DIE ZWEI ??

Beschreibung und Dokumentation

Christopher Berg und David Wenkemann Programmieren 3 - Projekt

INHALTSVERZEICHNIS

BESCHREIBUNG DES PRODUKTES

Es wird eine Art Social-Media-Plattform für Fotografen entwickelt. User, welche sich angemeldet haben, haben die Möglichkeit neue Bilder hochzuladen. Diese werden dann von anderen Nutzern Kategorien, um die Suche nach bestimmten Bildern zu erleichtern. Je nach Gefallen werden die Bilder auch von den Nutzern bewertet und nach Möglichkeit auch kommentiert. Um die Nutzer anzuspornen gibt es ein Rankingsystem in dem man mit Qualität und Beliebtheit der Bilder Erfahrung sammeln kann.

KLASSEN

Das Programm ist in zwei Klassenarten aufgeteilt. Zum einen gibt es die Datenklassen, in denen die ganzen Informationen gespeichert werden. Zum anderen gibt es Controllerklassen, welche Zugriff auf die Datenklassen haben und mit diesen arbeiten. In ihnen ist auch die ganze Logik verarbeitet. Um Sachen zu umgehen, die explizit in diesem Projekt ausgeschlossen sind, haben wir uns dafür entschieden mit doppelt verketteten Listen zu arbeiten. Diese bieten alles, was wir in diesem Projekt bisher gebraucht hatten. Hier fungieren die Controllerklassen als „Listen“ und die Datenklassen als Elemente der Listen.

USER

Beschreibung:

> Die Klasse „User“ beschreibt einen Account in diesem System. Wenn sich später ein neuer Nutzer registriert wird eine Instanz dieser Klasse erstellt und speichert seine Informationen. Um Bilder hochzuladen zu kommentieren oder zu bewerten ist es zwingend notwendig angemeldet zu sein!

Variablen:

Zu jeder Variable wurden die dazu passenden Get und Set Methoden implementiert.

Es besteht die Variable startXP, welche allen Usern eine gewisse Anzahl an Starterfahrungspunkten mit auf den Weg gibt.

- private static int startXP = 100;

Die folgenden Variablen sind Datenvariablen die rein für die Informationsspeicherung genutzt werden.

- private int user_ID;
- private String user_email;
- private String user_name;
- private String user_password;
- private int user_xp;
- private int user_uploads;

Um in der doppelt verketteten Liste mitwirken zu können muss die Instanz auch noch ihr nächstes beziehungsweise vorheriges Element kennen. In die Liste wird es direkt nach dem erzeugen vom User-Controller eingefügt.

- private user next;
- private user before;

Methoden:

Da es sich eigentlich um eine reine Datenklasse handelt, werden keine Methoden benötigt. Trotzdem wurde eine Methode entwickelt, die die Erfahrungspunkte des Nutzers erhöht, um Programmstrukturen an anderen Stellen zu erleichtern.

- public void getXP()

PICTURE

Beschreibung:

Die Klasse „Picture“ beschreibt ein, von einem User hochgeladenes, Bild. Dieses ist Teil der Liste, die der Picture-Controller erzeugt. Wie auch bei der Klasse User werden hier hauptsächlich Informationen gespeichert. Bilder können von Usern kommentiert werden, werden in eine Kategorie eingeordnet oder mit einem anderen Bild vergleichen.

Variablen:

Zu jeder Variable wurden die dazu passenden Get und Set Methoden implementiert.

Folgende Variablen sind die Datenvariablen, die für die Speicherung der Informationen zuständig sind.

- private int pic_ID;
- private user relatedUser;
- private int pic_elo;
- private String pic_link;
- private comment pic_firstcomment;
- private Category pic_category;
- private int rateCount;

Der Picture-Controller erstellt die Instanzen und fügt diese dann gleich in die Liste hinzu, wofür die folgenden Variablen noch benötigt werden.

- private picture before;
- private picture next;

Methoden:

Da es sich eigentlich um eine reine Datenklasse handelt, werden keine Methoden benötigt. Trotzdem wurde eine Methode entwickelt, die die Mitzählt, wenn ein Bild bewertet wurde, um Programmstrukturen an anderen Stellen zu erleichtern.

- public void countRateCount()

COMMENT

Beschreibung:

Die Klasse „Comment“ wird erzeugt, wenn ein User ein Bild kommentiert. Hier wird der Inhalt des Kommentares und die dazugehörigen User und Bild gespeichert.

