Praktikum zum Programmierkurs (WI), Wintersemester 2021/2022

Aufgabe 2 (Erste Basisfunktionen)

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.*

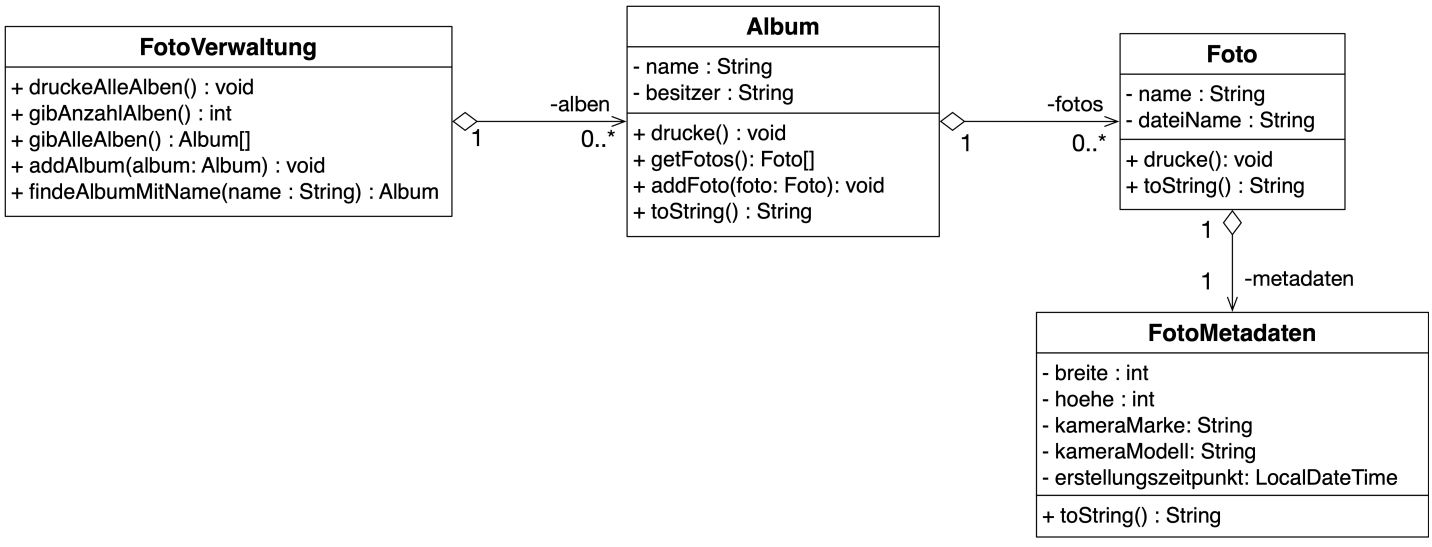
Fachliches Beispiel für alle Praktika: Foto-App

Unser fachliches Beispiel für alle folgenden Praktika ist eine App zur Verwaltung von Fotoalben. Die Anwendung wird fähig sein, Fotos und deren Metadaten in Form von Alben zu organisieren, in einer grafischen Oberfläche darzustellen, dateibasiert zu persistieren und in einem standardisierten Format zu exportieren.

Mit jedem Praktikum werden wir diese Anwendung weiterentwickeln, jeweils passend zu den aktuellen Inhalten der Vorlesung.

1. Fachklassen implementieren

Implementieren Sie die Klassen `Album`, `Foto`, `FotoMetadaten` und `Fotoverwaltung` gemäß folgendem Klassendiagramm:



Bitte berücksichtigen Sie bei der Implementierung die folgenden Punkte:

1. Legen Sie für den entstehenden Code ein neues Package mit Namen `pk.foto` an.

2. Fotos werden über die Klasse `Foto` modelliert. Die Metadaten eines Fotos sind in eine eigene Klasse `FotoMetadaten` ausgelagert. Realisieren Sie beide Klassen mit den im Klassendiagramm dargestellten Attributen. Beachten Sie insbesondere die dargestellte Assoziation zwischen den Klassen `Foto` und `FotoMetadaten`. Nutzen Sie für die Implementierung des Attributes `erstellungszeitpunkt` die Klasse `java.time.LocalDateTime` aus der Java-Standardklassenbibliothek. Recherchieren Sie mit Hilfe der Java 11 API-Dokumentation, wie Sie diese Klasse verwenden können.

