**ASSIGNMENT 6**

**Samarth Patel**

**22070126098**

1) Write an application to show the behavior of a Duck.  
2) Create classes as mentioned in <https://www.oreilly.com/api/v2/epubs/0596007124/files/figs/web/022fig01.png.jpg>  
3) Also create a new Behaviour :  
               a) <<interface>> SwimBehavior  
               b) Three different classes  Swim, Float, and Drown which implement SwimBehavior containing swim() method.  
               
4) Print which duck will fly, float or swim.  
5) Create a class diagram for entire program (including swim interface).

**Code:-**

public interface SwimBehaviour {  
 public void swim();  
   
}

public interface FlyBehaviour {  
 public void fly();  
}

public class Float implements SwimBehaviour{  
   
 public void swim(){  
 System.*out*.println("Floating mf");  
 }  
   
}

abstract public class Duck {  
  
 FlyBehaviour flyBehaviour;  
 QuackBehaviour quackBehaviour;  
 SwimBehaviour swimBehaviour;  
  
 public void setFlyBehaviour(FlyBehaviour fb){  
  
 flyBehaviour = fb;  
   
 }  
 public void setQuackBehaviour(QuackBehaviour qb){  
   
 quackBehaviour = qb;  
 }  
 abstract void display();  
  
 public void performFly(){  
   
 flyBehaviour.fly();  
 }  
 public void performQuack(){  
  
 quackBehaviour.quack();  
 }  
  
 public void performSwim(){  
 swimBehaviour.swim();  
 }  
}

public class FlyNoWay implements FlyBehaviour{  
 @Override  
 public void fly(){  
 System.*out*.println("Can't fly mf");  
 }  
}

public class FlyWithWings implements FlyBehaviour{  
 @Override  
 public void fly(){  
 System.*out*.println("Flying with wings mf");  
 }  
}

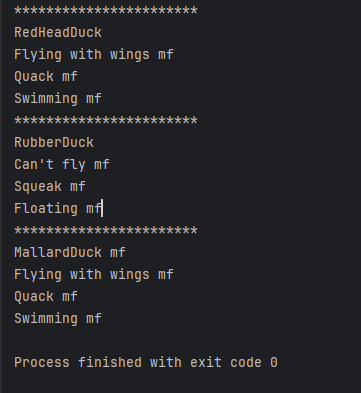
public class MallardDuck extends Duck {  
  
 public MallardDuck(){  
   
 flyBehaviour = new FlyWithWings();  
 quackBehaviour = new Quack();  
 swimBehaviour = new Swim();  
 }  
 @Override  
 void display(){  
 System.*out*.println("MallardDuck mf");  
 }  
   
}

public class NoSwim implements SwimBehaviour{  
 public void swim(){  
 System.*out*.println("Can't swim mf");  
 }  
}

public class Quack implements QuackBehaviour {  
 @Override  
 public void quack(){  
 System.*out*.println("Quack mf");  
 }  
}

public class RubberDuck extends Duck {  
 public RubberDuck(){  
 flyBehaviour = new FlyNoWay();  
 quackBehaviour = new Squeak();  
 swimBehaviour = new Float();  
 }  
  
 @Override  
 void display(){  
 System.*out*.println("RubberDuck");  
 }  
   
}

public class Main {  
 public static void main(String[] args) {  
 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("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  
 MallardDuck mallardDuck = new MallardDuck();  
 mallardDuck.display();  
 mallardDuck.performFly();  
 mallardDuck.performQuack();  
 mallardDuck.performSwim();  
  
 }  
}

**Output:-**

**Class Diagram:-**