|  |
| --- |
| Wyższa Szkoła Bankowa |
| Programowanie Obiektowe |
| Ćwiczenia 1 - zadania |

|  |
| --- |
| Autor: Lesław Pawlaczyk  2023/03/22 |

Spis treści

[Rozdział 1 – Omówienie Struktury 2](#_Toc130398276)

[Rozdział 2 – Działanie programu 3](#_Toc130398277)

[Rozdział 3 – Omówienie interfejsów 5](#_Toc130398278)

[Rozdział 4 – Stosowane techniki 6](#_Toc130398279)

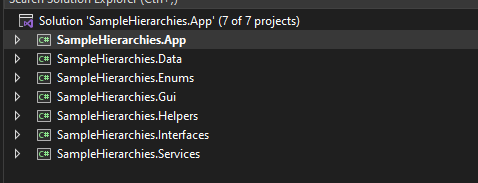
[Rozdział 5 – Zadania do zrealizowania 7](#_Toc130398280)

[Rozdział 6 – Ocenianie 13](#_Toc130398281)

[Bibliografia 14](#_Toc130398282)

# Rozdział 1 – Omówienie Struktury

Na rysunku poniżej widzimy strukturę projektu:



Poniżej wyjaśnienie celów poszczególnych projektów:

1. SampleHierarchies.App to główny projekt uruchomieniowy z definicją startu programu oraz hostem budującym wstrzykiwanie zależności.

2. SampleHierarchies.Data – to projekt z definicjami klas opisującymi dane, które są wymieniane w systemie. Można je podpiąć do kategorii Data Transfer Objects.

3. SampleHierarchies.Enums – enumeracje, które są między innymi wykorzystywane w wyborach na ekranach wprowadzania danych.

4. SampleHierachies.Gui – to są definicje ekranów dla poszczególnych funkcjonalności programu (przykładowo ekran wprowadzania danych na temat psów).

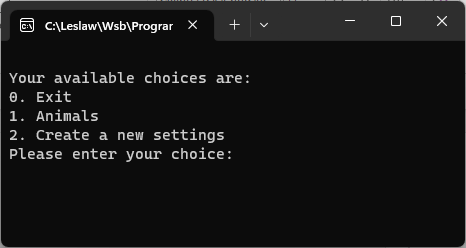
5. SampleHierarchies.Helpers – są to klasy pomocnicze, często będące rozwinięciem istniejących już klas.

6. SampleHierarchies.Interfaces – są to interfejsy używane w programie. Należy przyjąć zasadę, że każdy serwis oraz każda klasa typu DTO powinna mieć swój interfejs.

7. SampleHierarchies.Services – klasy serwisowe wykonujące logikę biznesową.

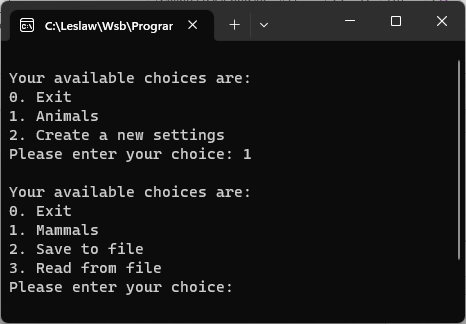
# Rozdział 2 – Działanie programu

Program po wystartowaniu ma następujący układ:

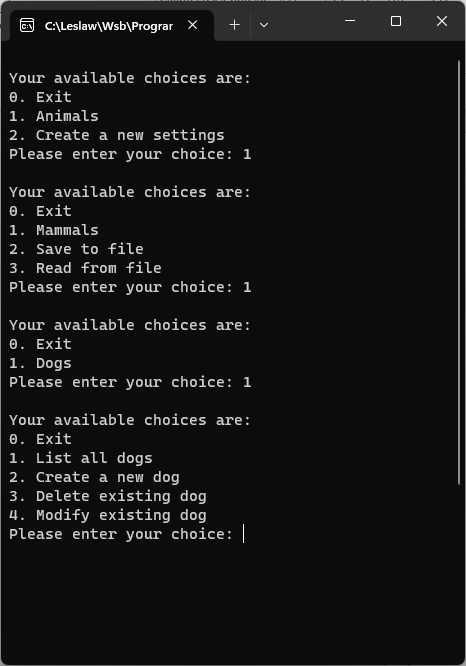


Aby przejść do dalszych części należy wybierać opcję od 0 do 2.