3. Implementieren Sie folgende Methoden in der Klasse `Foto:`

    1. `drucke` : Gibt die Daten des Fotos auf der Konsole aus. Die Ausgabe soll in folgender Form erfolgen (`$attribut` bezeichnet den Wert des Attributes `attribut`):

        ```
        Fotoname: $name
        Dateiname: $dateiName
        Größe: $metadaten.$breite x $metadaten.$hoehe
        Kamera: $metadaten.$kameraMarke $metadaten.$kameraModell
        Erstellungsdatum: $metadaten.$erstellungszeitpunkt
        ```

        Beispiel:

        ```
        Fotoname: Meer
        Dateiname: images/16524319597_c735c777a8_o.jpg
        Größe: 1024 x 1024
        Kamera: NIKON CORPORATION NIKON D750
        Erstellungsdatum: 08.07.1989 11:25:00
        ```

        Geben Sie den Geburtstag, wie im obigen Beispiel gezeigt, im Format `TT.MM.JJJJ HH:mm:ss` aus. Nutzen Sie dazu die Klasse `java.time.format.DateTimeFormatter` aus der Java-Standardklassenbibliothek (Informationen zur Benutzung der Klasse finden Sie in der Java 11 API-Dokumentation).

    2. `toString` : Diese Methode überschreibt die gleichnamige Methode der Klasse Object und gibt eine lesbare String-Repräsentation eines `Foto` -Objektes zurück. Implementieren Sie diese Methode so, dass in der String-Repräsentation alle Attribute eines `Foto` -Objektes enthalten sind.

4. Realisieren Sie analog zur Klasse `Foto` auch eine `toString` -Methode in der Klasse `FotoMetadaten` .

5. Fotoalben werden über die Klasse `Album` realisiert. Implementieren Sie die Klasse mit den im

Klassendiagramm dargestellten Attributen. Beachten Sie auch die dargestellte Assoziation zwischen `Album` und `Foto`: Realisieren Sie die Verwaltung der Fotos mit Hilfe eines Arrays. Beim Erzeugen einer `Album`-Instanz soll die Kapazität dieses Arrays zunächst auf *zwei Elemente* festgelegt werden.

6. Implementieren Sie folgende Methoden in der Klasse `Album:`

    1. `drucke`: Gibt das Album *mitsamt der enthaltenen Fotodaten* auf der Konsole aus. Die Ausgabe soll in folgender Form erfolgen (`$attribut` bezeichnet den Wert des Attributes `attribut`):

        ```
        Name: $name
        Besitzer: $telefon
        // fuer jedes Foto X in $fotos:
        === Foto X ===
            // Ausgabe von foto.drucke()
        ```

        Beispiel:

        ```
        Name: Naturbilder
        Besitzer: Sonja
        === Foto 1 ===
            Fotoname: Wiese
            Dateiname: images/7861351302_74a45956dd_o.jpg
            Größe: 1024 x 1024
            Kamera: NIKON CORPORATION NIKON D750
            Erstellungsdatum: 08.07.1989 11:25:00
        === Foto 2 ===
            Fotoname: Wald
            Dateiname: images/9207173484_852e8d983a_o.jpg
            Größe: 1024 x 1024
            Kamera: NIKON CORPORATION NIKON D750
            Erstellungsdatum: 08.07.1989 11:25:00
        ```

        Vermeiden Sie redundanten Code!

        Hinweis: Eine Einrückung können Sie mit Hilfe der Zeichenkette `\t` erzielen.

    2. `toString`: Gibt eine lesbare String-Repräsentation eines `Album`-Objektes zurück. Realisieren Sie diese Methode analog zur Klasse `Foto`.

    3. `getFotos`: Gibt alle im Album enthaltenen Fotos zurück.

    4. `addFoto`: Fügt ein neues Foto zu dem Album hinzu. Das neue Foto wird dabei als Parameter übergeben. Falls das Array bereits voll ist, soll ein neues, größeres Array erzeugt werden. Alle bisherigen Fotos sowie das neue Foto sollen zu diesem neuen Array hinzugefügt werden.

