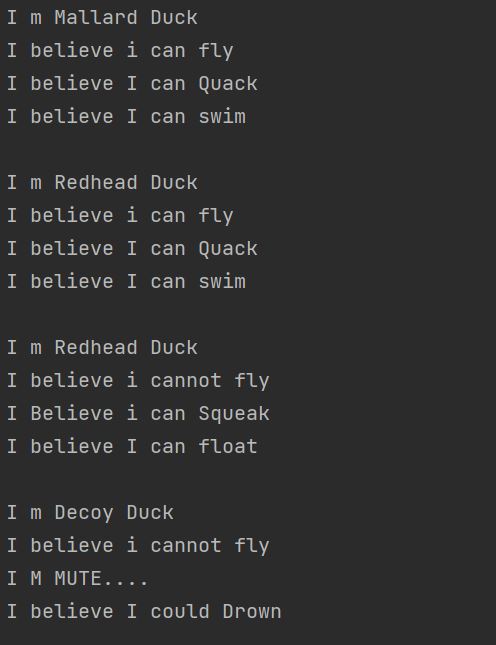
//Nachiket Ropia

//21070126056

public interface FlyBehavior {  
 public void fly();  
}  
public interface QuackBehavior {  
 public void quack();  
}  
public interface SwimBehavior {  
 public void swim();  
}  
abstract class Duck {  
 FlyBehavior flyBehavior;//object of interface  
 QuackBehavior quackBehavior;  
 SwimBehavior swimBehavior;  
 abstract void display();// display would be override  
 public void performQuack(){  
 quackBehavior.quack();  
 }  
 public void performFLy(){  
 flyBehavior.fly();  
 }  
 public void performSwim(){  
 swimBehavior.swim();  
 }  
}  
public class FlyNoWay implements FlyBehavior{  
 @Override  
 public void fly() {  
 System.*out*.println("I believe i cannot fly");  
 }  
}  
public class FlyWithWings implements FlyBehavior{  
 @Override  
 public void fly() {  
 System.*out*.println("I believe i can fly");  
 }  
}  
public class MuteQuack implements QuackBehavior{  
 @Override  
 public void quack() {  
 System.*out*.println("I M MUTE....");  
 }  
}  
public class Quack implements QuackBehavior{  
 @Override  
 public void quack() {  
 System.*out*.println("I believe I can Quack");  
 }  
}  
public class Squeak implements QuackBehavior{  
 @Override  
 public void quack() {  
 System.*out*.println("I Believe i can Squeak");  
 }  
}  
public class Swim implements SwimBehavior{  
 @Override  
 public void swim() {  
 System.*out*.println("I believe I can swim");  
 }  
}  
public class Float implements SwimBehavior{  
 @Override  
 public void swim() {  
 System.*out*.println("I believe I can float");  
 }  
}  
public class Drown implements SwimBehavior{  
 @Override  
 public void swim() {  
 System.*out*.println("I believe I could Drown");  
 }  
}  
public class MallardDuck extends Duck{  
 public MallardDuck() {  
 quackBehavior = new Quack();  
 flyBehavior = new FlyWithWings();  
 swimBehavior=new Swim();  
 }  
 @Override  
 void display() {  
 System.*out*.println("I m Mallard Duck");  
 }  
}  
public class RedHeadDuck extends Duck{  
 public RedHeadDuck(){  
 quackBehavior = new Quack();  
 flyBehavior = new FlyWithWings();  
 swimBehavior=new Swim();  
 }  
 @Override  
 void display() {  
 System.*out*.println("I m Redhead Duck");  
 }  
}  
public class DecoyDuck extends Duck{  
 public DecoyDuck(){  
 quackBehavior = new MuteQuack();  
 flyBehavior = new FlyNoWay();  
 swimBehavior=new Drown();  
 }  
 @Override  
 void display() {  
 System.*out*.println("I m Decoy Duck");  
 }  
}

public class Main {  
 public static void main(String args[]){  
 MallardDuck mallarduck= new MallardDuck();  
 mallarduck.display();  
 mallarduck.performFLy();  
 mallarduck.performQuack();  
 mallarduck.performSwim();  
 System.*out*.println(" ");  
 RedHeadDuck redHeadDuck=new RedHeadDuck();  
 redHeadDuck.display();  
 redHeadDuck.performFLy();  
 redHeadDuck.performQuack();  
 redHeadDuck.performSwim();  
 System.*out*.println(" ");  
 RubberDuck rubberDuck= new RubberDuck();  
 rubberDuck.display();  
 rubberDuck.performFLy();  
 rubberDuck.performQuack();  
 rubberDuck.performSwim();  
 System.*out*.println(" ");  
 DecoyDuck decoyDuck=new DecoyDuck();  
 decoyDuck.display();  
 decoyDuck.performFLy();  
 decoyDuck.performQuack();  
 decoyDuck.performSwim();  
 }  
}

Output



Github Repository Link

https://github.com/NachiketRopia2003/Java-Assignment/tree/main/Assignment%201