

Aufgabe: Streamverdopplung (Streams, Exceptions, Decorator)

Schreiben Sie eine Klasse **OutputStreamDoubler**, die sich für die Nutzer der Klasse genauso wie ein **OutputStream** verhalten soll. Die Ausgaben des Streams sollen jedoch an zwei weitere **OutputStream**-Objekte durchgereicht werden, die dem **OutputStreamDoubler** bei der Initialisierung mitgegeben werden.

- a) Implementieren Sie einen Konstruktor, welcher zwei **OutputStream**-Objekte entgegennimmt und für die weitere Verwendung speichert.
- b) Implementieren Sie die Methode

```
public void close() throws IOException
```

Die Methode soll die beiden vorgehaltenen **OutputStream**-Objekte schließen. Sollte beim Schließen eines Streams eine **IOException** auftreten, so soll dennoch der andere Stream geschlossen werden. Wenn mindestens eine **Exception** ausgelöst wurde, soll eine neue **IOException** geworfen werden, die angibt, welcher oder welche Streams, den Fehler ausgelöst haben.

- c) Implementieren Sie die Methode

```
public void write(int b) throws IOException
```

Jeder übergebene Wert soll dabei in die beiden vorgehaltenen **OutputStream**-Objekte geschrieben werden. Sollte beim Schreiben in einen der Streams eine **IOException** auftreten, so ist diese zu fangen, der Stack-Trace auf der Konsole/dem Terminal auszugeben, und mit der Verarbeitung fortzufahren. D. h. also: Probleme beim Schreiben in einen Stream verhindern nicht, dass Werte in den anderen Stream geschrieben werden.

Schreiben Sie eine **main**-Methode, in der zwei **FileOutputStream**-Objekte angelegt werden, mit deren Hilfe in die Dateien **file1.txt** und **file2.txt** geschrieben werden kann. Anschließend sollen die beiden **FileOutputStream**-Objekte mit Hilfe des **OutputStreamDoubler** zusammengefasst werden und eine beliebige Bytefolge in den Strom geschrieben werden. Schließen Sie danach alle beteiligten Streams.