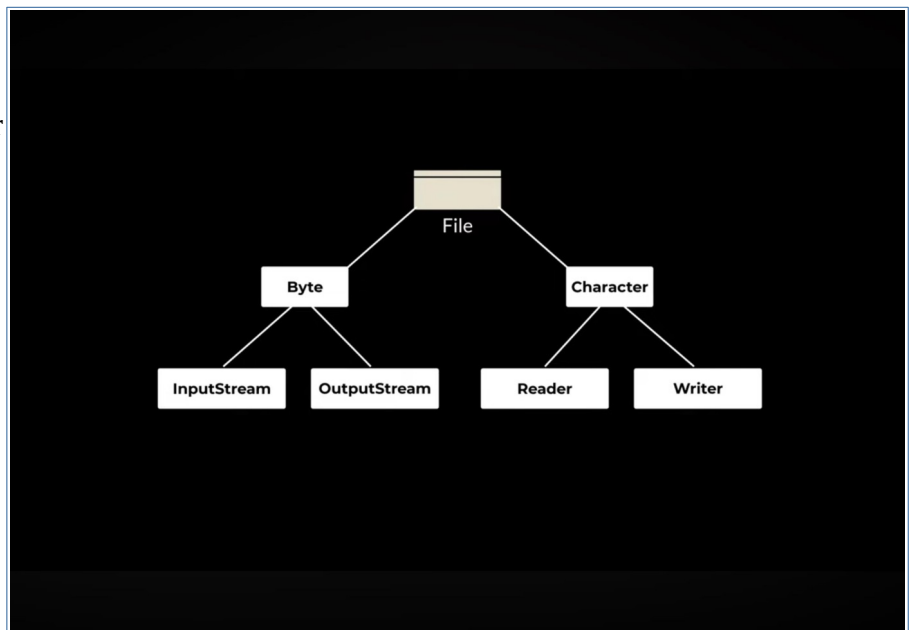


„File-Handling“

Dateien Lesen & Schreiben:

Also ich habe ein Programm und einen File. Auf meinem File habe ich Dateien, die ich mit Hilfe meiner Programm lesen kann oder ich habe auf meinem Programm Dateien, die ich auch mit Hilfe mein Programm auf diesem File schreiben möchte



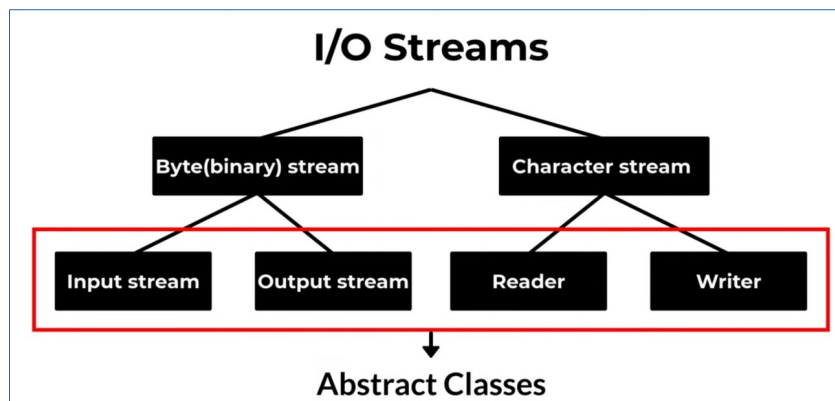
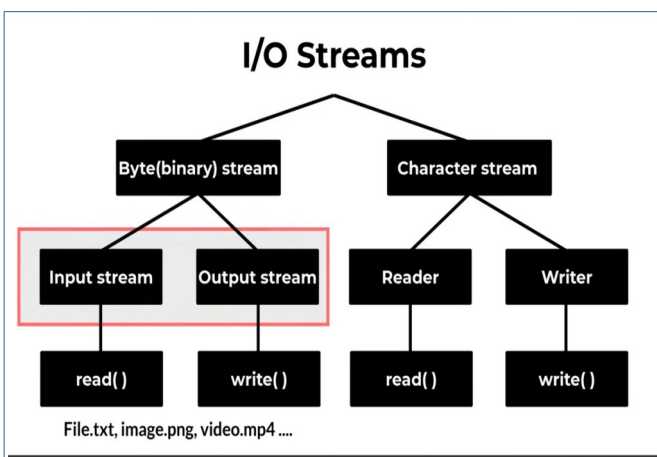
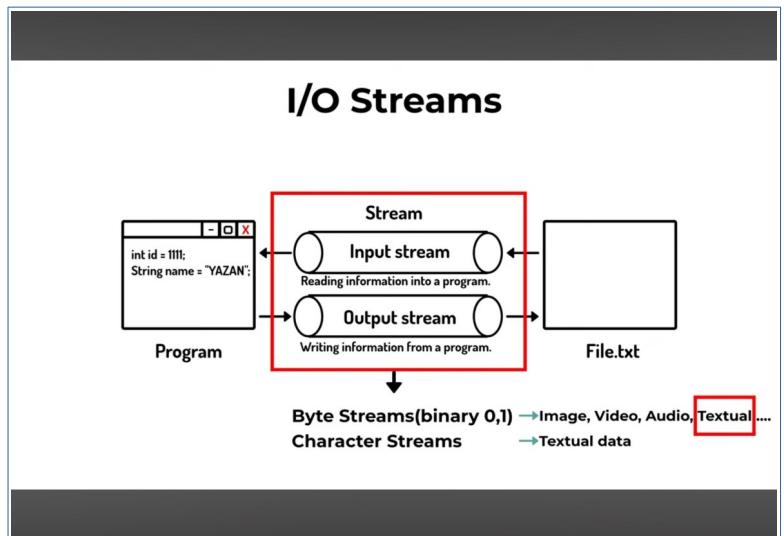
Was sind diese Dateien?

