

Vererbung in der Java-API

Beispiel `java.io`

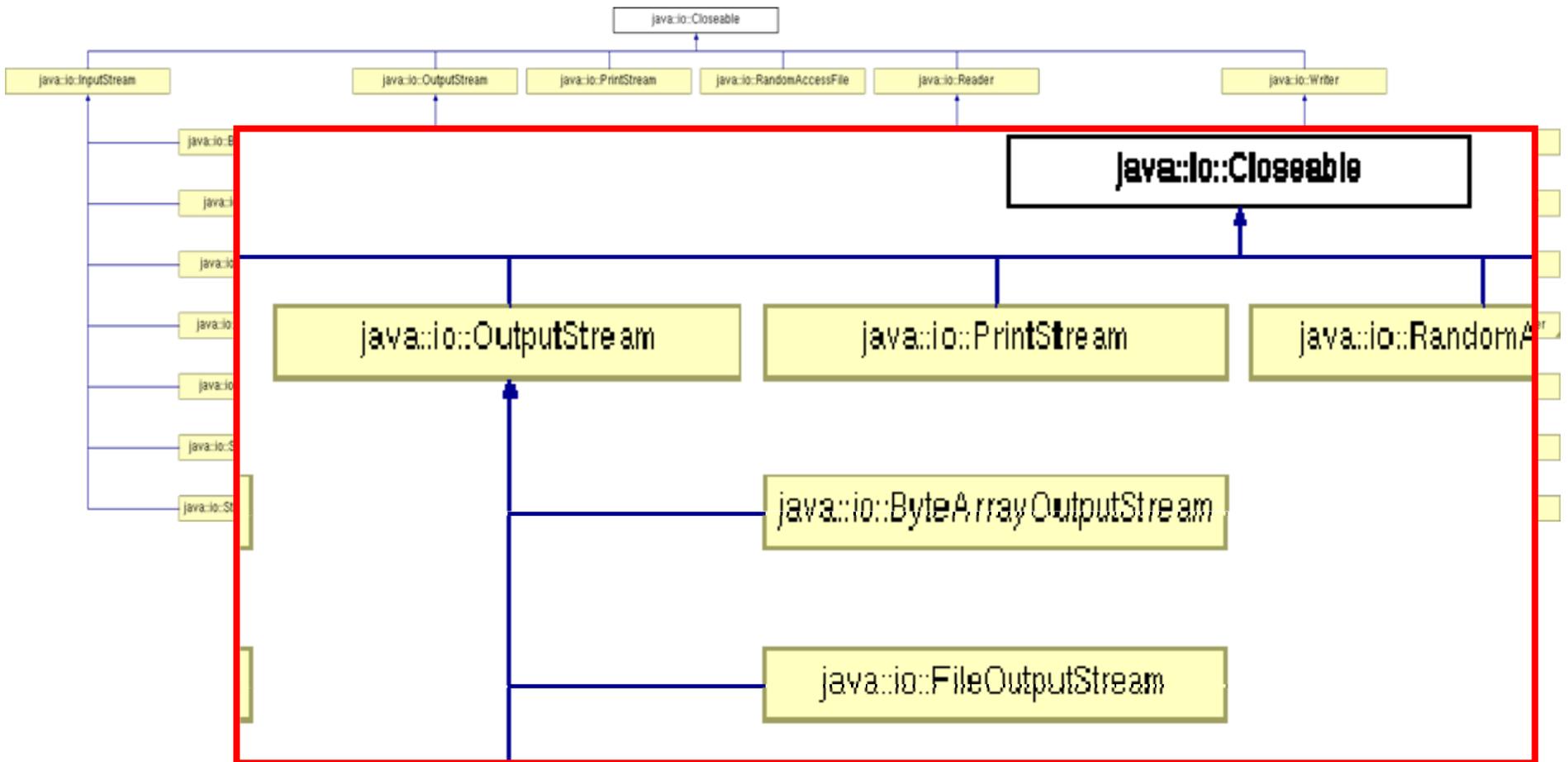
Im Folgenden: Keine korrekte UML-Syntax

java.io

- API für Ein-/Ausgaben
- So generisch wie möglich:
 - Files
 - Netzwerkverbindungen
 - Objektspeicherung
 - Performanz und

Was haben alle diese Anforderungen
gemeinsam?

java.io.Closeable



java.io.Closeable

- `void java.io.Closeable::close`
() throws [IOException](#)

