https://www.jdoodle.com/online-java-compiler

## A. Let's Start with the basics

Q1

class Animal {

void walk() {

System.out.println("I am walking");

}

}

class Bird extends Animal {

void fly() {

System.out.println("I am flying");

}

void sing() {

System.out.println("I am singing");

}

}

public class Solution {

public static void main(String[] args) {

Bird bird = new Bird();

bird.walk();

bird.fly();

bird.sing();

}

}

Q1 (a) I unit tested it

Q1 (B) I optimized the code for maintainability by

Q2

class Animal {

void walk() {

System.out.println("I am walking");

}

}

class Bird extends Animal {

void fly() {

System.out.println("I am flying");

}

void sing() {

System.out.println("I am singing");

}

}

class Duck extends Bird {

void swim() {

System.out.println("I am swimming");

}

void sound() {

System.out.println("Quack, quack");

}

}

class Chicken extends Bird {

void fly() {

System.out.println("I cannot fly");

}

void sound() {

System.out.println("Cluck, cluck");

}

}

public class Solution {

public static void main(String[] args) {

Duck duck = new Duck();

Chicken chicken = new Chicken();

duck.sound();

duck.swim();

chicken.sound();

chicken.fly();

}

}

Q3 (a)

class Animal {

void walk() {

System.out.println("I am walking");

}

}

class Bird extends Animal {

void fly() {

System.out.println("I am flying");

}

void sing() {

System.out.println("I am singing");

}

}

class Duck extends Bird {

void swim() {

System.out.println("I am swimming");

}

void sound() {

System.out.println("Quack, quack");

}

}

class Chicken extends Bird {

void fly() {

System.out.println("I cannot fly");

}

void sound() {

System.out.println("Cluck, cluck");

}

}

class Rooster extends Chicken {

void sound() {

System.out.println("Cock-a-doodle-doo");

}

}

public class Solution {

public static void main(String[] args) {

Rooster rooster = new Rooster();

rooster.sound();

}

}

Q3 (b) A rooster is a male chicken.

Q3 (c) Composition

Q4.

class Animal {

void walk() {

System.out.println("I am walking");

}

}

class Bird extends Animal {

void fly() {

System.out.println("I am flying");

}

void sing() {

System.out.println("I am singing");

}

}

class Duck extends Bird {

void swim() {

System.out.println("I am swimming");

}

void sound() {

System.out.println("Quack, quack");

}

}

class Chicken extends Bird {

void fly() {

System.out.println("I cannot fly");

}

void sound() {

System.out.println("Cluck, cluck");

}

}

class Rooster extends Chicken {

void sound() {

System.out.println("Cock-a-doodle-doo");

}

}

class Parrot extends Bird {

void sound(String name) {

if (name=="dog")

System.out.println("Woof, woof");

if (name=="cat")

System.out.println("Meow");

if (name=="rooster")

System.out.println("Cock-a-doodle-doo");

}

}

public class Solution {

public static void main(String[] args) {

Parrot parrot = new Parrot();

String name;

name = "dog";

parrot.sound(name);

name = "cat";

parrot.sound(name);

name = "rooster";

parrot.sound(name);

}

}

Q4 (d) I keep the parrot code maintainable by adding a parameter to its sound function.

If we need another parrot lives near a Duck or near a phone that rings frequently, we can implement a new sound in the existing code with a new parameter.

## B. Model fish as well as other swimming animals

Q1.

class Animal {

void walk() {

System.out.println("I am walking");

}

}

class Fish extends Animal {

void walk() {

System.out.println("I don't walk");

}

void sing() {

System.out.println("I don't sing");

}

void swim() {

System.out.println("I can swim");

}

}

public class Solution {

public static void main(String[] args) {

Fish fish = new Fish();

fish.sing();

fish.walk();

fish.swim();

}

}

Q2.

class Animal {

void walk() {

System.out.println("I am walking");

}

}

class Fish extends Animal {

void walk() {

System.out.println("I don't walk");

}

void sing() {

System.out.println("I don't sing");

}

void swim() {

System.out.println("I can swim");

}

}

class Shark extends Fish {

void look() {

System.out.println("I am large and grey");