Obecnie wspierane opcje to ścieżka wszystkie opcje z ekranu poniżej:



Są to opcje Mammals->Dogs

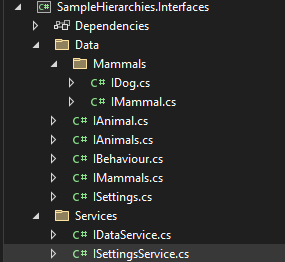


Program wspiera:

1. Zapis danych o zwierzętach do pliku JSON.
2. Odczyt danych o zwierzętach z pliku JSON.
3. Wyświetlanie listy psów.
4. Stworzenie nowego wpisu dotyczącego psa.
5. Usunięcie wpisu dotyczącego wybranego psa.
6. Modyfikacja wpisu dotyczącego wybranego psa.

# Rozdział 3 – Omówienie interfejsów

Poniżej przedstawione są najważniejsze interfejsy:



Poniżej jest lista interfejsów od najbardziej ogólnych do bardziej szczegółowych:

1. IAnimal - zawiera podstawowe informacje o zwierzęciu.
2. IBehaviour – zawiera podstawowe informacje o zachowaniu zwierzęcia.
3. IMammal – interfejs dziedziczący po IAnimal opisujący ssaka.
4. IDog – interfejs opisujący psa.
5. ISettings – interfejs opisujący podstawowe informacje dotyczące ustawień. Obecnie nie zawiera nic.
6. IAnimals – kolekcja obiektów typu IAnimal.
7. IMammals – kolekcje obiektów typu IMammal.
8. IDataService – interfejs z metodami zapisu lub odczytu informacji z pliku JSON dla kolekcji IAnimals.
9. ISettingsService – interfejs z metodami do zapisu lub odczytu ustawień.

# Rozdział 4 – Stosowane techniki

W omówionych szkieletowym rozwiązaniu stosowane następujące techniki:

1. Separacja logiki biznesowej z użyciem osobnych serwisów.
2. Wstrzykiwanie zależności.
3. Stosowanie interfejsów do łatwego testowania jednostkowego.
4. Stosowanie osobnych bibliotek, aby uniknąć przeładowanych funkcjonalnością pojedynczych bibliotek. Stosowanie małych bibliotek jest często stosowane.
5. Stosowanie enumeracji celem unikania „magicznych wartości”.
6. Dokumentacja, każdej najmniejszej części oprogramowania klas, interfejsów.
7. Stosowanie serializacji / deserializacji obiektów do JSON.
8. Stosowanie kolekcji klas.
9. Klasy abstrakcyjne.
10. Metody wirtualne i ich przeciążanie.

# Rozdział 5 – Zadania do zrealizowania

1. Dodanie ustawień dotyczących kolorystyki poszczególnych ekranów:
2. Należy zmodyfikować interfejs ISettings oraz klasę Settings i dodać pola typu string opisująych kolorystykę poszczególnych ekranów. Ekrany, które powinny być opisane to ekran główny, ekran zwierząt, ekran ssaków, ekran psów. Kolory mogą się powtarzać.
3. Dodać opcję zapisu i odczytu ustawień do plików. Zmodyfikować interfejs ISettingsService oraz SettingsService.
4. Dodać wyświetlanie ekranów w kolorze określonym w ustawieniach.
5. 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.
6. Dodać kolejny rodzaj ssaka inny od punktu 2.
7. Dodać kolejny rodzaj ssaka inny od punktów 2 oraz 3.

