

Dateien lesen

FileReader und BufferedReader

Eine Möglichkeit, Dateien mit Java einzulesen, bietet der `FileReader` in Kombination mit dem `BufferedReader`. Hierzu müssen die entsprechenden Funktionen mit `import java.io.*` eingebunden werden.

```
1 FileReader fr = new FileReader("input.txt");
2 BufferedReader br = new BufferedReader(fr);
3 String zeile = br.readLine();
```

Listing 1: Dateien lesen

Zeile 1: `FileReader` anlegen, im Konstruktor gibt man dabei den Dateinamen der einzulesenden Datei an. Diese liegt im Hauptordner des Projektes!

Zeile 2: Der `BufferedReader` dient wie der `Scanner` dazu, Texte zu lesen.¹ Hiermit wird also der Inhalt der Datei gelesen.

Zeile 3: Anschließend können wir mit der Methode `readLine()` des `BufferedReader`-Objektes eine Zeile einlesen.

Kann keine Zeile mehr gelesen werden, so liefert `br.readLine()` `null` zurück. Dies kann z. B. in einer `while`-Schleife als Bedingung genutzt werden.

1. Aufgabe

Erstelle eine einfache Textdatei `input.txt` mit einem **mehrzeiligen** Inhalt.

Programmiere anschließend eine `Main`-Klasse mit `main`-Methode, welche die Datei komplett liest und auf der Konsole ausgibt.

Hinweis: Es werden zwei Fehler `Unhandled exception type FileNotFoundException` und `Unhandled exception type IOException`. Löse diese beiden Fehler indem du auf `Add throws declaration` klickst.

2. Aufgabe

Erweitere das Programm so, dass zunächst alle Zeilen in einen einzigen `String` gespeichert werden, dass du mit nur einem Aufruf von `System.out.println()` die komplette Datei auf der Konsole ausgeben lassen kannst.

Was fällt dir dabei auf?

¹Dateien könnte man auch mit dem `Scanner` lesen, ein `BufferedReader` arbeitet aber etwas effizienter.

static

Oft braucht man Methoden, die nicht an ein spezielles Objekt gekoppelt sind. Diese haben wir bisher als **static**-Methoden direkt in die Hauptklasse unter die **main**-Methode programmiert.

Zu besserer Strukturierung können wir diese ebenfalls in Klassen auslagern um das Programm übersichtlicher zu gestalten. Eine solche Klasse ist in der Hinsicht keine Vorlage für ein Objekt sondern dient lediglich als Hilfsklasse.

Um eine Methode aus dieser Klasse aufzurufen benötigt man auch keine Instanz! Man kann diese **static**-Methoden direkt über den Klassennamen aufrufen:

```
1 class Beispiel {
2     public static void Methode() {
3         [...]
4     }
5 }
```

Listing 2: Klasse Beispiel

```
1 class Main {
2     public static void main(String[] args) {
3         Beispiel.Methode();
4     }
5 }
```

Listing 3: Klasse Main

3. Aufgabe

Programmiere eine Klasse **Reader** mit einer **static**-Methode **readFile(String filename)**. Diese soll als Parameter den Dateinamen bekommen und den Inhalt der Datei als String zurückgeben.

Rufe dann aus der **main**-Methode die Methode auf:

```
1 public static void main(String[] args) {
2     String datei = Reader.readFile("input.txt");
3     System.out.println(datei);
4 }
```

Listing 4: Klasse Main

Hinweis: Löse die Fehler der nichtbehandelten Exceptions jeweils mit **Add throws declaration**.