

OCP Java SE 8

I/O Fundamentals

File

- Abstrakte Representation eines Pfads zu einer Datei oder einem Verzeichnis
- Pfad kann *absolut* oder *relativ* sein
- Kann Dateien und Verzeichnisse erzeugen, löschen, ihre Attribute abfragen und manipulieren

Streams

- Stream-Klassen sind für die Eingabe und Ausgabe von binär oder Byte-Daten gedacht
- Reader und Writer Klassen sind für Ein- und Ausgabe von Char- und String-Daten gedacht

Abstrakte Klassen

- InputStream
- OutputStream
- Reader
- Writer

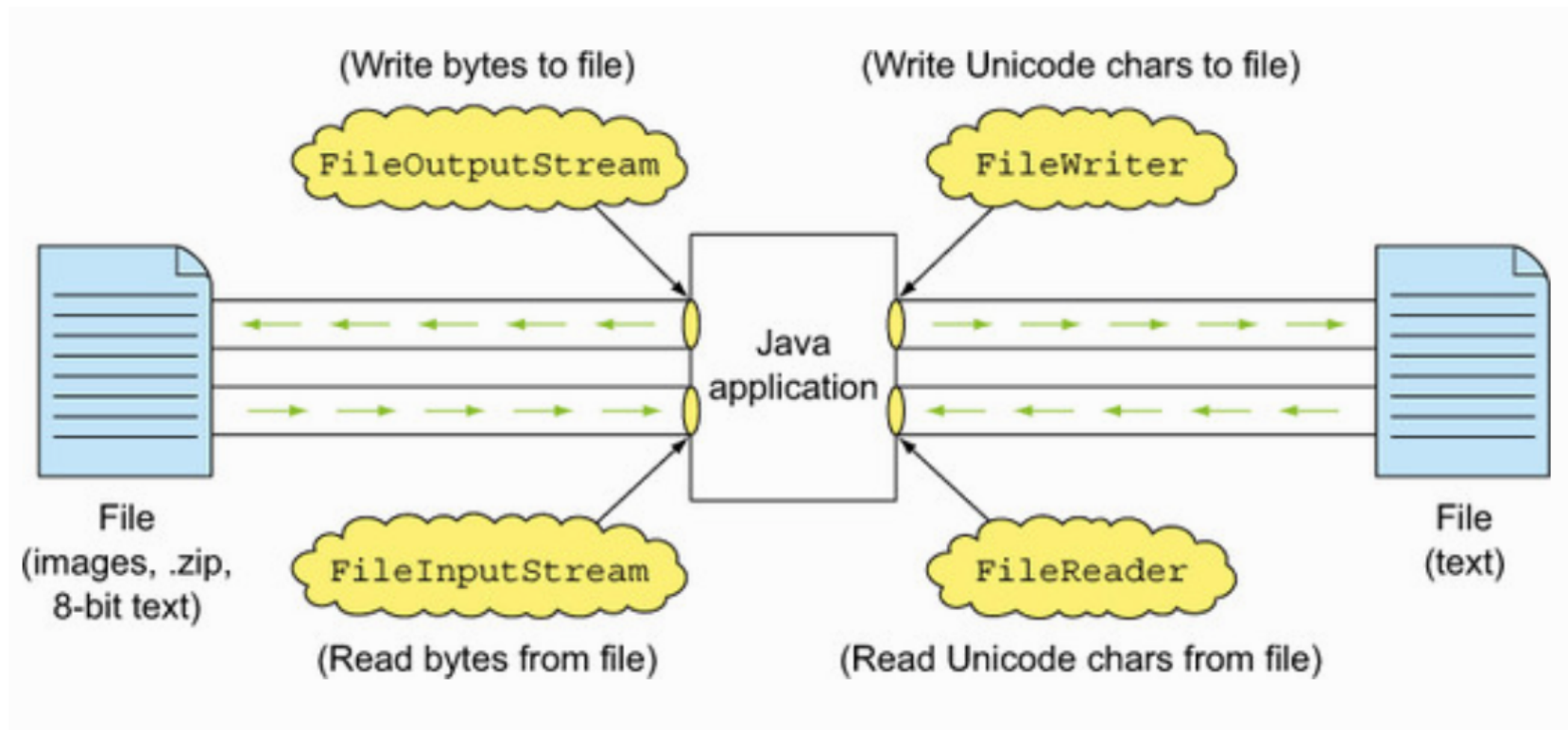
Streams

- Low-Level
 - Verbindet sich direkt mit der Ressource
- High-Level
 - Verwendet einem anderen Stream als Basis
 - einige Operationen auf dem high-level werden an die Basis weitergegeben (z.B. close)

Streams (Low-Level)

- FileInputStream
 - liest Inhalt der Datei als bytes
- FileOutputStream
 - schreibt Inhalt der Datei als bytes
- FileReader
 - liest Inhalt der Datei als Zeichen
- FileWriter
 - schreibt Inhalt der Datei als Zeichen

