przykład wywołania metody ReadFromJson



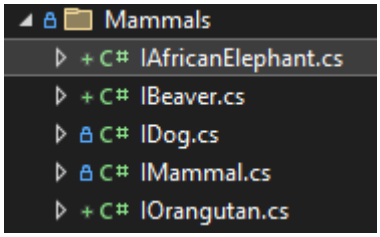
Rozdział 2 – Zadanie 2-4

2. 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.

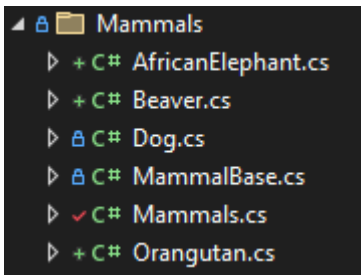
3. Dodać kolejny rodzaj ssaka inny od punktu 2.

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

Dodano interfejsy dla zwierząt



Dodano klasy implementujące te interfejsy



Przykład interfejsu i klasy, która go implementuje

```
public interface IAfricanElephant : IMammal
{
    #region Interface Members
    /// <summary>
    /// Height of elephant
    /// Weight of elephant
    /// TuskLength of elephant
    /// LongLifespan of elephant
    /// Social Behavior of elephant
    /// </summary>
    5 references
    float Height { get; set; }
    5 references
    float Weight { get; set; }
    5 references
    float TuskLength { get; set; }
    5 references
    int LongLifespan { get; set; }
    5 references
    string SocialBehavior { get; set; }

    #endregion // Interface Members
}
```

```
public class AfricanElephant : MammalBase, IAfricanElephant
{
    #region Public Methods

    /// <inheritdoc/>
    4 references
    public override void MakeSound()
    {
        Console.WriteLine("My name is: {0} and I am barking", Name);
    }

    /// <inheritdoc/>
    4 references
    public override void Move()
    {
        Console.WriteLine("My name is: {0} and I am running", Name);
    }

    /// <inheritdoc/>
    4 references
    public override void Display()
    {
        Console.WriteLine($"My name is: {Name}, my age is: {Age} and
    }

    /// <inheritdoc/>
    9 references
    public override void Copy(IAnimal animal)
    {
        if (animal is IAfricanElephant ad)
        {
            base.Copy(animal);
            Height = ad.Height;
            Weight = ad.Weight;
            TuskLength = ad.TuskLength;
            LongLifespan = ad.LongLifespan;
            SocialBehavior = ad.SocialBehavior;
        }
    }

    #endregion // Public Methods
```

Wszystkie zwierzęta i ich ekrany są dodawane w ten sam sposób.

Tak wygląda wyjście tekstowe do konsoli

```
Your available choices are:
0. Exit
1. Animals
2. Create a new 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: 1

Your available choices are:
0. Exit
1. Dogs
2. Elephants
3. Orangutans
4. Beaver
Please enter your choice: 2

Your available choices are:
0. Exit
1. List all elephants
2. Create a new elephant
3. Delete existing elephant
4. Modify existing elephant
Please enter your choice:
```