Adapter design pattern is one of the structural design pattern.

An Adapter Pattern says that just "converts the interface of a class into another interface that a client wants".

In other words, to provide the interface according to client requirement while using the services of a class with a different interface.

The Adapter Pattern is also known as **Wrapper**.

There are the following specifications for the adapter pattern:

*Target Interface*: This is the desired interface class which will be used by the clients.

*Adapter class*: This class is a wrapper class which implements the desired target interface and modifies the specific request available from the Adaptee class.

*Adaptee class / Interface* : This is the class/interface which is used by the Adapter class to reuse the existing functionality and modify them for desired use.

*Client*: This class will interact with the Adapter class.

Use-cases :

PowerAdapter and Socket

XML to Json Parser

Weight Machine ( Pounds to KG )

Adaptee :

public interface WeightMachine {  
 *//return the weight in Pound* public double getWeightInPound();  
  
  
}

public class WeightMachineForBabies implements WeightMachine{  
  
 @Override  
 public double getWeightInPound() {  
 return 28;  
 }  
  
}

Adapter :

public interface WeightMachineAdapter {  
  
 public double getWeightInKg();  
  
}

public class WeightMachineAdapterImpl implements WeightMachineAdapter{  
  
 WeightMachine weightMachine; // Adaptee reference  
  
 public WeightMachineAdapterImpl(WeightMachine weightMachine) {  
 this.weightMachine = weightMachine;  
 }  
  
 @Override  
 public double getWeightInKg() {  
  
 double weightInPound = weightMachine.getWeightInPound();  
  
 *//Convert it to KGs* double weightInKg = weightInPound \* .45;  
 return weightInKg;  
 }  
  
  
  
}

Client :

public class AdapterDesignPattern {  
  
 public static void main(String[] args) {  
 WeightMachineAdapter weightMachineAdapter = new WeightMachineAdapterImpl(new WeightMachineForBabies());  
  
 System.*out*.println(weightMachineAdapter.getWeightInKg());  
 }  
}

Output :

12.6

Process finished with exit code 0