

Aufgabe: Streamverdopplung (Streams, Exceptions, Decorator)

Schreiben Sie eine Klasse **OutputStreamDoubler**, die sich für die Nutzer der Klasse wie ein **OutputStream** verhalten soll. Die Ausgaben sollen jedoch auf zwei weitere **OutputStreams** 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 wird diese propagiert. Tritt die Exception beim Schließen des ersten Streams auf, bleibt der zweite Stream geöffnet.

- c) Implementieren Sie die Methode

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

Jeder übergebene Wert soll dabei in die beiden vorgehaltenen OutputStreams geschrieben werden. Sollte beim Schreiben in einem der Streams eine IOException auftreten, so ist diese zu fangen und mit der Verarbeitung fortzufahren, d. h. Probleme beim Schreiben in einen Stream verhindern nicht, dass Werte in den anderen Stream geschrieben werden. Ist beim Schreiben eines Wertes mindestens eine IOException aufgetreten, so soll nach den Schreibvorgängen in beide Streams eine neue IOException geworfen werden, andernfalls ist die Methode regulär zu beenden.

- d) Schreiben Sie eine main-Methode, in der die Implementierung des OutputStreamDoubler getestet werden soll.  
Dazu sollen zwei FileOutputStream-Objekte angelegt werden, mit deren Hilfe in die Dateien `file1` und `file2` geschrieben werden können. Anschließend sollen die beiden FileOutputStreams mit Hilfe des OutputStreamDoubler zusammengefasst werden und eine beliebige Bytefolge in den Strom geschrieben werden. Sorgen Sie dafür, dass alle beteiligten Streams geschlossen werden.