Variablen:
Zu jeder Variable wurden die dazu passenden Get und Set Methoden implementiert.

Folgende Variablen sind die Datenvariablen, die für die Speicherung der Informationen zuständig sind.

- private user autor;
- private String content;
- private picture relatedPic;

Der Picture-Controller erstellt die Instanzen und fügt diese dann gleich in die Liste hinzu, wofür die folgenden Variablen noch benötigt werden.

- private comment next;
- private comment before;

USERCONTROLLER

Beschreibung:

Die Klasse „userController“ ist die zum User gehörende Klasse. Diese ist die Verbindung zwischen der Datenklassen „user“ und dem Interface zum Benutzer. Sie ist der Kopf zu der Liste in der alle User gespeichert werden. Hier passiert alles, was mit den Usern zu tun haben, ob sie erstellt, gelöscht oder geändert werden.

Variablen:

Zu jeder Variable wurden die dazu passenden Get und Set Methoden implementiert. Die Liste kennt ihren ersten und ihren letzten eintrag und zählt die UserID mit.

- private user first;
- private user last;
- private int actualUserID;

Methoden:

public user userCreate(String newMail, String newName, String newPW)

Die Methode erstellt einen neuen User und fügt ihn in die Liste ein, die es verwaltet. Hier wird auch überprüft ob die Liste leer ist, oder schon etwas eingefügt wurde. Des weiteren prüft die Methode ob ggf der übergebene Nutzername der die E-Mail schon verwendet sind. Bei erfolgreichem erstellen eines Users, wird dies in der Konsole ausgegeben.

public int userDelete(user OldUser, picController picctrl)

Hier kann ein Nutzer gelöscht warden, sprich er wird aus der Liste entfernt. Des weiteren werden alle Bilder, die dieser hinzugefüt hat, ebenso aus dem System genommen.

public int changeEmail(user changingUser, String newMail)

Hier wird einem Nutzer eine neue Email zugewiesen. Vorher wird überprüft ob diese Email schon einmal verwendet wurde.

public int changePW(user changingUser, String newPW)

Hier bekommt ein User ein neues Passwort zugewiesen.

public int printUser(user User)

Der aktuelle User wird mit all seinen wichtigen Informationen in der Konsole ausgegeben.

PICCONTROLLER

Beschreibung:

Die Klasse „picController“ ist die zum User gehörende Klasse. Diese ist die Verbindung zwischen der Datenklassen „Picture“ und dem Interface zum Benutzer. Hier passiert alles, was mit den Bilder zu tun hat. Sie hat wiederum kaum Variablen.

Variablen:

Zu jeder Variable wurden die dazu passenden Get und Set Methoden implementiert. Die Liste kennt ihr erstes und ihr letztes Element. Des weiteren wird die aktuelle Bild ID mitgezählt.

- private int actual_picID;
- private picture first;
- private picture last;

Methoden:

public picture picCreate(user Creator, String link)

Hier wird ein neues Bild erstellt, also eine Instanz der Klasse picture erzeugt. Diese wird in die Liste eingefügt. Es werden die passenden Informationen in das Bild gespeichert und das Bild wird seinem Erzeuger zugewiesen, welcher dafür mit Erfahrungspunkten belohnt wird.

public picture picDelete(picture pic)

Hier wird das aktuelle Bild gelöscht und zurückgegeben.

public comment picComment(picture pic, user autor, String newContent)

Wird ein Bild kommentiert wird eine neue Instanz der Klasse Comment erzeugt. Falls es noch keinen vorherigen Kommentar gibt, wird dieser als FirstComment gesetzt, alternativ hinten an die Liste der Kommentare angehängt. Der Kommentierende und der Bildeigentümer erhalten Erfahrungspunkte.

public picture randomPic()

Die Methode gibt ein zufälliges Bild zurück, welches in der Liste des Picture-Controllers gespeichert ist.

public int picRate(user rater)

Die Methode sucht zufällig zwei Bilder aus dem Picture Controller und bietet diese einem Nutzer zum bewerten an. Aktuell wird immer das erste Bild als besser bewertet, da es keine Kommunikation mit dem User vor dem PC gibt und es sich so besser testen ließ. Der bewertende User und der Besitzer des Bildes kriegen erfahrungspunkte. Die Elo des besseren Bildes steigt und die des schlechteren sinkt.

public int picRateWithPics(user rater, picture pic1, picture pic2)

Diese Funktion ist sehr ähnlich zu der vorherigen, nur dass die Bilder nicht zufällig rausgesucht werden, sondern übergeben werden.

