Structural Design Patterns — this kind of design patterns cope with structural composition of software components to simplify the complex objects using mostly inheritance.

Adapter — this design patterns, as name suggests, is a converter pattern that connects incompatible interfaces. Examples are:

* [java.util.Arrays](https://docs.oracle.com/en/java/javase/15/docs/api/java.base/java/util/Arrays.html) — asList()
* [java.util.Collections](https://docs.oracle.com/en/java/javase/15/docs/api/java.base/java/util/Collections.html) — list(), enumeration()
* [java.io.InputStreamReader](https://docs.oracle.com/en/java/javase/15/docs/api/java.base/java/io/InputStreamReader.html)