



Das File `hotels.db` enthält Informationen über Hotels<sup>1</sup>. Die Datei besitzt einen einzigartigen Aufbau. Es hilft, sich vorzustellen, mit einem binär gespeicherten Datenbanktable zu arbeiten.

- Die Datei beginnt mit einer ID(4 Byte),
- und danach einem Offset(4 Byte) `offset`.

## 1 Columns

Nach dem Offset folgt ein Block, welcher die Namen der Columns und die Column-Längen definiert. Er beginnt mit der Anzahl an Columns(2 Byte). Danach folgen für **jede** Column folgende Informationen:

- Die Länge des Column-Namens in Byte(2 Byte) `n`
- Der Column-Name(`n` Byte)
- Die Länge der Column(2 Byte)

Lies diese Informationen aus und stelle sicher, dass der entsprechende Test passed.

## 2 Hotels

Schreibe nun eine Klasse `Hotel`, welche Instanzvariablen ähnlich den gelesenen Columns beinhaltet. Das Byte `offset` ist das erste Byte, welches Hotelinformationen beinhaltet.

- Vor jedem Hotel stehen 2 Byte, welche entscheiden, ob das Hotel als gelöscht markiert wurde oder nicht. Ist der gelesene Wert `0x0000`, so ist das Hotel gültig, `0x8000` bedeutet gelöscht. Andere Werte sind mit einer `IllegalArgumentException` abzuhandeln.
- Danach kommen die eigentlichen Hotelinformationen; die Bytezahl pro Column wurde bereits ausgelesen und muss hier verwendet werden. Die Daten wurden allesamt zuerst in Strings verwandelt und dann binär geschrieben<sup>2</sup>, also in etwa durch Code wie `output.write(column.toString().getBytes());`
- Da Geldbeträge nicht in Fließkommadarstellung gespeichert werden sollten<sup>3</sup>, ist die `rate` in Cent zu speichern.

## 3 Methoden

- `public static Map<String, Short> readColumns(String filename) throws IOException`  
Liest die Namen der Columns aus und weist jeder die nötige Anzahl an Bytes zu. Die **Reihenfolge** der Columns muss dabei so bleiben, wie in der Datei spezifiziert, da sonst das Auslesen der Hotels unmöglich wird.
- `public static Set<Hotel> readHotels(String filename) throws IOException`  
Liefert die Hotels aus der Datei sortiert nach Ort und dann nach Name.

## 4 Test

Vervollständige die Testmethode `createsHotel`.

<sup>1</sup>Quelle: Java Programmer Certification

<sup>2</sup>Dies ist **nicht** die Standardvorgehensweise für binär gespeichert Daten.

<sup>3</sup>`0.1 + 0.1 + 0.1 != 0.3`