

JUnit Exkurs: Lesen aus Dateien beim Testen mit JUnit

Es kommt häufig vor, dass beim Testen Testdaten von Dateien eingelesen werden müssen. Der Umgang mit Dateien beim Testen ist aber eher umständlich und unhandlich. Wird ein automatischer Testdurchlauf gestartet, so kann es vorkommen dass viele Dateien mit Testdaten geöffnet werden. Immer wieder müssen Dateien gelöscht, angepasst und neu angelegt werden. Deshalb wird im Folgenden eine Möglichkeit vorgestellt, anhand welcher in einem Test vollautomatisch aus einem String eine Datei aufgebaut wird und Methoden, die mit Dateien arbeiten sollen, mit diesem zu einer Datei konvertierten String arbeiten können.

Grundlage der Überlegungen ist die Java-Klasse `BufferedReader`. Anhand der Aufrufe

```
String fileString = "Sepp;MALE;null;null\nElsa;FEMALE;null;null\nKurt;MALE;Elsa;Sepp";
BufferedReader reader = new BufferedReader(new StringReader(fileString));
```

bzw.

```
String fileName = "dateiname.txt";
FileInputStream in = new FileInputStream(fileName);
BufferedReader reader = new BufferedReader(new InputStreamReader(in));
```

wird ein Dateiobjekt einerseits aus einem String generiert und andererseits mit tatsächlich vorhandenen Textdateiinhalten gefüllt.

Mit dem Aufruf

```
String line = reader.readLine();
```

kann aus dem so generierten Dateiobjekt zeilenweise gelesen werden.

Die Klasse `BufferedReader` stellt zudem die Methoden `mark()` und `reset()` zur Verfügung. Damit kann eine Stelle in der Datei markiert werden, damit später an diese Stelle zurück gesprungen werden kann.

Wenn also beim Anlegen eines Objektes dieses mit den Inhalten einer Datei bzw. mit einer aus einem String generierten Datei gefüllt werden soll, kann dies folgendermaßen bewerkstelligt werden:

```
public class PersonList extends ArrayList<Person>
{
    public PersonList(BufferedReader reader) throws IOException {
        readPersons(reader);
    }

    public PersonList(String filename) throws FileNotFoundException, IOException {
        FileInputStream in = new FileInputStream(filename);
        BufferedReader reader = new BufferedReader(new InputStreamReader(in));
        readPersons(reader);
    }

    private void readPersons(BufferedReader reader) throws IOException {
        ...
    }
}
```

In der Testmethode kann das Objekt instanziiert und mit einem zu einer Datei konvertierten String folgendermaßen gefüllt werden:

```
String fileString = "Sepp;MALE;null;null\nElsa;FEMALE;null;null\nKurt;MALE;Elsa;Sepp";
BufferedReader reader = new BufferedReader(new StringReader(fileString));
PersonList list = new PersonList(reader);
```