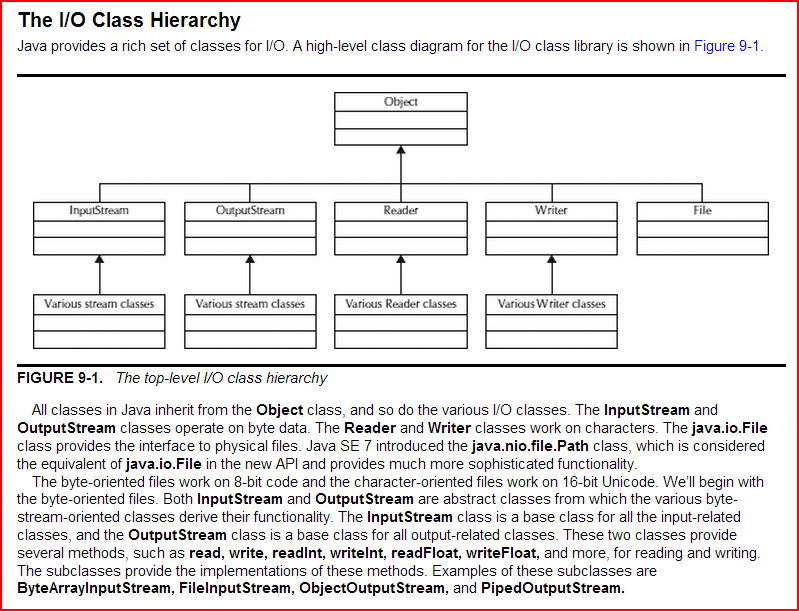
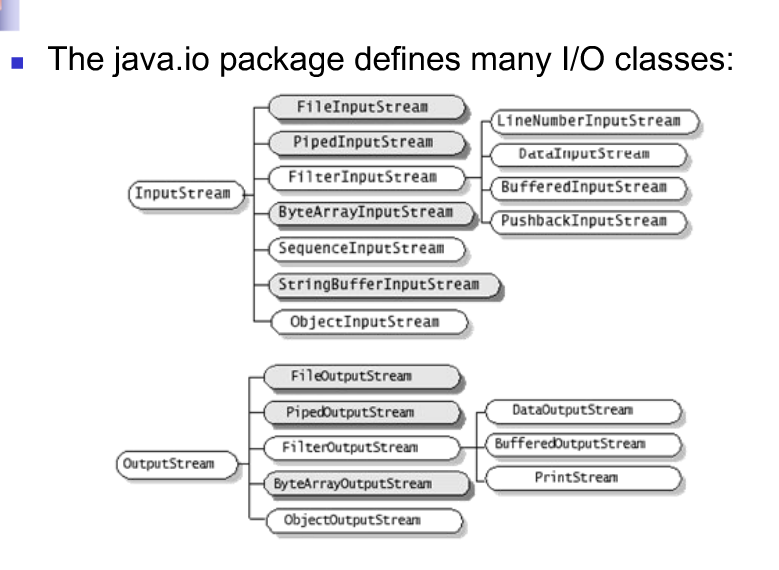
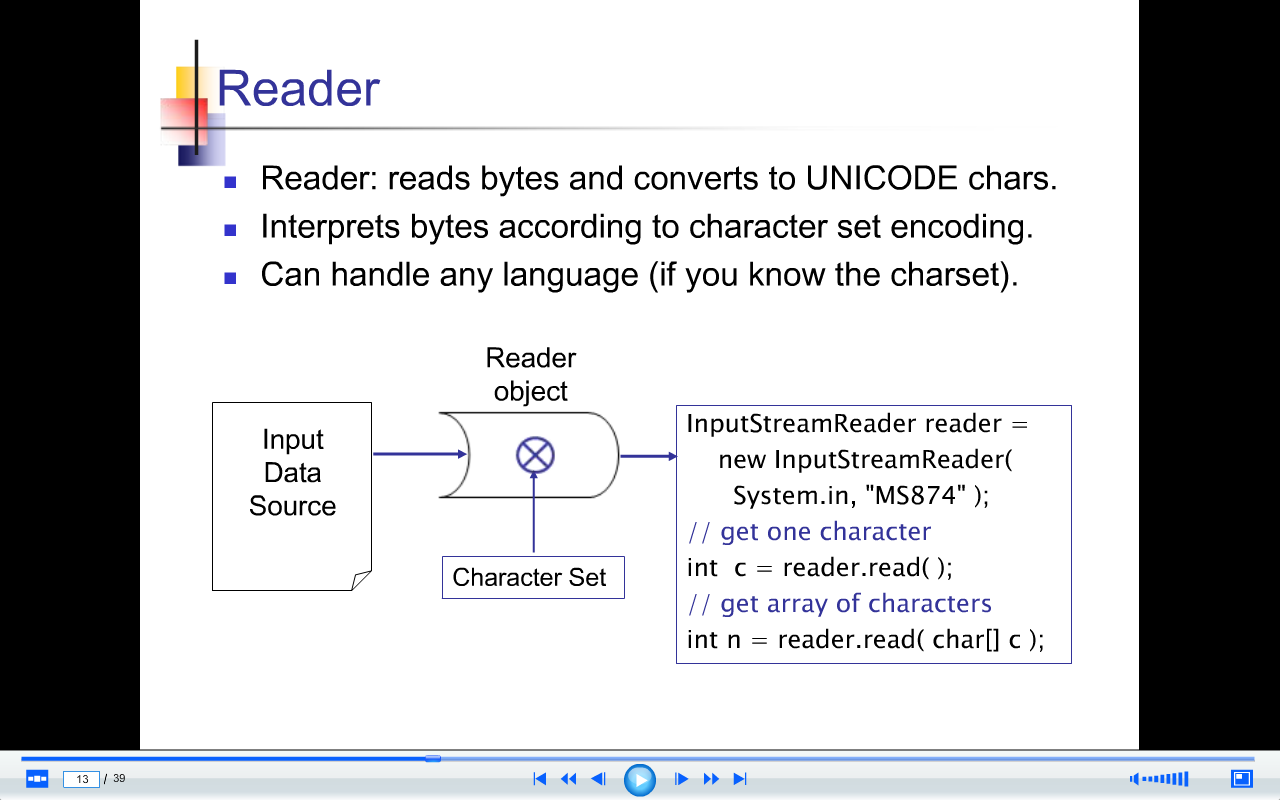
JAVA I/O package- safari books online.





