

Aufgabe 7 (IO mit Dateien)

Hinweis zu Bonuspunkten: Zur Vergabe der Bonuspunkte werden wir in regelmäßigen Abständen den von Ihnen produzierten Quellcode begutachten. Informationen darüber, wann wir welche Praktikumsaufgaben begutachten, wie viele Bonuspunkte es gibt und wie Sie Ihren Quellcodestand einreichen, werden rechtzeitig bekanntgegeben.

Hinweis zum entstehenden Code: Verwenden Sie für den entstehenden Code das in Praktikum 1 in GitLab angelegte Projekt.

1. CSV-Export

Erweitern Sie die "Foto-App" um eine neue Funktion, die einen Export aller vorhandenen Alben mitsamt zugehöriger Foto-Daten als CSV-Datei ermöglicht. CSV (**Comma-Separated Values**) ist ein [simples Dateiformat](#) zum strukturierten Speichern von Daten. Bei diesem Format gilt:

- Zur Trennung von Datensätzen wird ein Zeilenumbruch verwendet.
- Zur Trennung von einzelnen Datenfeldern eines Datensatzes wird ein Komma (,) verwendet.

Daten in diesem Format lassen sich z.B. leicht in gängige Tabellenkalkulationsprogramme wie z.B. Microsoft Excel importieren.

Realisieren Sie die neue Funktion folgendermaßen:

1. Erstellen Sie eine neue Schnittstelle `CsvExportable`. Die Schnittstelle soll eine Funktion `exportiereAlsCsv` deklarieren, welche eine Zeichenkette im CSV-Format als `String` zurückliefert.
2. Implementieren Sie die Schnittstelle `CsvExportable` in den Klassen `Fachobjekt`, `Album`, `Foto` und `FotoMetadaten`. Auf diese Weise sind beliebige Objekte dieser Klassen in das CSV-Format exportierbar. Vermeiden Sie bei der Implementierung doppelten Code.
3. Ergänzen Sie die `FotoVerwaltung` um eine neue Methode `void exportiereEintraegeAlsCsv(File datei)`. Diese Methode schreibt alle in der `FotoVerwaltung` enthaltenen Einträge im CSV-Format in die gegebene Datei `datei`. Verwenden Sie zur Implementierung die in 1. und 2. vorbereitete Schnittstelle `CsvExportable` und die aus der Vorlesung bekannte Klasse `java.nio.file.Files`. In der Methode ggf. auftretende Exceptions sollen nicht in der Methode behandelt, sondern stattdessen an den Aufrufer weitergegeben werden.

Beispiel einer Ausgabe der Methode:

```
Album-ID,Name,Besitzer,Foto-ID,Name,Dateiname,Breite,Höhe,Kameramarke,Kameramodell,Erstellungsdatum
d71ec58c,Menschen,Peter,22718147,Meer,images/16524319597_c735c777a8_o.jpg, 1024, 1024,
    ↳ NIKON CORPORATION, NIKON D750, 08.07.1989 11:25:00
d71ec58c,Menschen, Peter,7e489f14,Artisten,images/35835723323_b3ed4bf5d1_o.jpg, 1024, 1024,
    ↳ NIKON CORPORATION, NIKON D750, 08.07.1989 11:25:00
9199ff47,Naturbilder,Petra,d3a97fd1,Wiese,images/7861351302_74a45956dd_o.jpg, 1024, 1024,
    ↳ NIKON CORPORATION, NIKON D750, 08.07.1989 11:25:00
```

Die ↳ -Zeichen im obigen Beispiel sind *keine* Umbrüche in der CSV-Ausgabe, sondern dienen lediglich der verbesserten Darstellung auf diesem Aufgabenzettel. Ein Datensatz entspricht einer Zeile und endet jeweils nach dem Erstellungsdatum (im Beispiel 08.07.1989 11:25:00).

Hinweis: Falls Sie eine Software zur Tabellenkalulation verfügbar haben (z.B. Microsoft Excel, LibreOffice Calc, Numbers) können Sie die resultierende CSV-Datei dort testweise importieren.

4. Ergänzen Sie das Menü der Anwendung um einen neuen Menüpunkt für den CSV-Export:

```
Foto-App

1. Album hinzufügen
2. Drucke alle Alben
3. Drucke Album mit Name
4. CSV-Export
5. Beenden

Bitte Aktion wählen:
```

5. Bei Auswahl des neuen Menüpunktes sollen BenutzerInnen zur Eingabe eines Dateinamens aufgefordert werden. Verwenden Sie für diesen Zweck wieder ein JOptionPane .
6. Berücksichtigen Sie bei der Implementierung folgende Fehlersituationen bei der Eingabe des Dateinamens:
1. Die BenutzerIn hat einen leeren Dateinamen eingegeben. In diesem Fall wird die BenutzerIn über einen Hinweisdialog (*message dialog*) darüber informiert und muss nach Bestätigung des Hinweises die Eingabe erneut vornehmen.
 2. Der von der BenutzerIn eingegebene Dateiname bezieht sich auf eine Datei, die bereits existiert. In einer solchen Situation soll die BenutzerIn über eine JOptionPane gefragt werden, ob die existierende Datei wirklich überschrieben werden soll. Ermitteln Sie mit Hilfe der [API-](#)

[Dokumentation zu JOptionPane](#) eine geeignete Möglichkeit, einen solchen Bestätigungsdialog (*confirm dialog*) anzuzeigen. Antwortet die BenutzerIn mit "Ja", wird mit dem Export fortgefahren. Antwortet die BenutzerIn mit "Nein", so landet sie wieder im Dialog zur Eingabe eines Dateinamens.

3. Wird in der Methode `exportiereEintraegeAlsCsv` in der `FotoVerwaltung` eine Exception geworfen, so soll der BenutzerIn eine Fehlermeldung über einen Hinweisdialog angezeigt und der Export-Vorgang abgebrochen werden. Die BenutzerIn landet nach Bestätigung des Hinweisdialoges wieder im Menü.

2. Commit und Push

1. Schreiben Sie den entstandenden Code per Commit in Ihrem lokalen Repository fest. Verwenden Sie als Commit-Message "Aufgabe 7: IO mit Dateien".
2. Bringen Sie die Änderungen dann per Push auf den GitLab-Server. Kontrollieren Sie in [GitLab](#), dass Ihre Änderungen angekommen sind.