**Adapter Design Pattern**

Adapter [design pattern](https://www.journaldev.com/1827/java-design-patterns-example-tutorial) is one of the **structural design pattern** and its used so that two unrelated interfaces can work together. The object that joins these unrelated interface is called an **Adapter**.

One of the great real life example of Adapter design pattern is mobile charger. Mobile battery needs 3 volts to charge but the normal socket produces either 120V (US) or 240V (India). So the mobile charger works as an adapter between mobile charging socket and the wall socket.

We will try to implement multi-adapter using adapter design pattern in this tutorial.

So first of all we will have two classes – Volt (to measure volts) and Socket (producing constant volts of 120V).

1. **Class Adapter** – This form uses [**java inheritance**](https://www.journaldev.com/644/inheritance-java-example) and extends the source interface, in our case Socket class.
2. **Object Adapter** – This form uses [**Java Composition**](https://www.journaldev.com/1325/composition-in-java-example) and adapter contains the source object.

Notice that both the adapter implementations are almost same and they implement the SocketAdapter interface. The adapter interface can also be an [**abstract class**](https://www.journaldev.com/1582/abstract-class-in-java).

### Adapter Design Pattern Example in JDK

Some of the adapter design pattern example I could easily find in JDK classes are;

* java.util.Arrays#asList()
* java.io.InputStreamReader(InputStream) (returns a Reader)
* java.io.OutputStreamWriter(OutputStream) (returns a Writer)

