

# Aufgabenblatt 7

## Allgemeines zum Lösen von Aufgabenblättern

Verwenden Sie Ihr existierendes IntelliJ-IDEA-Projekt und lösen Sie darin die Aufgaben. Eine gute Lösung erfüllt die Vorgaben der Aufgabenbeschreibung, ist kurz und einfach gehalten, ist mit informativen Kommentaren versehen und wurde gut getestet. Objektvariablen sind `private`.

Das Projekt mit den gelösten Aufgaben muss rechtzeitig vor der Deadline als ZIP-Datei in TUWEL hochgeladen werden. Programmtext für das Testen soll im Projekt enthalten sein. Bei mehrfachem Hochladen zählt die zuletzt hochgeladene ZIP-Datei. Es gibt keine andere Möglichkeit zur Abgabe.

Jede gelöste Aufgabe muss in TUWEL angekreuzt werden. Lösungen angekreuzter Aufgaben müssen in der Übungseinheit präsentiert werden können. Nach der Deadline ist das Ändern der Kreuzchen nicht möglich.

## CSV-Dateien

Im Allgemeinen enthalten CSV-Dateien eine beliebige Anzahl von Datensätzen (Records). Jeder Datensatz enthält eine beliebige Anzahl von Feldern (Fields). Datensätze werden durch einen Record Separator getrennt, in unserem Fall durch ein Newline. Felder werden durch einen Field Separator getrennt, in unserem Fall durch einen Strichpunkt `;`. Für die ersten drei Aufgaben dürfen Felder keine Newlines und `;` enthalten.

Für die Ein- und Ausgabe von CSV-Dateien in den folgenden Aufgaben dürfen Sie Standard-Java-Libraries verwenden, aber keine Libraries von anderen Quellen.

## Aufgabe 1 (20%, 1 Punkt)

Erweitern Sie die Klasse `Playlist1` um eine nicht-statische Methode

```
void save(String filename)
```

die eine CSV-Datei mit dem Namen `filename` anlegt und die `Playlist1` dort abspeichert, und zwar jeden Song als Record (in derselben Reihenfolge wie in der `Playlist1`), wobei ein Song folgende Felder (in dieser Reihenfolge) enthält: `titel`, `band`, und `laenge` (in Dezimaldarstellung ohne angehängtes `s`).

In dieser Aufgabe muss Ihr Programm noch nicht mit Sonderzeichen in den Feldern umgehen können.

## Aufgabe 2 (verpflichtend, 30%, 1.5 Punkte) unbedingt lösen

Erweitern Sie die Klasse `Playlist1` um einen Konstruktor

```
Playlist1(String filename)
```

Dieser Konstruktor öffnet die CSV-Datei mit dem Namen `filename`, und liest daraus eine `Playlist1`.

In dieser Aufgabe muss Ihr Programm noch nicht mit von `"` umschlossenen Feldern umgehen können.

Einführung in die  
Programmierung 2

LVA-Nr. 185.A92  
2018 S  
TU Wien

**Thema:**

Ein- und Ausgabe mit  
Validierung,  
Ausnahmebehandlung

**Ausgabe:**

4. 6. 2018

**Abgabe (Deadline):**

11. 6. 2018, 6:00 Uhr

Lösungen hochladen und  
gelöste Aufgaben  
ankreuzen (TUWEL)

**Skriptum:**

Seiten 121–134  
Aufgaben 4.1–4.16

### Aufgabe 3 (verpflichtend, 20%, 1 Punkte) unbedingt lösen

Baut auf Aufgabe 2 auf

Neben den üblichen Dateifehlern (z.B. "nicht gefunden") kann auch der Inhalt der Datei nicht dem erwarteten Format entsprechen. Welche Formatfehler können in Aufgabe 2 auftreten? Definieren Sie Exceptions für diese Formatfehler, mit `IOException` als Oberklasse. Fügen Sie entsprechende `throw`-Anweisungen ein.

Fügen Sie im Konstruktor `Playlist1(String filename)` eine Ausnahmebehandlung ein, die diese Formatfehler abfängt, und ggf. eine entsprechende Meldung ausgibt. Datensätze mit Formatfehlern sollen nicht als Songs in die `Playlist1` aufgenommen werden, aber die Verarbeitung der Datei soll nicht abgebrochen, sondern mit dem nächsten Datensatz fortgesetzt werden.

### Aufgabe 4 (30%, 1.5 Punkte)

Erweiterung der CSV-Dateien: Jedes Feld kann durch doppelte Hochkommas " umschlossen sein (also ein " am Anfang und eines am Ende), dann können die Nutzdaten auch Newlines und Strichpunkte enthalten, und auch ein " am Anfang. Wenn die Nutzdaten so eines Feldes " enthalten, steht in der CSV-Datei anstelle jedes " der Nutzdaten "" (z.B. für die Nutzdaten "x" steht im Feld ""x""). Wenn die Nutzdaten eines Feldes ein " an einer anderen Stelle als am Anfang haben, muss es nicht durch " umschlossen werden; das " wird in diesem Fall auch nicht verdoppelt. Beispiele:

```
abc;d"e;fgh
"abc";"d""e";"fgh"
abc;"d"e";fgh
```

sind zwei korrekte Datensätze mit je drei Feldern, wobei die Felder den gleichen Inhalt haben, gefolgt von einem fehlerhaften Datensatz: das mittlere Feld enthält einen CSV-Fehler (nicht-verdoppeltes ", das nicht von einem Separator gefolgt wird).

```
abc;"d""e;fgh
abc";d"e;fgh
```

ist *ein* Datensatz mit vier Feldern, wobei das zweite Feld folgenden String enthält, also sowohl ein Newline als auch einen Strichpunkt:

```
d"e;fgh
abc
```

Erweitern Sie die Klasse `Playlist1` so, dass auch Felder mit Newlines, Strichpunkt, und " am Anfang geschrieben und gelesen werden können. Neue Formatfehler sollen durch neue Exceptions repräsentiert und wie oben behandelt werden. Dabei sollen auch Dateien eingelesen werden können, die von anderen Programmen geschrieben wurden, nicht nur die, die Ihr eigenes `save` erzeugt hat.