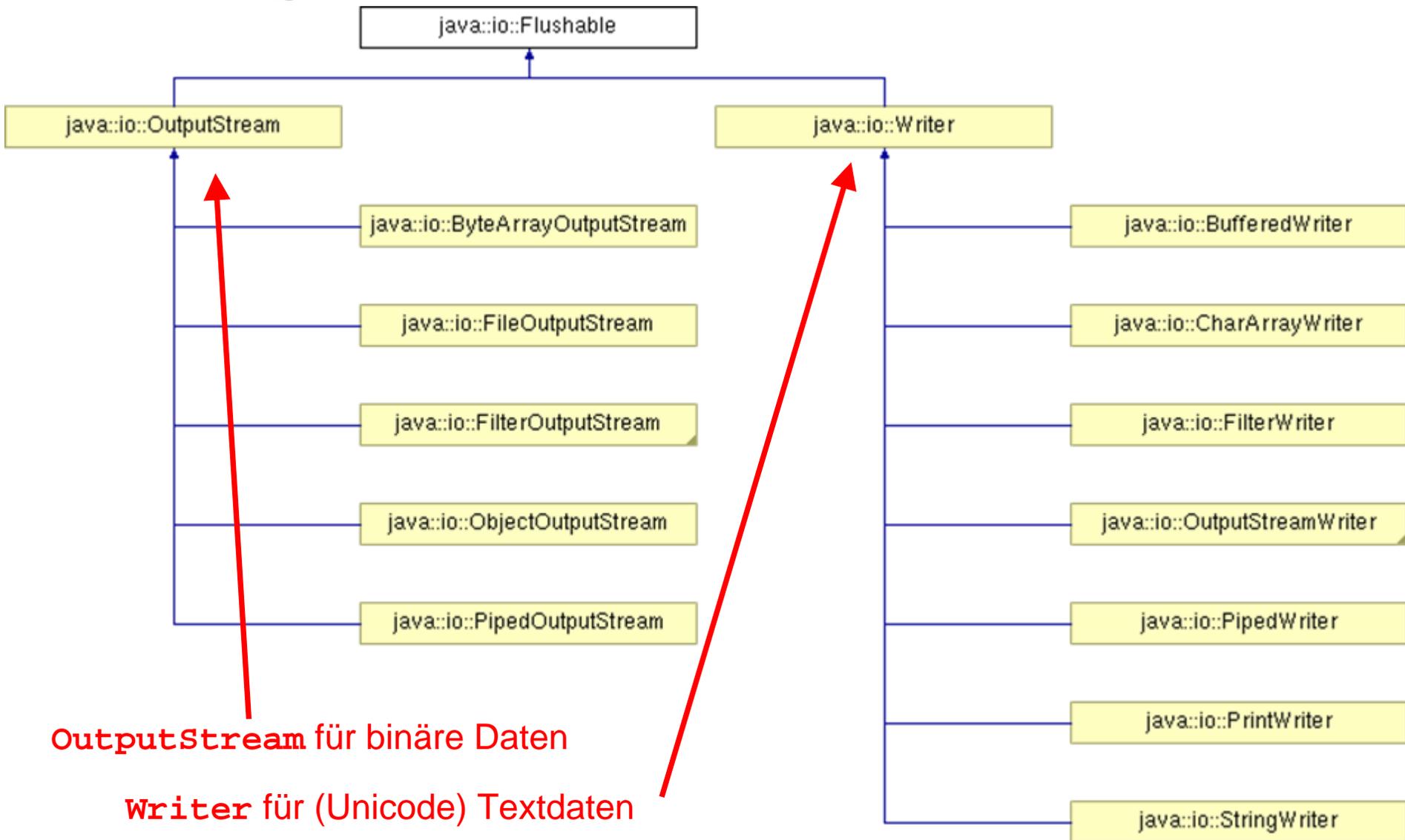
Closes this stream and releases any system resources associated with it. If the stream is already closed then invoking this method has no effect.