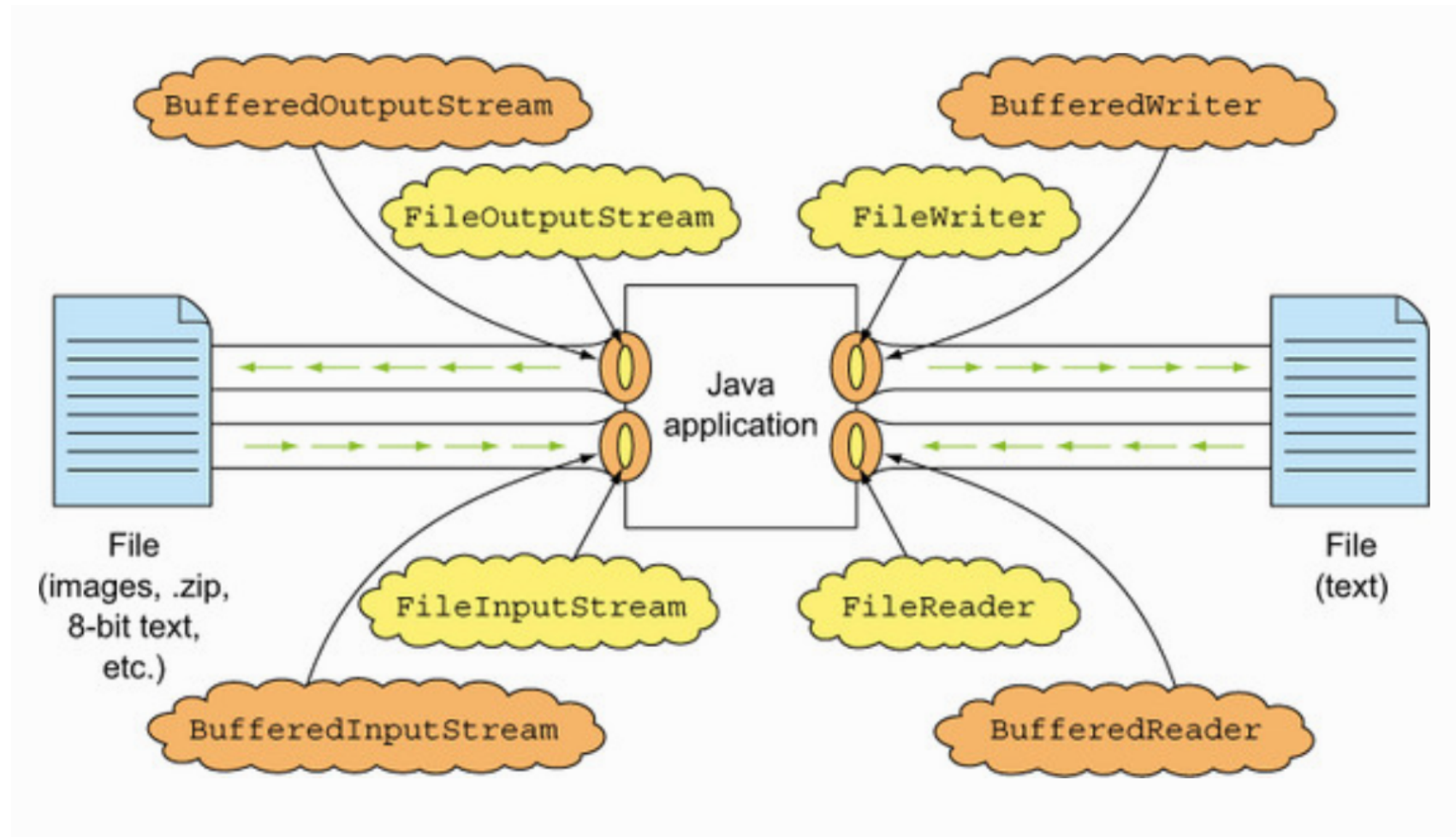
Streams



Buffering

- In vielen Fällen wird nach einem write nicht sofort geschrieben sondern es wird gewartet, bis sich eine gewisse Menge von Daten in Puffer (engl. Buffer) angesammelt hat.
- Mit der flush-Methode kann man das Schreiben erzwingen
- `public void flush()`
- Schreiben aller noch ausstehenden Daten, alles was sich durch Aufrufe von write angesammelt hat, aber noch nicht geschrieben worden ist.

Buffered Streams



Streams (High-Level)

- `BufferedReader`
- `BufferedWriter`
- `ObjectInputStream`
 - Deserialisierung von Java-Objekten
- `ObjectOutputStream`
 - Serialisierung von Java-Objekten

Streams (High-Level)

- InputStreamReader
 - Liest Zeichen aus einem InputStream
- OutputStreamWriter
 - Schreibt Zeichen in ein OutputStream
- PrintStream
 - Schreibt formatierte Representationen von Java-Objekten in ein Binary-Stream
- PrintWriter
 - Schreibt formatierte Representationen von Java-Objekten in ein Zeichenbasierten-Output-Stream

Buffered Streams

- Decorator Klassen
 - BufferedInputStream
 - BufferedOutputStream

Data Streams

- Lesen und Schreiben primitiver Datentypen
- Verwendet FileInputStream/FileOutputStream
 - DataInputStream
 - DataOutputStream

Object Streams

- Lesen und Schreiben von Objekten
- Verwendet FileInputStream/FileOutputStream
- Objekte müssen serialisierbar sein
 - ObjectOutputStream
 - ObjectInputStream

Reader

- implementiert AutoCloseable
 - FileReader
 - BufferedReader

Writer

- implementiert AutoCloseable
 - FileWriter
 - BufferedWriter

PrintWriter

- Ermöglicht ein formatiertes Schreiben
 - write
 - print
 - println
 - format
 - printf

Console

- Ermöglicht das Lesen und Schreiben von und auf der Console
- Kein public Konstruktor, wird über `System.console()` geholt

Console

- `Console format(String fmt, Object... args)`
 - Schreibt einen formatierten String in die Console
- `Console printf(String format, Object... args)`
 - Übliche Methode für formatierte Ausgabe
- `String readLine()`
 - Liest eine Zeile von der Console
- `String readLine(String fmt, Object... args)`
 - Ein formatierter Prompt. Liest eine Zeile von der Console
- `char[] readPassword()`
 - Liest ein Password (es gibt keine Ausgabe)
- `char[] readPassword(String fmt, Object... args)`
 - Ein formatierter Passwort Prompt (es gibt keine Ausgabe)