C#   
INTERFACES

## Objective

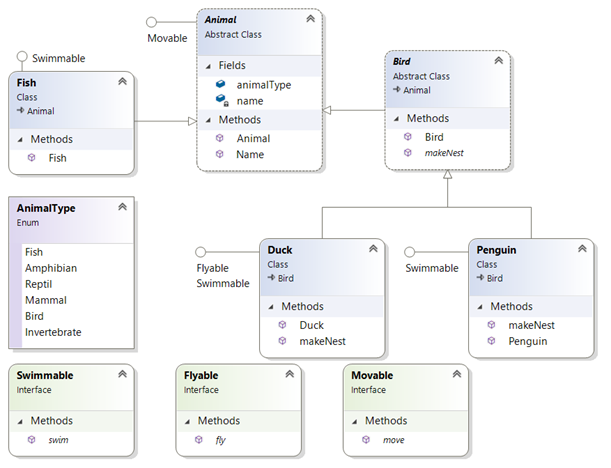
The primary objective of this lab is to provide you with the skills necessary to be able to:

* Define and implement interfaces

## Overview

In this lab you will implement a few interfaces to practice this subject.

The class diagram after completing the lab can be seen below.



### Step by step

1. Following the code you generated in the previous lab, add the following interfaces to the **QALibrary** project
2. **Swimmable** with a method called **Swim**()
3. **Flyable** with a method called **Fly**()
4. **Movable** with a method called **Move**()

See example below for a movable interface:

**public** **interface** **Movable** {

**void** Move();

}

1. Please view the class diagram above and then implement the interfaces.   
   Just display a message for every implemented method. For example in the Duck class:  
     
   **public** **void** **Move** {

**Console.WriteLine**("Moving like a Duck!");

}  
**public** **void** **Fly** {

**Console.WriteLine**("Flying like a Duck!");

}

1. Open the Labs Console’s Program class and write a few lines of code to test the interfaces’ code.
2. Copy the code you wrote for Lab9() method to create a new method called Lab10().
3. Call Lab10() from within the Main() method.
4. In the ‘foreach’ loop that iterates over the **animals** array write an ifstatementto test if an Animal object is Flyable and if it is, cast it as **Flyable** and then call its **Fly()** method.
5. Run your code to make sure it works.
6. Inside the loop, test ifthe Animal object is **Swimmable** and if it is, cast it as Swimmable in order to call its Swim()method.

Please note, the Penguin class extends Bird but implements Swimmable which is an attribute of a fish! Interfaces allow us to do things like this and categorize and group widely different types together. Interfaces allow huge flexibility in your design.

**\*\* End \*\***