Daten speichern

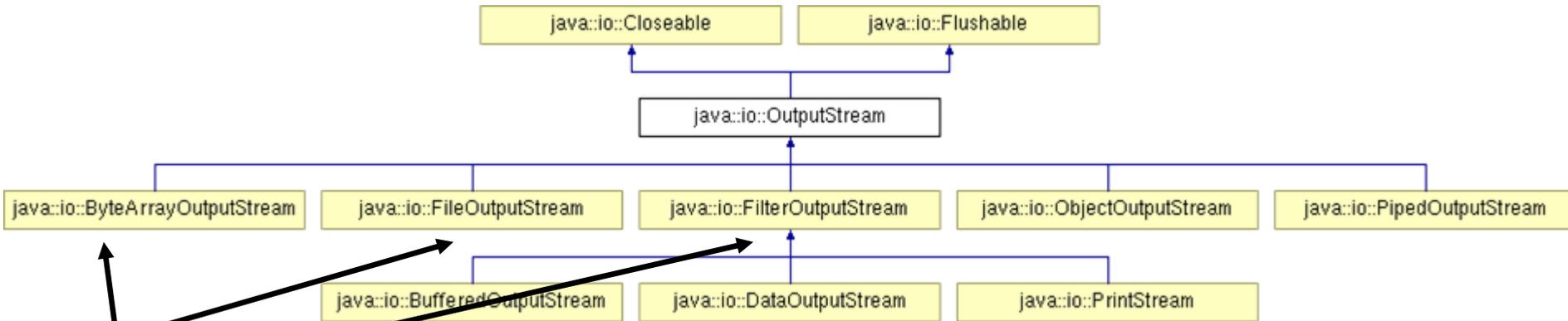
- Textdaten vs. binäre Daten
- Verschiedene Ausgabemedien

Was haben verschiedene Ausgaben
gemeinsam?

