**Client**

|  |
| --- |
| ***Duck*** |
| **FlyBehaviour flyBehaviour QuackBehaviour quackBehaviour** |
| swim() display()  performQuack()  performFly()  setFlyBehaviour()  serQuackBehaviour()  // Other duck like methods… |

|  |
| --- |
| **RedHeadDuck** |
| display() {  // looks like a redhead } |

|  |
| --- |
| **MallardDuck** |
| display() {  // looks like a mallard } |

|  |
| --- |
| **RubberDuck** |
| display() {  // looks like a rubberduck } |

|  |
| --- |
| **DecoyDuck** |
| display() {  // looks like a decoy duck } |

**Encapsulated fly behaviour**

|  |
| --- |
| <<interface>>  ***FlyBehaviour*** |
| fly() |

|  |
| --- |
| **FlyWithWings** |
| fly() {  // implements duck flying  } |

|  |
| --- |
| **FlyNoWay** |
| fly() {  // do nothing – can’t fly!  } |

**Encapsulated quack behaviour**

|  |
| --- |
| <<interface>>  ***QuackBehaviour*** |
| quack() |

|  |
| --- |
| **Quack** |
| quack() {  // implements duck quacking  } |

|  |
| --- |
| **Squeak** |
| quack() {  // implements duck squeaking  } |

|  |
| --- |
| **MuteQuack** |
| quack() {  // do nothing – can’t quack  } |

**Relationships**

* **MillardDuck is a Duck**
* **RedHeadDuck is a Duck**
* **RubberDuck is a Duck**
* **DecoyDuck is a Duck**
* **Duck has a FlyBehavior**
* **Duck has a QuackBehaviour**