



Course : Object Oriented Programming
Practicle Work N°3 : Abstract classes

An abstract class is almost identical to a regular class, just like the ones you're now used to coding. That said, it has one key distinction: you cannot instantiate it. This type of class allows you to define abstract methods, which have a particular feature — they do not have a body!

Note that an abstract method can only exist within an abstract class. If a class contains an abstract method, you must declare the class itself as abstract.

In a zoo, several animals are distinguished by their voices and the way they walk. Each animal can walk, but in its own particular way. This zoo contains several types of animals, namely dogs, cats, lions, and tigers. There are also people who want to visit this zoo.

Each animal has the method `String makeNoise()` reflecting the action of the animal's cry or sound. However, this sound differs depending on the animal. For example, a dog barks while a cat meows.

People, on the other hand, have the method `String speak()` reflecting the action of speaking.

Write a Java program including the following components:

1. **An interface WALKING**
 - Contains the method `String walk()`.
2. **An abstract class Animal**
 - Implements the WALKING interface.
 - Has the attributes: color and weight.
 - Includes two abstract methods:
 - `String makeNoise()`
 - `void eat()`
 - Includes one concrete method:
 - `void drink()`
3. **Two abstract classes Feline and Canine**
 - Both extend the Animal class.
 - Override the `eat()` method.
4. **Four concrete classes:**
 - Cat and Lion extending Feline
 - Dog and Tiger extending Canine
 - Each must have a constructor and implement the `makeNoise()` method.
5. **A class Person**
 - Has the behavior: `String speak()`.
6. **A main class Test**
 - Creates an array of Animal objects.
 - Also creates a Person object.