Wyższa Szkoła Bankowa

Programowanie Obiektowe

Sprawozdanie z ćwiczenia nr 1

Imię i nazwisko: Mykhailo Zaliznyi

Adres email: zaliznyimh@gmail.com

Numer albumu: **144388**

Data: 27.08.2023

Link do repo na Github: https://github.com/zaliznyimh/ProgramowanieObiektoweZadanie1

Rozdział 1 – Dodanie ustawień dotyczących kolorystyki ekranów

Dla pierwszego zadania byli zmodyfikowane następne Interfejsy oraz Klasy:

Interfejsy: ISettingsService

Klasy: SettingsService, MainScreen, MammalsScreen, AnimalsScreen,

DogsScreen, DolphinsScreen, WolfsScreen, BengalTigersScreen

1) Do interfejsu **ISettingsService** było dodano pole typu String z nazwą «**ColorOfScreen**», również było dodano metodę «**ReadNameOfColor»**

```
Ссылок: 8
T ReadNameOfColor<T>(string screenName, T defaultColor);
```

```
/// <summary>
/// Property to save the the value of text color
/// </summary>
Ссылок: 15
public string? ColorOfScreen { get; set; }
```

2) Do klasy **SettingsService** metoda «**ReadNameOfColor**» została zaimplementowana oraz do tej klasy było dodane pole «**FilePath**» które który zawiera nazwę pliku **JSON**.

Zrzut ekranu metody «ReadNameOfColor»

```
Ссылок: 8
public T ReadNameOfColor <T>(string screenName, T defaultColor)
{
    try
    {
        if (File.Exists(FilePath))
        {
            string json = File.ReadAllText(FilePath);
            dynamic? jsonValue = JsonConvert.DeserializeObject(json);
            if (jsonValue is not null && jsonValue?[screenName] is not null)
            {
                return jsonValue[screenName].ToObject<T>();
            }
        }
    }
    catch
    {
        Console.WriteLine($"There was an error during an attempt to read from {FilePath} file.");
    }

    return defaultColor;
}
```

3) Za pomocą wstrzykiwania zależności na każdym ekranie (**MainScreen, MammalsScreen, AnimalsScreen, DogsScreen, DolphinsScreen, WolfsScreen, BengalTigersScreen**) była wykożystana metoda «**ReadNameOfColor**». Następnie metoda ta jest wywoływana na każdym ekranie.

Zrzut ekranu dla metody **InitialisingColor()** i jak ona wywołuje się w metodzie **Show()** dla ekranu WolfsScreen

```
/// <summary>
/// Method which initialise color of "WolfsScreen"
/// </summary>
Ссылок: 1
private void InitialisingColor()
{
    _settingsService.ColorOfScreen = _settingsService.ReadNameOfColor("WolfsScreen", "White");
    Console.ForegroundColor = (ConsoleColor)Enum.Parse(typeof(ConsoleColor), _settingsService.ColorOfScreen);
}
```

```
/// <summary>
/// Method for showing wolf's main screen
/// </summary>
Ссылок: 2
public override void Show()
{
    while (true)
    {
        InitialisingColor();

        Console.WriteLine();
        Console.WriteLine("Your available choices are:");
        Console.WriteLine("0. Exit");
        Console.WriteLine("1. List all wolfs");
        Console.WriteLine("2. Create a new wolf");
        Console.WriteLine("3. Delete existing wolf");
        Console.WriteLine("4. Modify existing wolf");
        Console.Write("Please choose something: ");
```

4) Pliki JSON dla tego zadania: **ColorSettings.json**

Zrzut ekranu tego pliku JSON:

```
{
  "MainScreen": "Blue",
  "AnimalsScreen": "Magenta",
  "MammalsScreen": "Yellow",
  "DogsScreen": "Green",
  "WolfsScreen": "Cyan",
  "DolphinsScreen": "Red",
  "BengalTigerScreen": "Cyan"
}
```

Rozdział 2 – Dodanie nowego ssaka do programu

W tym zadaniu dodałem nowe zwierzę – Wilka(Wolf).