Diese Dateien sind entweder

Bytes (Bilder, Videos, Audio, Dokumente, Archive(ZIP), Datenbanken)
oder **Character** (ASCII, ISO, Unicode) also für Texte

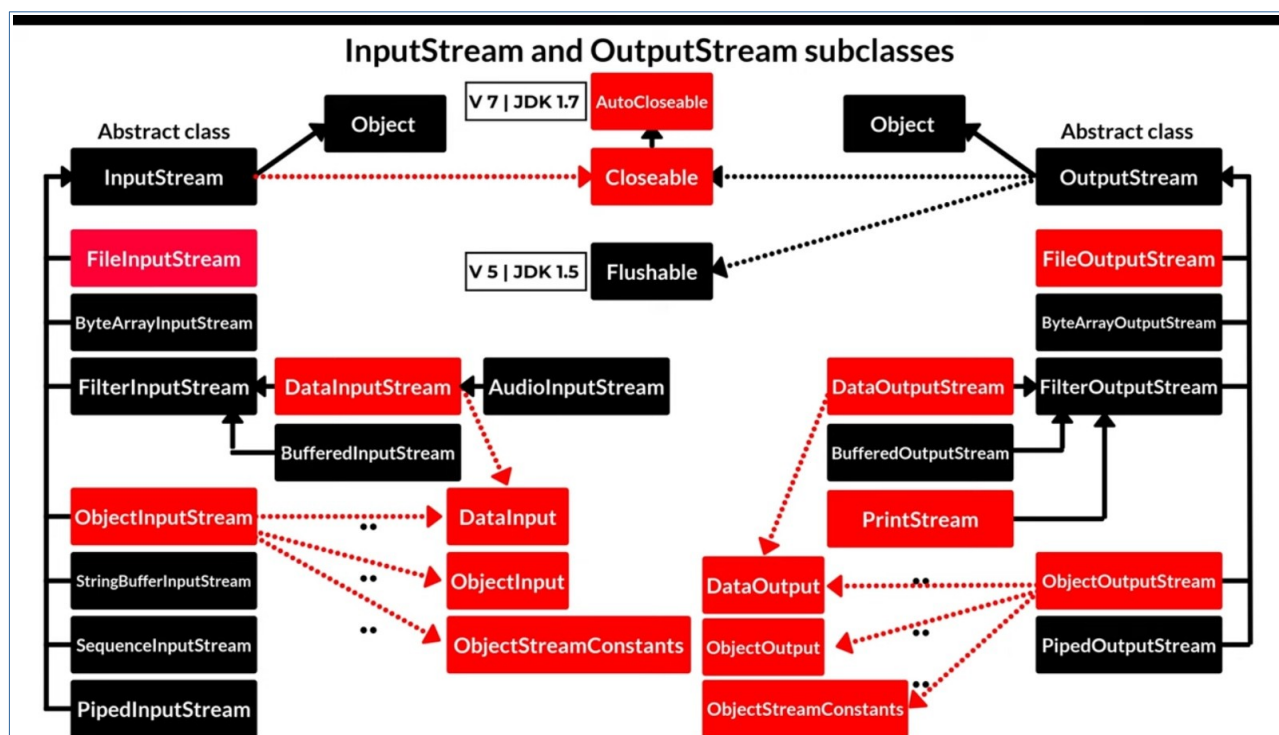
Bytes:

für Bytes habe ich die Zwei (abstrakte) Klassen **FileInputStream** für (lesen) und **FileOutputStream** für (schreiben) in diesen Zwei Klassen habe ich mehrere Methoden, die ich verwenden kann um lesen und schreiben.



Byte Klassen:

- hier sind alle Klassen, die mit Byte zu tun haben.
- Die **InputStream** und **OutputStream** sind die Parentklassen alle anderen Klassen!