Schließlich soll das alte Array durch das neue Array ersetzt werden. Nutzen Sie zur Implementierung dieser Funktionalität die statische Hilfsmethode `copyOf` der Klasse `Arrays` aus der Standard-Klassenbibliothek.

7. Implementieren Sie die Klasse `FotoVerwaltung`. Beachten Sie die Assoziation zwischen `FotoVerwaltung` und `Album`. Realisieren Sie die Verwaltung der Alben mit Hilfe eines Arrays. Auch hier soll die initiale Kapazität dieses Arrays zunächst auf *zwei Elemente* festgelegt werden.

8. Implementieren Sie die folgenden Methoden in der Klasse `FotoVerwaltung`:

    1. `druckeAlleAlben`: Gibt alle verwalteten Alben mitsamt der zugehörigen Fotodaten auf der Konsole aus.

        Beispielausgabe:

        ```
        === Album 1 ===
        Name: Naturbilder
        Besitzer: Sonja
        === Foto 1 ===
            Fotoname: Wiese
            Dateiname: images/7861351302_74a45956dd_o.jpg
            Größe: 1024 x 1024
            Kamera: NIKON CORPORATION NIKON D750
            Erstellungsdatum: 08.07.1989 11:25:00
        === Foto 2 ===
            Fotoname: Wald
            Dateiname: images/9207173484_852e8d983a_o.jpg
            Größe: 1024 x 1024
            Kamera: NIKON CORPORATION NIKON D750
            Erstellungsdatum: 08.07.1989 11:25:00
            [...]
        === Album 2 ===
        Name: Menschen
        Besitzer: Aurel
        === Foto 1 ===
            Fotoname: Meer
            Dateiname: images/16524319597_c735c777a8_o.jpg
            Größe: 1024 x 1024
            Kamera: NIKON CORPORATION NIKON D750
            Erstellungsdatum: 08.07.1989 11:25:00
            [...]
        ```

    2. `gibAnzahlAlben`: Gibt zurück, wie viele Alben aktuell in `FotoVerwaltung` enthalten sind.

    3. `gibAlleAlben`: Gibt alle Alben zurück, die aktuell in `FotoVerwaltung` enthalten sind.

    4. `addAlbum`: Fügt ein neues Album zu `FotoVerwaltung` hinzu. Das neue Album wird dabei

als Parameter übergeben. Falls das Array voll ist, gehen Sie analog zur Methode `addFoto` (siehe 6.4) vor.

5. `findeAlbumMitName` : Gibt das Album zurück, bei welchem der Wert des Attributes `name` mit dem gegebenen Argument `name` übereinstimmt. Wurde kein passendes Album gefunden oder ist das gegebene Argument `null` , so gibt die Methode `null` zurück.

Hinweis: Das Klassendiagramm ist unvollständig. Fügen Sie bei Bedarf weitere Attribute und Methoden hinzu, z.B. private Hilfsmethoden oder solche Methoden, die Sie z.B. zum Zugriff auf den Zustand ihrer Objekte benötigen.

2. Java-Anwendung

Implementieren Sie (ebenfalls im Package `pk.foto`) eine ausführbare Java-Anwendung mit einer weiteren Klasse `Tester` . Beim Ausführen der Java-Anwendung soll Folgendes geschehen:

1. Es soll eine Instanz der Klasse `FotoVerwaltung` erzeugt werden.
2. Es sollen mindestens zwei Alben erzeugt und der `FotoVerwaltung` hinzugefügt werden. Jedes Album soll mindestens zwei Fotos enthalten.
3. Rufen Sie *alle weiteren Methoden* der Klasse `FotoVerwaltung` auf und geben Sie das Ergebnis jeweils auf der Konsole aus.

3. Commit und Push

1. Schreiben Sie den entstandenden Code per Commit in Ihrem lokalen Repository fest. Verwenden Sie als Commit-Message "Aufgabe 2: Erste Basisfunktionen".
2. Bringen Sie die Änderungen dann per Push auf den GitLab-Server. Kontrollieren Sie in GitLab, dass Ihre Änderungen angekommen sind.