Dla pierwszego zadania byli zmodyfikowane następne Interfejsy oraz Klasy:

Interfejsy: IWolf,

Klasy: MammalsScreen, Wolf, WolfsScreen, WolfsScreenChoise

1) Do Interfejsu **IWolf** było dodano następujące właściwości: `IsPackHunting`, `PackHunting`, `IsCommunicating`,`Communication`, `Diet`, `IsStrongPaws`, `StrongPaws`, `SenceOfSmell`

Zrzut ekranu Interfejsu IWolf

```
/// <summary> Properties that describe does an wolf hunt in a group
Ссылок: 4
public bool IsPackHunting { get; set; }
Ссылок: 5
public string PackHunting { get; set; }

/// <summary> Property that describes how wolf are communicating
Ссылок: 4
public bool IsCommunicating { get; set; }
Ссылок: 5
public string Communication { get; set; }

/// <summary> Property that describes what wolf eat
Ссылок: 5
public string Diet { get; set; }

/// <summary> Property that describes how wolf use it's paws
Ссылок: 2
public bool IsStrongPaws { get; set; }
Ссылок: 5
public string StrongPaws { get; set; }

/// <summary> Property that how wolf uses it's sence of smell
Ссылок: 5
public string SenceOfSmell { get; set; }
```

2) Powyższe właściwości zostały zaimplementowane w klasie **Wolf**, a metody **Show() i Copy()** zostały zmienione.

```
Ссылок: 4
public override void Display()
{
    Console.WriteLine($"Hi, My name is {Name}, I'm {Age} years old.");
    Console.WriteLine($"Pack hunter: {PackHunting}, Howling communication: {Communication}, Carnivorous diet:{Diet}, " +
                      $"Strong paws: {StrongPaws}, Good sense of smell: {SenceOfSmell}");
}
```

```
public override void Copy(IAnimal animal)
{
    if (animal is IWolf ad)
    {
        base.Copy(animal);
        IsPackHunting = ad.IsPackHunting;
        PackHunting = ad.PackHunting;
        IsCommunicating = ad.IsCommunicating;
        Communication = ad.Communication;
        Diet = ad.Diet;
        StrongPaws = ad.StrongPaws;
        SenceOfSmell = ad.SenceOfSmell;
    }
}
```

3) W klasie **WolfsScreen** dodane metody **WolfList(), CreateWolf(), DeleteWolf(), ModifyWolf(), AddEditWolf()** oraz z tym istnieje metoda **InitialisingColor()** z pierwszego zadania.

4) Również było utworzona **enum Klasa WolfScreenChoise**

```
namespace SampleHierarchies.Enums;

Ссылок: 7
public enum WolfScreenChoise
{
    Exit = 0,
    List = 1,
    Create = 2,
    Delete = 3,
    Modify = 4
}
```

5) Zrzut ekranu konsoli podczas działania WolfsScreen.

```
Your available choices are:
0. Exit
1. List all wolfs
2. Create a new wolf
3. Delete existing wolf
4. Modify existing wolf
Please chooise something: 2
What name of the wolf?: Alex
What is the wolf's age?: 12
Does it hunts in group?(Write Yes or No): No
Does it communicate by howl(Please write Yes or No): No
What does it eat? Small animals
Does the wolf have stong paws?(Please write Yes or No): No
How it helps good sence of smell?: Helps with hunting
Wolf with the name: Alex was added to a list of wolves

Your available choices are:
0. Exit
1. List all wolfs
2. Create a new wolf
3. Delete existing wolf
4. Modify existing wolf
Please chooise something: 1

Here's a list of wolves:
Wolf's number is 1, Hi, My name is Alex, I'm 12 years old.
Pack hunter: Wolf doesn't hunt in group., Howling communication: Wolf doesn't communicate by using howl. It communicate's with gestures
 and smells, Carnivorous diet:Small animals, Strong paws: Wolf doesn't have strong paws, Good sense of smell: Helps with hunting
```

Rozdział 3 – Dodanie kolejnego ssaka do programu

W tym zadaniu dodałem nowe zwierzę – Delfin(Dolphin).