A typical usage of Decorator pattern is Java IO classes.

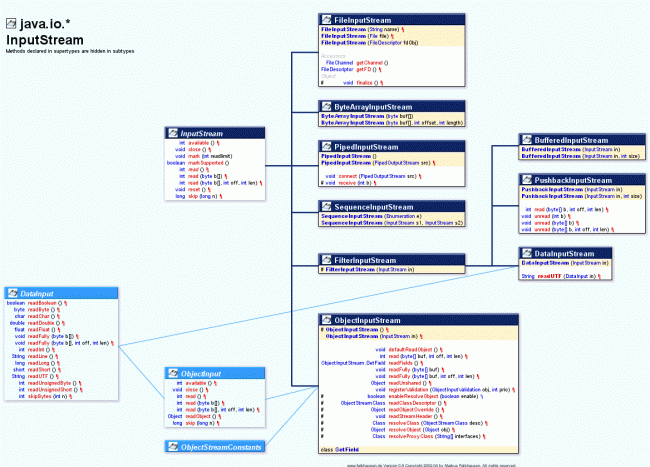
Here is a simple example - BufferedReader decorates InputStreamReader.

|  |
| --- |
| BufferedReader input = **new** BufferedReader(**new** InputStreamReader(System.in));  *//System.in is an InputStream object* |

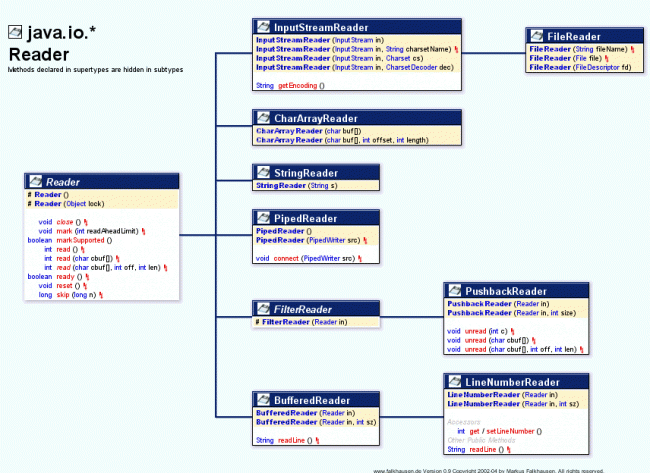
I**nputStreamReader**(InputStream in) - bridge from byte streams to character streams. InputSteamReader reads bytes and translates them into characters using the specified character encoding.

**BufferedReader(Reader in)** - read text from a character stream and buffer characters in order to provide efficient reading methods(e.g., readLine()).

**Class hierarchy diagram for java.io.inputstream**

[](http://www.programcreek.com/wp-content/uploads/2012/05/java-io-class-hierarchy-diagram-inputStream.gif)

**Class hierarchy diagram for java.io.reader**

[](http://www.programcreek.com/wp-content/uploads/2012/05/java-io-class-hierarchy-diagram.gif)

Streams are byte-oriented or character-oriented. Each type has input streams and output streams.

1. Byte-oriented streams.
   * Use for general-purpose input and output.
   * Data may be primitive data types or raw bytes.
2. Character-oriented streams.
   * Specialized for character data.
   * Transforms data from/to 16 bit Java *char* used inside programs to UTF format used externally.