-This pattern defines the skeleton of an algorithm in a method,allowing subclasses to override specific steps of the algorithm without changing its overall structure

Template — this design pattern provi- des flexibility to objects by abstracting several sub-parts and letting extenders to implement it. Please see my other [article](https://azeynalli1990.medium.com/template-design-pattern-in-java-and-spring-ddd262f15412) for more information about this pattern. Examples are:

* [java.io.InputStream](https://docs.oracle.com/en/java/javase/15/docs/api/java.base/java/io/InputStream.html)
* [java.io.OutputStream](https://docs.oracle.com/en/java/javase/15/docs/api/java.base/java/io/OutputStream.html)
* [java.io.Reader](https://docs.oracle.com/en/java/javase/15/docs/api/java.base/java/io/Reader.html)
* [java.io.Writer](https://docs.oracle.com/en/java/javase/15/docs/api/java.base/java/io/Writer.html)