Dla pierwszego zadania byli zmodyfikowane następne Interfejsy oraz Klasy:

Interfejsy: IDolphin,

Klasy: MammalsScreen, Dolphin, DolphinsScreen, DolphinsScreenChoise

1) Do Interfejsu **IDolphin** było dodano następujące właściwości

Zrzut ekranu Interfejsu **IDolphin**

```
/// <summary> Properties which describe the dolpihn's echolocation
Ссылок: 4
public bool UseEcholocation { get; set; }
Ссылок: 5
public string Echolocation { get; set; }

/// <summary> Property which desribes dolphin's social behavior
Ссылок: 5
public string SocialBehavior { get; set; }

/// <summary> Properties which playful behavior
Ссылок: 4
public bool IsPlayfulBehavior { get; set; }
Ссылок: 5
public string PlayfulBehavior { get; set; }

/// <summary> Property which desribes size of dolphin's brain in cubic centimete ...
Ссылок: 5
public int LargeBrain { get; set; }

/// <summary> Properties that desribe with which speed can swim dolphin
Ссылок: 4
public bool IsSwimmingAtHighSpeed { get; set; }
Ссылок: 5
public string SwimmingAtHightSpeed { get;}
```

2) Powyższe właściwości zostały zaimplementowane w klasie **Wolf**, a metody **Show() i Copy()** zostały zmienione.

```
Ссылок: 4
public override void Display()
{
    Console.WriteLine($"Hi, My name is {Name} and I'm {Age} years old. Echolocation: {Echolocation}, Social Behavior: {SocialBehavior}," +
                     $"Playful Behavior {PlayfulBehavior}, Large brain: My brain is {LargeBrain} cubic centimeters, Swimming at high speed: {SwimmingAtHightSpeed}");
}

/// <summary>
/// Method for copying species from doplhins
/// </summary>
/// <param name="animal"></param>
Ссылок: 9
public override void Copy(IAnimal animal)
{
    if (animal is IDolphin ad)
    {
        base.Copy(animal);
        UseEcholocation = ad.UseEcholocation;
        Echolocation = ad.Echolocation;
        SocialBehavior = ad.SocialBehavior;
        IsPlayfulBehavior = ad.IsPlayfulBehavior;
        PlayfulBehavior = ad.PlayfulBehavior;
        LargeBrain = ad.LargeBrain;
        IsSwimmingAtHighSpeed = ad.IsSwimmingAtHighSpeed;
        SwimmingAtHightSpeed = ad.SwimmingAtHightSpeed;
    }
}
```

3) W klasie Dolphin**Screen** dodane metody Dolphin**List(), Create**Dolphin **(), Delete**Dolphin **(), ModifyDolphin(), AddEdit**Dolphin **()** oraz z tym istnieje metoda **InitialisingColor()** z pierwszego zadania.

4) Również było utworzona **enum Klasa DolphinsScreenChoise**

```
namespace SampleHierarchies.Enums;

Ссылок: 7
public enum DolphinsScreenChoices
{
    Exit = 0,
    List = 1,
    Create = 2,
    Delete = 3,
    Modify = 4
}
```

5) Zrzut ekranu konsoli podczas działania DolphinScreen.

```
Your available choices are:
0. Exit
1. List all dolphins
2. Create a new dolphin
3. Delete existing dolphin
4. Modify existing dolphin
Please chooise something: 1

There are no dolphins on the list.

Your available choices are:
0. Exit
1. List all dolphins
2. Create a new dolphin
3. Delete existing dolphin
4. Modify existing dolphin
Please chooise something:
```

Rozdział 4 – Dodanie kolejnego ssaka do programu

W tym zadaniu dodałem nowe zwierzę – Bengal Tiger.

Dla pierwszego zadania byli zmodyfikowane następne Interfejsy oraz Klasy:

Interfejsy: IBengalTiger,

Klasy: MammalsScreen, BengalTiger, BengalTigerScreen, BengalTigerScreenChoise

1) Do Interfejsu **IBengalTiger** było dodano następujące

Zrzut ekranu Interfejsu **IBengalTiger**

```
/// <summary> Characteristics of the Bengal tiger as a predator
Ссылок: 4
public bool IsApexPredator { get; set; }
Ссылок: 5
public string ApexPredator { get; set; }

/// <summary> Property containing the Bengal tiger size value
Ссылок: 5
public float LargeSize { get; set; }

/// <summary> Characteristic which describes bengal tiger camouflage
Ссылок: 5
public string CamouflageFur { get; set; }

/// <summary> Characteristics of the Bengal tiger which desribe bengal tiger leg ...
Ссылок: 5
public bool IsPowerfulLegs { get; set; }
Ссылок: 5
public string PowerfulLegs { get; set; }

/// <summary> Property containing the Bengal tiger behavior as a single individu ...
Ссылок: 4
public bool IsSolitaryBehavior { get; set; }
Ссылок: 5
public string SolitaryBehavior { get; set; }
```

