

## Adapter Design Pattern (PassengerPlane.java)

- Created an interface called Plane which declares the method getModel, getcrewcapacity and getpassengercapacity.
- Created a class PassengerPlaneAdapter that uses the implements Plane interface.
- The constructor of PassengerPlaneAdapter takes an instance of PassengerPlane as a parameter.
- The getcrewcapacity() method access the crew capacity fields of the PassengerPlane class while the getpassengercapacity() method access the passenger capacity fields of the PassengerPlane class.
- Using the Plane interface, we can get the passenger and crew capacity of any plane model including passengerPlane instances.

```
package flight.reservation.plane;

1 usage 1 implementation 1 related problem
public interface Plane {
    no usages 1 implementation
    int getcrewcapacity();

    no usages 1 implementation
    int getpassengercapacity();
    no usages 1 implementation
    String getModel();
}
```

```

package flight.reservation.plane;

no usages new *
public class PassengerPlaneAdapter implements Plane {
    4 usages
    private final PassengerPlane passengerPlane;

    no usages new *
    public PassengerPlaneAdapter (PassengerPlane passengerPlane) {
        this.passengerPlane = passengerPlane;
    }

    no usages new *
    @Override
    public int getcrewcapacity() {
        return passengerPlane.crewCapacity;
    }

    no usages new *
    @Override
    public int getpassengercapacity(){
        return passengerPlane.passengerCapacity;
    }

    no usages new *
    @Override
    public String getModel(){
        return passengerPlane.getModel();
    }
}

```