---------------------------------------------------------------

1. African elephant (Loxodonta africana).

Properties:

1. Height (float)

2. Weight (float)

3. Tusk length (float)

4. Long lifespan (int)

5. Social behavior (string)

Example individuals:

a) Height: 3.3 meters, Weight: 6,000 kilograms, Tusk length: 2 meters, Lifespan: 70 years, Social behavior: Lives in herds led by a matriarch

b) Height: 3.5 meters, Weight: 7,000 kilograms, Tusk length: 2.5 meters, Lifespan: 70 years, Social behavior: Uses infrasonic communication to communicate with other elephants

c) Height: 3.2 meters, Weight: 5,500 kilograms, Tusk length: 1.8 meters, Lifespan: 70 years, Social behavior: Has complex social hierarchies within herds

---------------------------------------------------------------

2. Polar bear (Ursus maritimus).

Properties:

1. Thick fur coat (string)

2. Large paws (string)

3. Carnivorous diet (string)

4. Semi-aquatic (bool)

5. Excellent sense of smell (string)

Example individuals:

a) Fur coat: White and fluffy, Paws: 30 centimeters, Diet: Mostly seals, Semi-aquatic: Can swim for long distances in search of food, Sense of smell: Can detect prey from miles away

b) Fur coat: Yellowish tinge, Paws: 35 centimeters, Diet: Also eats fish and birds, Semi-aquatic: Can dive to depths of over 3 meters, Sense of smell: Can detect prey under snow and ice

c) Fur coat: Thick and oily, Paws: 40 centimeters, Diet: Opportunistic scavenger, Semi-aquatic: Can hunt in water and on land, Sense of smell: Can track prey over long distances

---------------------------------------------------------------

3. Chimpanzee (Pan troglodytes).

Properties:

1. Opposable thumbs (bool)

2. Complex social behavior (string)

3. Tool use (bool)

4. High intelligence (int)

5. Flexible diet (string)

Example individuals:

a) Opposable thumbs: Allows for dextrous manipulation of objects, Social behavior: Lives in groups of up to 100 individuals, Tool use: Can use sticks to extract insects from logs, Intelligence: Can solve complex puzzles, Diet: Eats mainly fruit and leaves

b) Opposable thumbs: Allows for tool use, Social behavior: Has complex social hierarchies, Tool use: Can use rocks to crack nuts, Intelligence: Can communicate using sign language, Diet: Also eats insects and small animals

c) Opposable thumbs: Allows for intricate hand gestures, Social behavior: Learns from older individuals in the group, Tool use: Can use sticks to probe for insects in bark, Intelligence: Can recognize themselves in mirrors, Diet: Eats mainly fruit

---------------------------------------------------------------

4. Lion (Panthera leo).

Properties:

1. Apex predator (bool)

2. Pack hunter (bool)

3. Mane (string)

4. Roaring communication (bool)

5. Territory defense (bool)

Example individuals:

a) Apex predator: Preys on large herbivores, Pack hunter: Hunts in groups, Mane: Dark and full, Roaring communication: Used to signal presence and territorial boundaries, Territory defense: Will fight to defend its territory

b) Apex predator: Can take down prey larger than itself, Pack hunter: Males work together to protect the pride, Mane: Light and spiky, Roaring communication: Can be heard from over 8 kilometers away, Territory defense: Will mark territory with urine

c) Apex predator: Can take down prey larger than itself, Pack hunter: Males work together to protect the pride, Mane: Light and spiky, Roaring communication: Can be heard from over 8 kilometers away, Territory defense: Will mark territory with urine

---------------------------------------------------------------

5. Bottlenose dolphin (Tursiops truncatus).

Properties:

1. Echolocation (bool)

2. Social behavior (string)

3. Playful behavior (bool)

4. Large brain (int)

5. Ability to swim at high speeds (bool)

Example individuals:

a) Echolocation: Used to locate prey, Social behavior: Lives in groups of up to 20 individuals, Playful behavior: Often seen riding waves and interacting with boats, Large brain: Can recognize themselves in mirrors, Ability to swim at high speeds: Can reach speeds of up to 35 kilometers per hour

b) Echolocation: Allows for navigation and communication, Social behavior: Has complex social hierarchies, Playful behavior: Can toss objects back and forth, Large brain: Has a brain-to-body-weight ratio similar to that of humans, Ability to swim at high speeds: Can accelerate quickly to capture prey

c) Echolocation: Used to detect predators and locate food, Social behavior: Can communicate using a variety of sounds, Playful behavior: Often seen performing acrobatic displays, Large brain: Has the ability to process information quickly, Ability to swim at high speeds: Can travel long distances in search of food