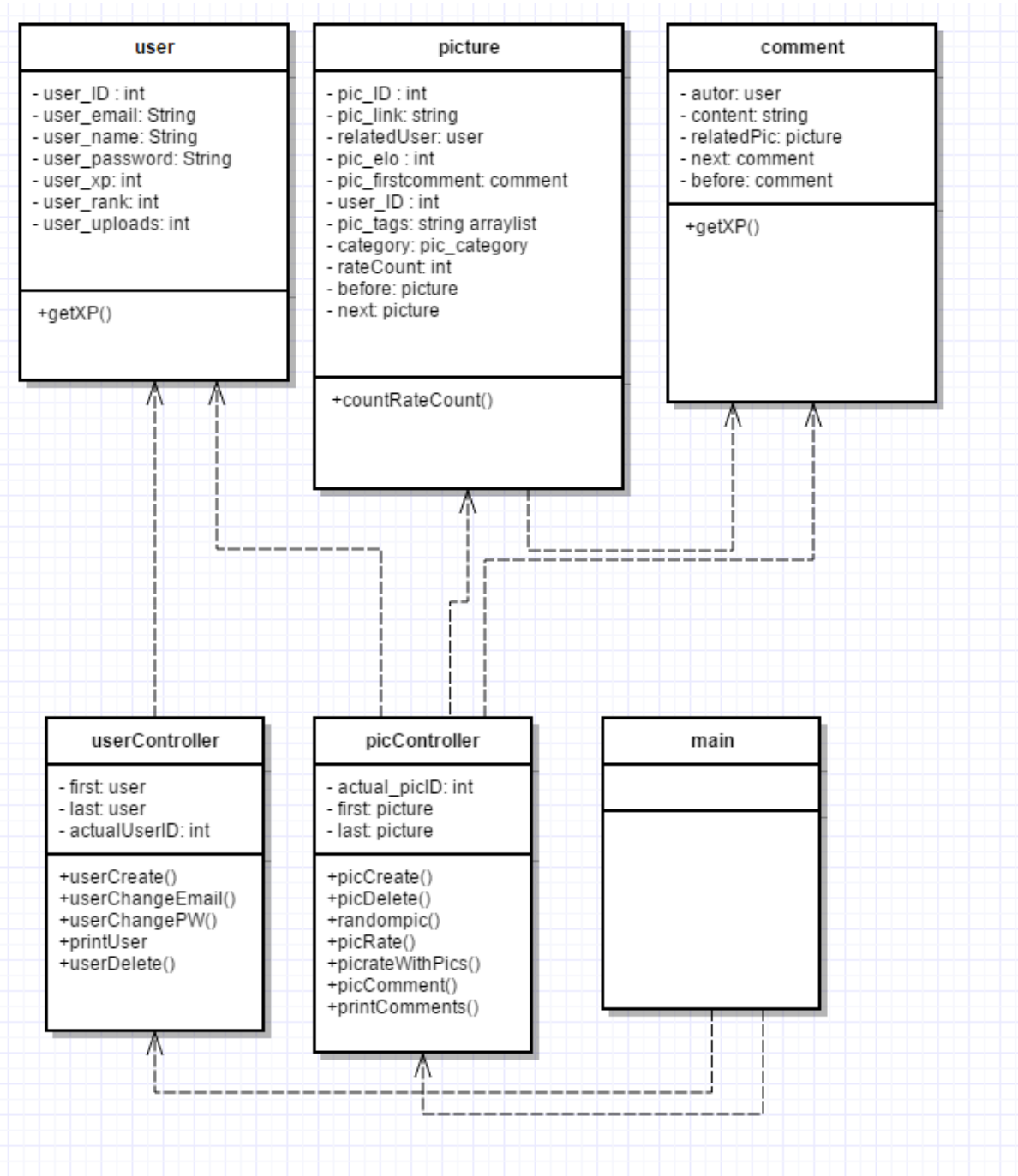
public int printComments(picture pic)

Hier werden alle Kommentare des übergebenen Bildes ausgegeben.

TESTKLASSEN

Des weiteren wurden noch für jede Klasse eine Testklasse angelegt, in der jede Funktion mit halbwegs ausreichender Intelligenz nach bestem Wissen und Gewissen getestet wird.

UML Diagramm



Sonstiges:

Bei der Ausführung/Installation sind keine besonderen Dinge zu beachten