Computing 2 - Labs

Lab 8: Polymorphism among Animals

Suppose you have an inheritance hierarchy with a base class `Animal` and derived classes `Bird`, `Cat`, and `Dog` as shown below.

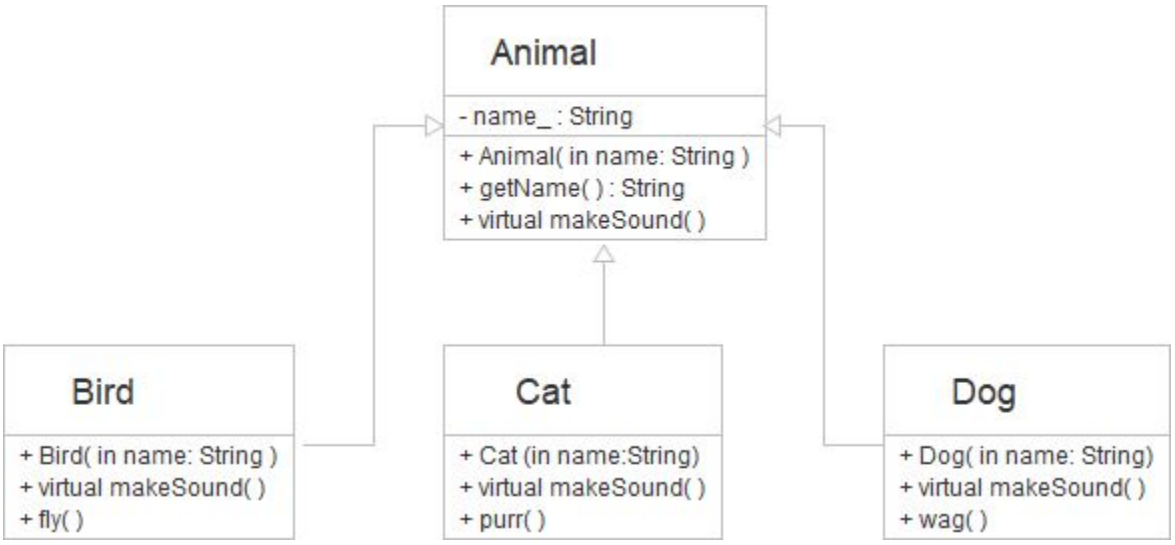

Figure 1 UML diagram showing a base class `Animal` and three derived classes `Bird`, `Cat`, and `Dog`. There is a virtual function `makeSound()` that has to be suitably overridden by the derived classes. We added the comment <<virtual>> to the base class function and to the derived class functions to emphasize `makeSound()is` virtual.

The derived classes each have one constructor of the derived class the virtual function `makeSound`() and another function typical for each animal, `fly()`, `purr()`, `wag()`.The corresponding header files, as well as the file Animal.cpp in which the functions of the class Animal are implemented, are already available.

Task 1:

a) Implement the constructors of the three derived classes.

b) Implement / Override the `makeSound()` function for the `Bird`, `Cat` and `Dog` class. Pay attention to the fact that the name of each animal is `private` in the base class `Animal` and that there is a base class member function `getName()`.
Think about what the base class's `makeSound()` member function should (could) do when called from a base class object. Common to all calls is that first the name of the beast is to be output, then the text " the " followed by the name of the class in small letters then " goes: " and then for a dog the text "bark!\n", for a cat the text "meeow!\n", for a bird the text "tschirp, tschirp\n" can be written.

c) Implement the function `fly()`,`purr()` and `wag`() of the classes Bird, Cat and Dog. The following output should appear: If it is a bird "- " followed by the name of the animal " flies through the skies -\n", if it is a cat "- "followed by the name of the animal " purrs -", if it is a dog "- "followed by the name of the animal " wags tail -"

d) Test your functions with the provided main.cpp file. the output of the element functions of the classes `Bird`,`Cat` and `Dog` should be then:

```
Birdy the bird goes: tschirp, tschirp
- Birdy flies through the skies -
Kitty the cat goes: meeow!
- Kitty purrs -
Puppy the dog goes: bark!
- Puppy wags tail -
```

Task 2 (main.cpp):

Your task is now to complete the main.cpp file code. Now point one pointer of each derived class to the corresponding derived class object (`bird, cat or dog`) and call the two element functions for the class with the pointers. In the case of `Bird`, the code is already present in main.cpp. The previous code for bird leads to the output of the following text:

```
Birdy the bird goes: tschirp, tschirp
- Birdy flies through the skies -
```

If you solve the task in the order cat,dog the following output should then appear additionally:

```
Kitty the cat goes meeow!
- Kitty purrs-
Puppy the dog goes: bark!
- Puppy wags tail -
```

Task 3 (main.cpp):

Further code should now be added to the main.cpp file.. At this point we are now dealing with dynamic binding (polymorphism). To do this, you should set a base class pointer to one of the derived class objects and call the virtual function `makeSound()` via this base class pointer.Add the appropriate code at the designated place in the main.cpp file, letting the pointer point to `cat, dog, bird` one after the other The following output should then be generated:

```
Kitty the cat goes: meeow!
Puppy the dog goes: bark!
Birdy the bird goes: tschirp, tschirp!
```

When all has been implemented, your main program should yield the following result:

```
Static Binding
Birdy the bird goes: tschirp, tschirp!
- Birdy flies through the skies -
Kitty the cat goes: meeow!
- Kitty purrs-
Puppy the dog goes: bark!
- Puppy wags tail -

Aiming derived-class pointers at derived class objects
Birdy the bird goes: tschirp, tschirp!
- Birdy flies through the skies -
Kitty the cat goes: meeow!
- Kitty purrs-
Puppy the dog goes: bark!
- Puppy wags tail -

Dynamic binding by setting a base class pointer to derived class objects
Kitty the cat goes: meeow!
Puppy the dog goes: bark!
Birdy the bird goes: tschirp, tschirp!
```