---------------------------------------------------------------

6. Grizzly bear (Ursus arctos horribilis).

Properties:

1. Hibernation (bool)

2. Omnivorous diet (string)

3. Large size (float)

4. Curved claws (bool)

5. Good sense of smell (string)

Example individuals:

a) Hibernation: Can hibernate for up to 6 months, Omnivorous diet: Eats both plants and animals, Large size: Can weigh up to 680 kilograms, Curved claws: Used for digging and climbing, Good sense of smell: Can detect prey from a distance of over 1 kilometer

b) Hibernation: Can hibernate for up to 8 months, Omnivorous diet: Eats berries, roots, and small animals, Large size: Can stand over 2 meters tall when standing on hind legs, Curved claws: Used for digging up tubers and roots, Good sense of smell: Can detect a dead animal from over 30 kilometers away

c) Hibernation: Can hibernate for up to 7 months, Omnivorous diet: Eats fish, berries, and small mammals, Large size: Can weigh up to 450 kilograms, Curved claws: Used for digging for insects and roots, Good sense of smell: Can detect carrion from a distance of over 2 kilometers

---------------------------------------------------------------

7. Orangutan (Pongo pygmaeus).

Properties:

1. Arboreal lifestyle (bool)

2. Opposable thumbs (bool)

3. High intelligence (int)

4. Solitary behavior (bool)

5. Slow reproductive rate (bool)

Example individuals:

Arboreal lifestyle: Lives in trees, Opposable thumbs: Allows for tool use and manipulation of objects, High intelligence: Can solve complex problems, Solitary behavior: Lives alone or in small family groups, Slow reproductive rate: Females only give birth every 6-8 years

Arboreal lifestyle: Swings from branches using its arms and legs, Opposable thumbs: Can grasp tree branches and manipulate objects, High intelligence: Can communicate using sign language, Solitary behavior: Males often live alone, Slow reproductive rate: Females only give birth to one offspring at a time

Arboreal lifestyle: Builds nests in trees to sleep in, Opposable thumbs: Can peel bark to access insects, High intelligence: Can recognize itself in mirrors, Solitary behavior: Males may have overlapping territories with other males, Slow reproductive rate: Takes many years for offspring to reach sexual maturity

---------------------------------------------------------------

8. Gray wolf (Canis lupus).

Properties:

1. Pack hunter (bool)

2. Howling communication (bool)

3. Carnivorous diet (string)

4. Strong jaws (bool)

5. Good sense of smell (string)

Example individuals:

a) Pack hunter: Hunts in coordinated groups, Howling communication: Used to locate other pack members and establish territory, Carnivorous diet: Feeds on large herbivores, Strong jaws: Capable of crushing bones, Good sense of smell: Can detect prey from a distance of over 1.5 kilometers

b) Pack hunter: Works together to take down large prey, Howling communication: Can be heard from over 10 kilometers away, Carnivorous diet: Also eats small rodents and birds, Strong jaws: Used for crushing bones and tearing meat, Good sense of smell: Can detect scent trails from prey over long distances

c) Pack hunter: Can hunt animals larger than itself, Howling communication: Used to coordinate hunting strategy and warn of danger, Carnivorous diet: Eats a variety of prey including deer and elk, Strong jaws: Can crush through bone and flesh, Good sense of smell: Can detect prey from a distance of over 2 kilometers

---------------------------------------------------------------

9. Bengal tiger (Panthera tigris tigris).

Properties:

