

Aufgabe: Zeitmessung un-/gepufferter Streams (ByteStreams, Exceptions)

Buffered Streams verwenden im Hintergrund einen Puffer. Ein Puffer ist ein Zwischenspeicher im Hauptspeicher.

Buffered Input Streams greifen nur auf das Betriebssystem zu, wenn der Puffer leer ist.

Buffered Output Streams greifen nur auf das Betriebssystem zu, wenn der Puffer voll ist.

Ein Buffered Stream lässt sich einfach „um“ einen bestehenden Stream legen, bspw.:

```
FileOutputStream fos = new FileOutputStream("test.mp3");  
BufferedOutputStream bos = new BufferedOutputStream(fos);
```

Ihre Aufgabe ist es, eine Musikdatei einzulesen und eine Kopie

- byteweise mit Hilfe eines ungepufferten Streams
- byteweise mit Hilfe eines gepufferten Streams
- mit einem **byte[]** der Größe 1024 mit Hilfe eines ungepufferten Streams

zu erstellen.

Der Nutzer soll den Pfad zu der Musikdatei, die kopiert werden soll, eingeben können. Stellen Sie in Ihrer Implementierung sicher, dass sie eine **FileNotFoundException** fangen und den aufgetretenen Fehler sinnvoll beheben kann.

Messen Sie für alle drei Fälle jeweils die Zeit!

Um welchen Faktor (bzw. um wie viel Prozent) unterscheidet sich der Zeitaufwand bei den drei Fällen?

Um eine genauere Aussage treffen zu können, führen Sie die Messungen zehnmal am Stück durch.

Überprüfen Sie, ob die kopierte Musikdatei nach dem Kopieren die gleiche Länge hat wie die Originaldatei.

Hinweis:

System.currentTimeMillis liefert die Anzahl der vergangenen Millisekunden seit 1. Januar, 1970 UTC (Coordinated Universal Time) zurück.

Mit folgendem Code kann die Dauer von Anweisungen gemessen werden.

```
long startTime = System.currentTimeMillis();  
//Code, dessen Dauer gemessen werden soll  
//...  
long endTime = System.currentTimeMillis();  
long duration = endTime - startTime;
```

Für kleinere Zeitauflösungen (Nanosekunden anstelle von Millisekunden) kann **System.nanoTime** anstelle von **System.currentTimeMillis** verwendet werden.