2) Powyższe właściwości zostały zaimplementowane w klasie BengalTiger, a metody **Show() i Copy()** zostały zmienione.

```
Ссылок: 4
public override void Display()
{
    Console.WriteLine($"Hi, My name is {Name} and I'm {Age} years old." +
                      $"ApexPredator: {ApexPredator}, LargeSize: {LargeSize}, Camouflage fur: {CamouflageFur}, Powerful legs: {PowerfulLegs}," +
                      $"Solitary behavior: {SolitaryBehavior}");
}

Ссылок: 9
public override void Copy(IAnimal animal)
{
    if (animal is IBengalTiger ad)
    {
        base.Copy(animal);
        IsApexPredator = ad.IsApexPredator;
        ApexPredator = ad.ApexPredator;
        LargeSize = ad.LargeSize;
        CamouflageFur = ad.CamouflageFur;
        IsPowerfulLegs = ad.IsPowerfulLegs;
        PowerfulLegs = ad.PowerfulLegs;
        IsSolitaryBehavior = ad.IsSolitaryBehavior;
        SolitaryBehavior = ad.SolitaryBehavior;
    }
}
```

3) W klasie **BengalTigerScreen** dodane metody **BengalTigerList(), CreateBengalTiger(), DeleteBengalTiger(), ModifyBengalTiger() , AddEditBengalTiger()** oraz z tym istnieje metoda **InitialisingColor()** z pierwszego zadania.

4) Również było utworzona **enum Klasa BengalTigerScreenChoise**

```
namespace SampleHierarchies.Enums;

Ссылок: 7
public enum BengalTigerScreenChoice
{
    Exit = 0,
    List = 1,
    Create = 2,
    Delete = 3,
    Modify = 4
}
```

5) Zrzut ekranu konsoli podczas działania **BengalTigerScreen**.

```
Your available choices are:
0. Exit
1. List all bengal tigers
2. Create a new bengal tiger
3. Delete existing tiger
4. Modify existing tiger
Please choose something: 1

Here's a list of bengal tigers:
Dolphin's number is 1, Hi, My name is Fido and I'm 13 years old.ApexPredator: Hunting for the food, LargeSize: 6, Camouflage fur: Beaut
iful gray fur, Powerful legs: Bengal tiger has weak legs,Solitary behavior: Fight's for the teritory

Your available choices are:
0. Exit
1. List all bengal tigers
2. Create a new bengal tiger
3. Delete existing tiger
4. Modify existing tiger
Please choose something:
```

Dodatek: JSON file z ze stworzonymi zwierzętami(Animals.json):


```json
    ],
    "Wolves": [
      {
        "IsPackHunting": false,
        "PackHunting": "Wolf doesn't hunt in group.",
        "IsCommunicating": true,
        "Communication": "Used to locate other pack members and establish territory",
        "Diet": "Eats small rodents and birds",
        "IsStrongPaws": true,
        "StrongPaws": "Strong paws help the wolf to get better food and fight for territory",
        "SenceOfSmell": "Can detect prey from a distance of over 1.5 kilometers",
        "Species": 2,
        "Name": "Fido",
        "Age": 5
      }
    ],
    "Dolphins": [
      {
        "UseEcholocation": true,
        "Echolocation": "Allows for navigation and communication",
        "SocialBehavior": "Has complex social hierarchies",
        "IsPlayfulBehavior": false,
        "PlayfulBehavior": "Dolphin doesn't like playing with dolphins. ",
        "LargeBrain": 23,
        "IsSwimmingAtHighSpeed": true,
        "SwimmingAtHightSpeed": "Can travel long distances in search of food",
        "Species": 3,
        "Name": "Dambo",
        "Age": 3
      }
    ],
    "BengalTigers": [
      {
        "IsApexPredator": false,
        "ApexPredator": "Bengal tiger doesn't show it's apex predator features",
        "LargeSize": 4.0,
        "CamouflageFur": "Brawn fur",
        "IsPowerfullLegs": true,
        "PowerfullLegs": "Can run at speeds up to 65 kilometers per hour",
        "IsSolitaryBehavior": true,
        "SolitaryBehavior": "Can travel long distances alone in search of prey",
        "Species": 4,
        "Name": "Alex",
        "Age": 3
      }
    ]
  }
```