JAVA ASSIGNMENT ASSIGNMENT-6 DATE:25-03-23

PROBLEM STATEMENT :

1) Write an application to show the behavior of a Duck.10025
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 the entire program (including the SwimBehavior interface).

-The program should contain different java files.
-Each operation should be a separate function.
- Program should contain at top of the Main file in comments: Name, PRN, Batch
- The program should follow all the coding guidelines.
- The program should contain comments for a particular block of logic.
- It is recommended to upload the Program to your GitHub account.
- Your Strategy repository on GitHub should contain a README file describing all functions or methods or definitions.

.

CODE:

Main

```
/*
Problem Statement :1) Write an application to show the behavior of a Duck.10025
    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 the entire program (including the SwimBehavior interface).

- The program should contain different java files.
- Each operation should be a separate function.
- Program should contain at top of the Main file in comments: Name, PRN, Batch
- The program should follow all the coding guidelines.
- The program should contain comments for a particular block of logic.
- It is recommended to upload the Program to your GitHub account.
- Your Strategy repository on GitHub should contain a README file describing all functions or methods or definitions.

Name - Nisarg Patel
PRN - 21070126060
Batch - AIML A3

LAB ASSIGNMENT - 6

```
 */
package Assignment_6;

public class Main {

    public static void main(String[] args) {
        MallardDuck mallardDuck = new MallardDuck();
        mallardDuck.display();
        mallardDuck.performFly();
        mallardDuck.performQuack();
        mallardDuck.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();

        System.out.println();

        ReadHeadDuck readHeadDuck = new ReadHeadDuck();
        readHeadDuck.display();
        readHeadDuck.performFly();
        readHeadDuck.performQuack();
        readHeadDuck.performSwim();
    }


}
```

Can swim

```
package Assignment_6;

public class CanSwim implements Swim{
    @Override
    public void swim() {
        System.out.println("I can Swim");
    }
}
```

DecoyDuck

```
package Assignment_6;

public class DecoyDuck extends Duck{

    public  DecoyDuck(){
```

```
        flyBehavior = new FlyNoWay();
        quackBehavior = new MuteQuack();
        swim = new Drown();
    }

    @Override
    public void display() {
        System.out.println("I am decoy duck");
    }
}
```

Drown

```
package Assignment_6;

public class Drown implements Swim{
    @Override
    public void swim() {
        System.out.println("I drown in water ;)");
    }
}
```

Duck

```
package Assignment_6;

abstract public class Duck {
    FlyBehavior flyBehavior;
    QuackBehavior quackBehavior;
    Swim swim;

//    public abstract void swim();
    public abstract void display();

    public void performQuack(){
        quackBehavior.quack();
    }

    public void performFly(){
        flyBehavior.fly();
```

```
    }

    public void performSwim(){
        swim.swim();
    }


    public  void setFlyBehavior(FlyBehavior fb){
        flyBehavior = fb;
    }

    public void setQuackBehavior(QuackBehavior qb){
        quackBehavior = qb;
    }

    public void setSwim(Swim s){
        swim = s;
    }


}
```

Float

```
package Assignment_6;

public class Float implements Swim{
    @Override
    public void swim() {
        System.out.println("I can float");
    }
}
```

Fly behaviour

```
package Assignment_6;

public interface FlyBehavior {
    void fly();
}
```

FlyNoWay

```
package Assignment_6;

public class FlyNoWay implements FlyBehavior{
    @Override
    public void fly() {
        System.out.println("I believe I cannot fly");
    }
}
```

FlyWithWings

```
package Assignment_6;

public class FlyWithWings implements FlyBehavior{
    @Override
    public void fly() {
        System.out.println("I believe I can fly");
    }
}
```

MallardDuck

```
package Assignment_6;

public class MallardDuck extends Duck{


    public MallardDuck(){
        quackBehavior = new Quack();
        flyBehavior = new FlyWithWings();
        swim = new CanSwim();
    }


    @Override
    public void display() {
        System.out.println("I am Mallard Duck");
    }
}
```

MuteQuack

```
package Assignment_6;

public class MuteQuack implements QuackBehavior{
    @Override
    public void quack() {
        System.out.println("I am on Mute.....");

    }
}
```

Quack

```
package Assignment_6;

public class Quack implements QuackBehavior{
    @Override
    public void quack() {
        System.out.println("I can quack");
    }
}
```

QuackBehaviour

```
package Assignment_6;

public interface QuackBehavior {
    void quack();
}
```

ReadHeadDuck

```
package Assignment_6;

public class ReadHeadDuck extends Duck{

    public ReadHeadDuck(){
        flyBehavior = new FlyWithWings();
        quackBehavior = new Quack();
```

```
        swim = new CanSwim();
    }



    @Override
    public void display() {
        System.out.println("I am a Read Head Duck");
    }
}
```

RubberDuck

```
package Assignment_6;

public class RubberDuck extends Duck{

    public RubberDuck(){
        quackBehavior = new Squeak();
        flyBehavior = new FlyNoWay();
        swim = new Float();
    }

    @Override
    public void display() {
        System.out.println("I am a Rubber Duck");
    }
}
```

Squake

```
package Assignment_6;

public class Squeak implements QuackBehavior{
    @Override
    public void quack() {
        System.out.println("I can squeak");
        System.out.println("Rubber duckie can quack");
    }
}
```

Swim

```
package Assignment_6;

public interface Swim {
    void swim();
}
```

OUTPUT :

File - Assignment_6.Main

```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent
:C:\Program Files\JetBrains\IntelliJ IDEA Community
Edition 2022.3.1\lib\idea_rt.jar=52804:C:\Program Files
\JetBrains\IntelliJ IDEA Community Edition 2022.3.1\bin
" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -
Dsun.stderr.encoding=UTF-8 -classpath E:\
Assignemnt_Java\out\production\Assignemnt_Java
Assignment_6.Main
I am Mallard Duck
I believe I can fly
I can quack
I can Swim

I am a Rubber Duck
I believe I cannot fly
I can squeak
Rubber duckie can quack
I can float

I am decoy duck
I believe I cannot fly
I am on Mute.....
I drown in water ;)

I am a Read Head Duck
I believe I can fly
I can quack
I can Swim

Process finished with exit code 0
```

Page 1 of 1

GitHub Repository: https://github.com/SnakeEyes1308/Java-Assignment-