**ASSIGNMENT NO.5**

Name: NEIL CARDOZ Roll no: 2307012079

Batch: AIML B1

Title: DUCK

1. Main.java

// Main.java

//Name - Neil Cardoz

//PRN - 23070126079

//Batch - AIML B1

public class Main {

public static void main(String[] args) {

MallardDuck md = new MallardDuck();

RubberDuck rd = new RubberDuck();

DecoyDuck dd = new DecoyDuck();

RedheadDuck rh = new RedheadDuck();

System.out.println("-----------------------------");

rd.display();

rd.performQuack();

rd.performSwim();

rd.performFly();

System.out.println("-----------------------------");

md.display();

md.performQuack();

md.performSwim();

md.performFly();

System.out.println("----------------------------");

dd.display();

dd.performQuack();

dd.performSwim();

dd.performFly();

System.out.println("----------------------------");

rh.display();

rh.performQuack();

rh.performSwim();

rh.performFly();

System.out.println("----------------------------");

System.out.println("---------------------------");

}

}

2. MallardDuck.java

public class MallardDuck extends Duck {

public MallardDuck() {

quackBehaviour = new Quack();

swimBehaviour = new Swimming();

flyingBehaviour = new Fly();

}

@Override

public void display() {

System.out.println("I am Mallard Duck");

}

}

3. Floating.java

public class Floating implements SwimBehaviour{

@Override

public void swim() {

System.out.println("I remain Floating......");

}

}

4. FlyingBehaviour.java

public interface FlyingBehaviour {

public void fly();

}

6. MallardDuck.java

public class MallardDuck extends Duck {

public MallardDuck() {

quackBehaviour = new Quack();

swimBehaviour = new Swimming();

flyingBehaviour = new Fly();

}

@Override

public void display() {

System.out.println("I am Mallard Duck");

}

}

7. Nofly.java

public class NoFly implements FlyingBehaviour{

@Override

public void fly() {

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

}

}

8. QuackBehaviour.java

public interface QuackBehaviour {

void quack();

}

9. RedheadDuck.java

public class RedheadDuck extends Duck

{

public RedheadDuck(){ // Constructor

quackBehaviour = new Quack();

swimBehaviour = new Swimming();

flyingBehaviour = new Fly();

}

@Override

public void display() { // method overriding

System.out.println("I am a Redhead Duck !!!");

}

}

10. RubberDuck.java

public class RubberDuck extends Duck {

public RubberDuck() {

quackBehaviour = new Squeak();

swimBehaviour = new Floating();

flyingBehaviour = new NoFly();

}

@Override

public void display() {

System.out.println("I'm a rubber duck");

}

}

11 Sink.java

public class Sink implements SwimBehaviour

{

@Override

public void swim() { // method overriding

System.out.println("I sink....");

}

}

12. Squeak.java

public class Squeak implements QuackBehaviour{

@Override

public void quack() {

System.out.println("Squeak Squeak!!");

}

}

13. SwimBehaviour.java

public interface SwimBehaviour {

public void swim();

}

14. Swimming.java

public class Swimming implements SwimBehaviour{

@Override

public void swim() {

System.out.println("I believe i can Swim !");

}

}

15. Output



5. Repository

https://github.com/Neil-Cardoz/Ducks-Java-Lab/tree/master