

Restricted

```
public abstract class BirdTwo {
```

3. Declare an instance Variable to store FlyBehaviour objects and the setter to assign the appropriate FlyBehaviourObject

```
    private FlyBehaviour flyBehaviour;  
  
    public void setFlyBehaviour(FlyBehaviour  
flyBehaviour){  
        this.flyBehaviour = flyBehaviour;  
    }  
  
    public void fly(){  
        if( flyBehaviour != null){  
            flyBehaviour.fly();  
        }  
    }  
}
```

4. The fly() method now executes the fly() method in the FlyBehaviour object.

```
    public void fly(){  
        if( flyBehaviour != null){  
            flyBehaviour.fly();  
        }  
    }  
}
```

1. Declare an interface for objects that describe flying behaviour.

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

2. Write as many FlyBehaviour objects as you need to encapsulate various flying behaviours.

```
class CanFly implements FlyBehaviour{  
  
    @Override  
    public void fly() {  
        System.out.println("I CAN fly");  
    }  
}
```

Restricted