1. Apex predator (bool)

2. Large size (float)

3. Camouflage fur (string)

4. Powerful legs (bool)

5. Solitary behavior (bool)

Example individuals:

a) Apex predator: Preys on large herbivores, Large size: Can weigh up to 300 kilograms, Camouflage fur: Helps to blend in with surroundings when stalking prey, Powerful legs: Allows for fast running and jumping, Solitary behavior: Lives alone except during mating season

b) Apex predator: Can take down prey larger than itself, Large size: Can be over 3 meters long, Camouflage fur: Has distinctive black stripes, Powerful legs: Can leap up to 6 meters in a single bound, Solitary behavior: Has a large territory it patrols alone

c) Apex predator: Can swim across rivers to reach prey, Large size: Can weigh up to 240 kilograms, Camouflage fur: Can vary in color from orange to brown, Powerful legs: Can run at speeds up to 65 kilometers per hour, Solitary behavior: Can travel long distances alone in search of prey

---------------------------------------------------------------

10. Common bottlenose whale (Hyperoodon ampullatus).

Properties:

1. Echolocation (bool)

2. Toothed whale (bool)

3. Long lifespan (int)

4. Sociable behavior (bool)

5. Feeds on squid (string)

Example individuals:

a) Echolocation: Used to locate prey and navigate, Toothed whale: Has a set of large teeth, Long lifespan: Can live up to 60 years, Sociable behavior: Lives in pods of up to 30 individuals, Feeds on squid: Also eats fish and shrimp

b) Echolocation: Has a specialized organ in the forehead to echolocate, Toothed whale: Uses teeth to catch and kill prey, Long lifespan: Can live up to 70 years, Sociable behavior: Can communicate using a variety of vocalizations, Feeds on squid: Can dive to depths of over 1 kilometer to hunt for prey

c) Echolocation: Can detect prey in dark and murky waters, Toothed whale: Has a single pair of teeth that are used for catching squid, Long lifespan: Can live up to 80 years, Sociable behavior: Has complex social structure within pods, Feeds on squid: Uses echolocation to locate and capture prey.

# Rozdział 6 – Ocenianie

**Rozwiązania powinny być indywidualne**.

1. Na podstawie <TODO> wykonać zadania wskazane z rozdziale 5.
2. Wynik w postaci linku do repozytorium na GitHub oraz sprawozdanie w PDF. Sprawozdanie w PDF powinno się nazywać: ProgramowanieObiektowe\_Zadanie1\_20230323\_Rozwiazanie\_LeslawPawlaczyk\_12345.pdf.
3. Repozytorium na Github nie powinno zawierać katalogów obj oraz bin. W katalogu głównym dodać sprawozdanie.

**Strona tytułowa**:

1. Imię i nazwisko.
2. Adres email.
3. Numer albumu.
4. Data.
5. Link do repo na Github.

**Rozdziały**:

1. Każde zadanie rozwiązane w osobnym rozdziale. W rozwiązaniu wskazać jakie klasy oraz interfejsy dodano, które klasy zmodyfikowano. Wystarczą tylko nazwyt, nie trzeba podawać szczegółów modyfikacji.
2. Kod musi być skomentowany.
3. Każde zadanie jeśli ma w rozwiązaniu wskazane plik JSON to dodać go do sprawozdania.
4. Podać zrzuty ekranów.

**Sposób oceniania**:

1. Ocena 3 – zadanie nr 1 oraz 2
2. Ocena 4 – zadania 1, 2, 3.
3. Ocena 5 – zadania 1, 2, 3 oraz 4.

**Oceniane będą**:

1. Jakość kodu.
2. Komentarze.
3. Poprawność działania programu.

**Czas do wykonania**:

1. Do 2023/09/15.

# Bibliografia

1. <https://learn.microsoft.com/pl-pl/dotnet/fundamentals/>
2. <https://learn.microsoft.com/pl-pl/dotnet/csharp/fundamentals/tutorials/oop>
3. <https://learn.microsoft.com/pl-pl/dotnet/standard/design-guidelines/common-design-patterns>
4. <https://learn.microsoft.com/pl-pl/visualstudio/get-started/csharp/tutorial-console?view=vs-2022>