}

void eat() {

System.out.println("I eat other fish");

}

}

class Clownfish extends Fish {

void look() {

System.out.println("I am small and colourful (orange)");

}

void joke() {

System.out.println("I make jokes");

}

}

public class Solution {

public static void main(String[] args) {

Shark shark = new Shark();

Clownfish clownfish = new Clownfish();

shark.look();

clownfish.look();

clownfish.joke();

shark.eat();

}

}

Q3.

class Animal {

void walk() {

System.out.println("I am walking");

}

}

class Fish extends Animal {

void walk() {

System.out.println("I don't walk");

}

void sing() {

System.out.println("I don't sing");

}

void swim() {

System.out.println("I can swim");

}

}

class Dolphin extends Animal {

public void swim() {

System.out.println("I can also swim");

}

}

public class Solution {

public static void main(String[] args) {

Dolphin dolphin = new Dolphin();

dolphin.swim();

}

}

Q3 (b). String class are immutable in Java.

## D. Model animals that change their behaviour over time

class Animal {

void walk() {

System.out.println("I am walking");

}

}

class Butterfly extends Animal {

void fly() {

System.out.println("A butterfly can fly");

}

void sound() {

System.out.println("A butterfly does not make a sound");

}

}

class Caterpillar extends Butterfly {

void fly() {

System.out.println("A caterpillar cannot fly");

}

void walk() {

System.out.println("A caterpillar can walk (crawl)");

}

}

public class Solution {

public static void main(String[] args) {

Butterfly butterfly = new Butterfly();

butterfly.fly();

butterfly.sound();

Caterpillar caterpillar = new Caterpillar();

caterpillar.fly();

caterpillar.walk();

}

}

## E. Counting Animals

Q1.

class Animal {

void walk(){

System.out.println("I am walking");

}

}

class Bird extends Animal {

void fly() {

System.out.println("I am flying");

}

void sing() {

System.out.println("I am singing");

}

}

class Duck extends Bird {

void swim() {

System.out.println("I am swimming");

}

void sound() {

System.out.println("Quack, quack");

}

}

class Chicken extends Bird {

void fly() {

System.out.println("I cannot fly");

}

void sound() {

System.out.println("Cluck, cluck");

}

}

class Rooster extends Chicken {

void sound() {

System.out.println("Cock-a-doodle-doo");

}

}

class Parrot extends Bird {

void sound(String name) {

if (name=="dog")

System.out.println("Woof, woof");

if (name=="cat")

System.out.println("Meow");

if (name=="rooster")

System.out.println("Cock-a-doodle-doo");

}

}

class Fish extends Animal {

void walk() {

System.out.println("I don't walk");

}

void sing() {

System.out.println("I don't sing");

}

void swim() {

System.out.println("I can swim");

}

}

class Shark extends Fish {

void look() {

System.out.println("I am large and grey");

}

void eat() {

System.out.println("I eat other fish");

}

}

class Clownfish extends Fish {

void look() {

System.out.println("I am small and colourful (orange)");

}

void joke() {

System.out.println("I make jokes");

}

}

class Dolphin extends Animal {

public void swim() {

System.out.println("I can also swim");

}

}

class Butterfly extends Animal {

void fly() {

System.out.println("A butterfly can fly");

}

void sound() {

System.out.println("A butterfly does not make a sound");

}

}

class Caterpillar extends Butterfly {

void fly() {

System.out.println("A caterpillar cannot fly");

}

void walk() {

System.out.println("A caterpillar can walk (crawl)");

}

}

class Dog extends Animal {

void sound() {

System.out.println("Woof, woof");

}

}

class Cat extends Animal {

void sound() {

System.out.println("Meow");

}

}

class Frog extends Animal {

void swim() {

System.out.println("can swim");

}

}

public class Solution {

public static void main(String[] args) {

Animal[] animals = new Animal[]{

new Bird(),

new Duck(),

new Chicken(),

new Rooster(),

new Parrot(),

new Fish(),

new Shark(),

new Clownfish(),

new Dolphin(),

new Frog(),

new Dog(),

new Butterfly(),

new Cat()

};

for(int i=0;i<animals.length;i++) {

animals[i].walk();

}

}

}