Hie sind alle Klassen:

I/O Streams

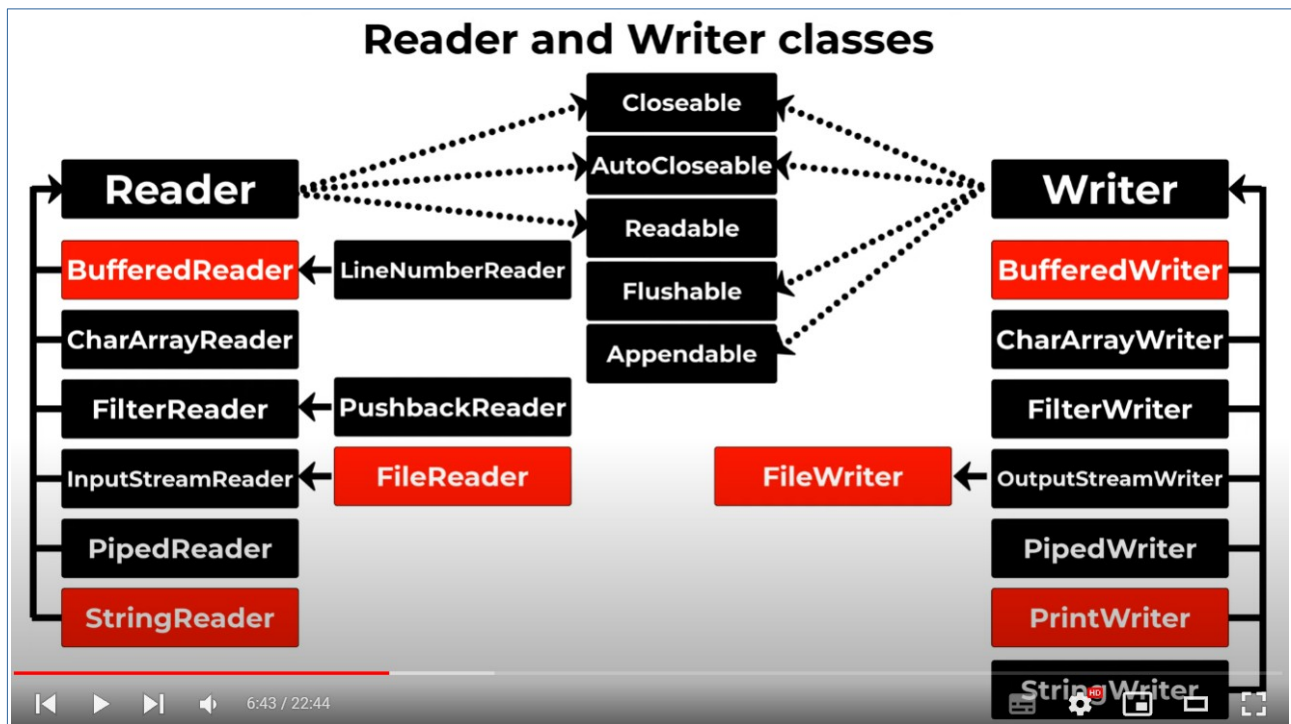
	Byte Based		Character Based	
	Input	Output	Input	Output
Basic	InputStream	OutputStream	Reader InputStreamReader	Writer OutputStreamWriter
Arrays	ByteArrayInputStream	ByteArrayOutputStream	CharArrayReader	CharArrayWriter
Files	FileInputStream RandomAccessFile	FileOutputStream RandomAccessFile	FileReader	FileWriter
Pipes	PipedInputStream	PipedOutputStream	PipedReader	PipedWriter
Buffering	BufferedInputStream	BufferedOutputStream	BufferedReader	BufferedWriter
Filtering	FilterInputStream	FilterOutputStream	FilterReader	FilterWriter
Parsing	PushbackInputStream StreamTokenizer		PushbackReader LineNumberReader	
Strings			StringReader	StringWriter
Data	DataInputStream	DataOutputStream		
Data - Formatted		PrintStream		PrintWriter
Objects	ObjectInputStream	ObjectOutputStream		
Utilities	SequenceInputStream			

Character:

für Character haben wir die zwei (abstrakte) Klassen **Reader** für (lesen) und **Writer** für schreiben

Merke: FileInputStream und FileOutputStream sind für File Texte nicht geeignet, alternativ Möglichkeit die abstrakte Klassen Reader und Writer

Reader & Writer Klassen:



* **Hinweis:** Die abstrakte Klasse Reader liest Zeichen für Zeichen und ruft die Methode read() jedes Mal auf, wenn ein neues Zeichen gelesen wird. * Wenn wir z.B. 10 Zeichen lesen möchten, wird die Methode read() 10 Mal aufgerufen. Alternative Möglichkeit: Die Klasse BufferedReader kann verwendet werden, um ganze Zeilen oder größere Blöcke von Zeichen auf einmal zu lesen, * was effizienter sein kann, da sie den Lesevorgang puffert und somit weniger oft auf die zugrunde liegende Datenquelle zugreift.