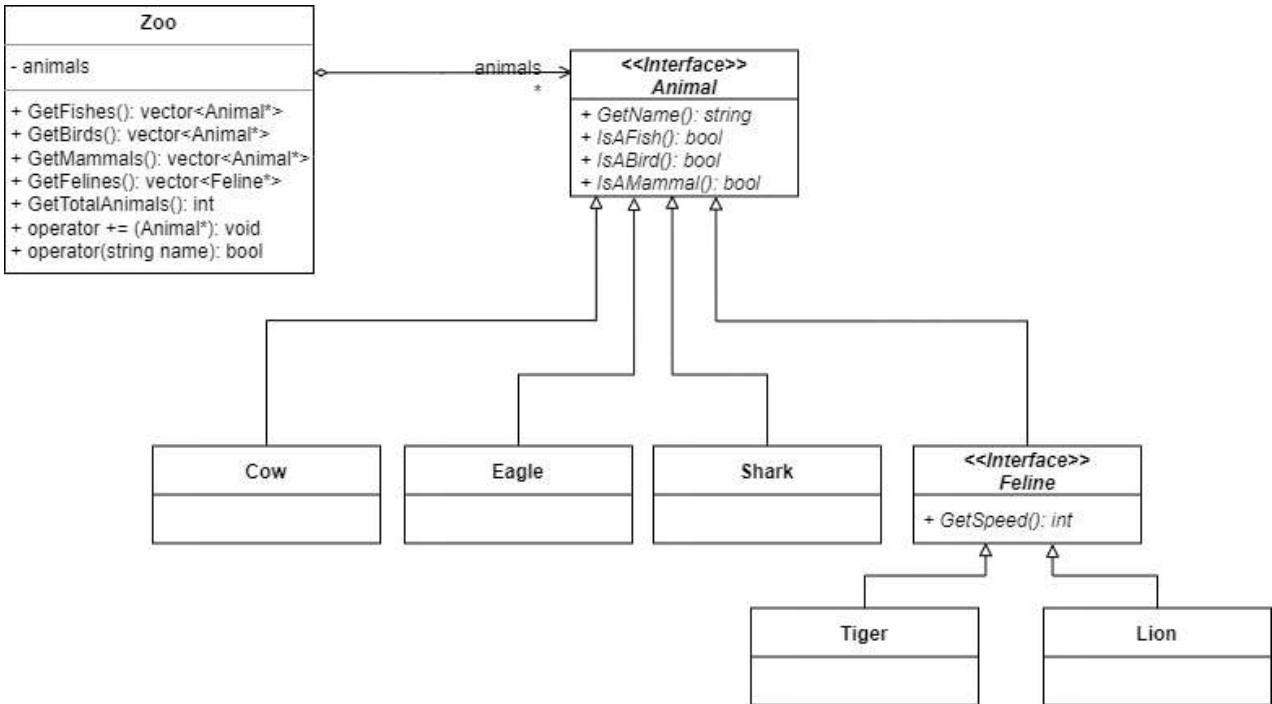


Examen de laborator (partea a 2-a)

Fie următoarea diagrama UML:



Construiți fișierele header si cpp corespunzătoare diagramei de mai sus astfel incat:

```
#include <iostream>
#include "Zoo.h"
#include "Shark.h"
#include "Eagle.h"
#include "Lion.h"
#include "Tiger.h"
#include "Cow.h"
int main()
{
    Zoo z;
    z += new Shark();
    z += new Eagle();
    z += new Tiger();
    z += new Lion();
    z += new Cow();
    std::cout << "Total animals in zoo: " << z.GetTotalAnimals() << std::endl;
    for (auto a : z.GetFishes())
        std::cout << "Fish: " << a->GetName() << std::endl;
    for (auto a : z.GetBirds())
        std::cout << "Bird: " << a->GetName() << std::endl;
    for (auto a : z.GetMammals())
        std::cout << "Mammal: " << a->GetName() << std::endl;
    for (auto a : z.GetFelines())
        std::cout << "Feline: " << a->GetName() << ", speed: " << a->GetSpeed() << std::endl;
    std::cout << "Zoo contains a tiger : " << std::boolalpha << z("Tiger") << std::endl;
    std::cout << "Zoo contains a monkey : " << std::boolalpha << z("Monkey") << std::endl;
    return 0;
}
```

execuția codului din **main** să afișeze pe ecran următoarele:

```
Total animals in zoo: 5
Fish: Shark
Bird: Eagle
Mammal: Tiger
Mammal: Lion
Mammal: Cow
Feline: Tiger, speed: 100
Feline: Lion, speed: 80
Zoo contains a tiger : true
Zoo contains a monkey : false
```

Observatii:

- Clasa **Animal** și toate subclasele derivate din ea trebuie construite inline (doar fișierul header, fără fișierul cpp).
- Exceptând clasa Zoo, nici una dintre celelalte clase NU are date membru ci doar metode! Adăugarea de date membru în aceste clase va impacta următoarele puncte: G₂2, G₂6 și G₂8
- Este recomandată utilizarea unor containere STL pentru rezolvarea acestei probleme.

Punctaj:

G₂1	Organizați proiectul astfel încât să conțină 10 fișiere: main.cpp , Animal.h , Feline.h , Lion.h , Tiger.h , Shark.h , Eagle.h , Cow.h , Zoo.h , and Zoo.cpp	1p WP
G₂2	Organizați fișierul header Zoo.h astfel încât să corespundă exact descrierii din diagrama UML (o dată membru, 2 operatori, 5 metode (getter-e)).	2p WP
G₂3	Implementarea metodei Zoo::GetTotalAnimals()	1p
G₂4	Zoo::GetFishes() , Zoo::GetBirds() și Zoo::GetMammals() (3p pentru fiecare metodă implementată corect)	9p AP
G₂6	Implementarea corectă a metodelor virtuale din clasele Tiger, Lion, Eagle, Cow și Shark. (1p fiecare)	5p
G₂6	Implementarea operatorului +=	1p
G₂7	Implementarea operatorului de tip funcție pentru a verifica dacă un animal există în zoo	3p AP
G₂8	Implementarea metodei Zoo::GetFelines() (presupune utilizarea unui cast dinamic în această metodă)	5p AP
G₂9	Programul compilează corect iar execuția produce un output conform specificațiilor de mai sus.	3p

Criterii:

AP ⇒ capacitatea de a aplica corect criteriile POO (mostenire, polimorfism, etc)

WP ⇒ capacitatea de a scrie un program C++ pe baza unor specificații.

Evaluarea va lua în calcul toate punctele relevante pentru un criteriu din cele prezentate mai sus.