Code assignment Java developer v2.0

* Given the following code...  **class Animal {**
* **void walk(){**
* **System.out.println("I am walking"); }**  **}**  **class Bird extends Animal { void fly() {**  **System.out.println("I am flying"); }**  **}**  **public class Solution { public static void main(String[] args) {**   **Bird bird = new Bird();**
* **bird.walk();**
* **bird.fly();**
* **bird.sing();**
* **} }**

1. Can you implement the sing() method for the bird?

I chose not to add this method in Bird class. The reason being, All birds can’t sing and those birds which can sing would have to implement CanSing interface. If this needs to be added mandatorily, we can add the method sing() within Bird class or we can make Bird class extend CanSing.

1. How did you unit test it?

I have Used Junit for writing Tests.

1. How did you optimize the code for maintainability?  *(Ask yourself the same question for all following exercises)*

I happened to revisit and add certain methods (both while writing UT and while writing basic code for each class)

1. Now, we have 2 special kinds of birds: the Duck and the Chicken... Can you implement them to make their own special sound?  a. A duck says: “Quack, quack” b. A duck can swim *c.* A chicken says: “Cluck, cluck” *d.* A chicken cannot fly *(assumption: its wings are clipped but ignore that)*

Implemented.

1. Now how would you model a rooster? a. A rooster says: “Cock-a-doodle-doo” b. How is the rooster related to the chicken? c. Can you think of other ways to model a rooster without using inheritance?

I approached this in a way that Rooster is considered to be a Male Chicken. Based on it’s gender, the behaviour like Fly and Speak can be changed.

1. Can you model a parrot? We are specifically interested in three parrots, one that lived in a house with dogs one in a house with cats, the other lived on a farm next to the rooster.  a. A parrot living with dogs says: “Woof, woof” b. A parrot living with cats says: “Meow” c. A parrot living near the rooster says: “Cock-a-doodle-doo” d. How do you keep the parrot maintainable? What if we need another parrot  lives near a Duck? Or near a phone that rings frequently?

Implemented it to match given requirement.

B. Model fish as well as other swimming animals

1. In addition to the birds, can you model a fish? a. Fishes don’t sing  b. Fishes don’t walk c. Fishes can swim
2. Can you specialize the fish as a Shark and as a Clownfish? a. Sharks are large and grey  b. Clownfish are small and colourful (orange) c. Clownfish make jokes d. Sharks eat other fish

Implemented points 1 & 2, as described.

1. Dolphins are not exactly fish, yet, they are good swimmers a. Can you model a dolphin that swims without inheriting from a fish class? b. How do you avoid duplicating code or introducing unneeded overhead?

I actually would like to make Dolphin extend Fish. But, abiding to the given requirement I made Dolphin extend Animal and acquire CanSwim behaviour.

D. Model animals that change their behaviour over time

1. Can you model a butterfly? a. A butterfly can fly

b. A butterfly does not make a sound

2. Can you optimize your model to account for the metamorphosis from caterpillar to butterfly?

a. A caterpillar cannot fly b. A caterpillar can walk (crawl)

Implemented this as per requirement.

E. Counting animals

Suppose you have an array of animals, e.g.

**Animal[] animals = new Animal[]{ new Bird(),**

**new Duck(),**

**new Chicken(),**

**new Rooster(),**

**new Parrot(),**

**new Fish(),**

**new Shark(),**

**new Clownfish(),**

**new Dolhpin(),**

**new Frog(),**

**new Dog(),**

**new Butterfly(),**

**new Cat()**

**};**

**Note**: The above instantiation may be different if you chose to set up your object model differently... *(hopefully you did)*

1. Can you share the code to count: a. how many of these animals can fly? b. how many of these animals can walk? c. how many of these animals can sing? d. how many of these animals can swim?   
Implemented, please refer to the class ***E1CountingOfAnimalsByType.java***

BONUS

If you still have time left, please consider the following:

1. Can you add a second language (if you know a language other than English) Use the rooster as a PoC for demonstrating this. For example, this is how the Rooster sounds differently depending on language. Please add the rooster sound in your native tongue.

* Danish: kykyliky
* Dutch: kukeleku
* Finnish: kukko kiekuu
* French: cocorico
* German: kikeriki
* Greek: kikiriki
* Hebrew: coo-koo-ri-koo
* Hungarian: kukuriku
* Italian: chicchirichi
* Japanese: ko-ke-kok-ko-o
* Portuguese: cucurucu
* Russian: kukareku
* Swedish: kuckeliku
* Turkish: kuk-kurri-kuuu
* Urdu: kuklooku

Couldn’t do this. However, I would model this requirement by using Internationalization.

2. Can you design a RESTful API for querying these animals?

Yes. I can think of following sample APIs and its various methods.

1. <URL>/animals
   1. When used with GET, returns all animals in a readable/json format
2. <URL>/bird
   1. GET –> returns all Birds
   2. PUT -> creates a new bird resource
3. <URL>/flyableanimals
   1. When used with GET, returns all animals which can Fly
4. <URL>/singinganimals
   1. GET method -> returns all animals that can sing
5. <URL>/bird/<birdname>
   1. GET method -> returns all properties of <birdname>
   2. POST method -> updates the properties of <birdname>
6. <URL>/fish
   1. GET -> returns all Fishes
   2. PUT -> creates a new Fish
7. <URL>/fish/<fishname>
   1. GET -> returns all detailed properties of <fishname>
   2. POST -> updates given properties for <fishname>
8. <URL>/mammal
   1. GET -> returns all mammals
   2. PUT -> creates a new mammal
   3. <URL>/mammal/<mammalname>
      1. GET -> returns detailed props of given mammal
      2. POST -> updates given mammal