java.io.Flushable



java.io.OutputStream

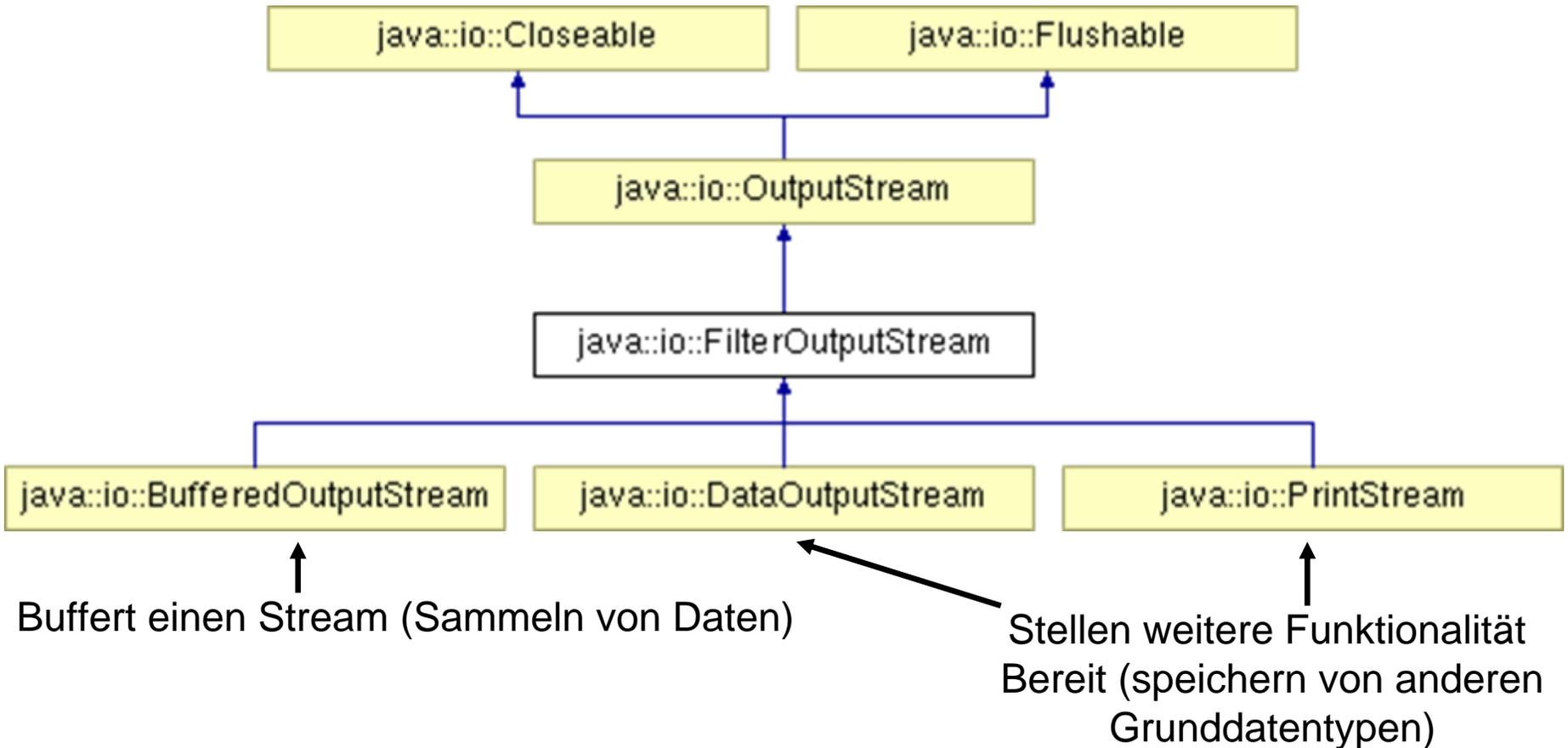


`ByteArrayOutputStream`: schreibt in ein Array von Bytes

`FileOutputStream`: schreibt in eine Datei

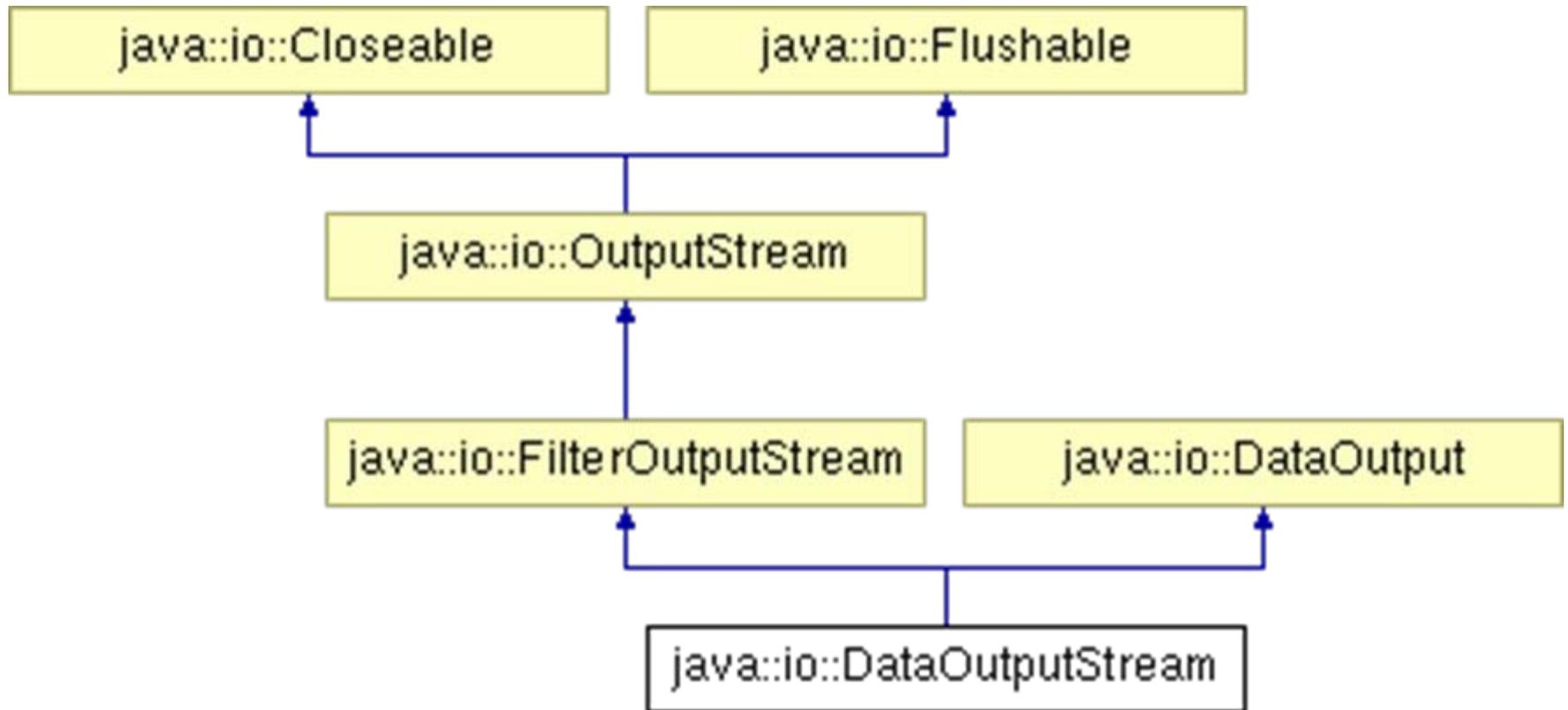
`FilterOutputStream`: schreibt in einen Stream

java.io.FilterOutputStream



Bekannteste Instanz: `System.out`

java.io.DataOutputStream



java.io.DataOutputStream

```
public static void test() throws IOException {
    DataOutputStream dos = new DataOutputStream(
        new BufferedOutputStream(
            System.out));
    dos.writeLong(Long.MAX_VALUE);
    dos.flush();
    dos.close();
}
```