

Teil 1: Design und Implementierung eines Objektmodells

Es soll eine Mehrbenutzer Todo-Listen-Anwendung in Java entstehen, die die Daten in eine Datei persistiert und auf einer graphischen Benutzeroberfläche ansprechend und gut bedienbar darstellt. In einem ersten Schritt soll das Objektmodell der Todo-Listen-Anwendung entstehen. Dieses muss die folgenden Anforderungen erfüllen:

- Darstellung von Benutzern
 - Benutzername (evtl. E-Mail?)
 - Passwort
- Verwaltung von Aufgaben in verschiedenen „Todo“ Listen
 - Benutzer dem die Aufgabe gehört
 - Titel der Aufgabe
 - Status „Erledigt“: ja/nein
 - Status „Priorisiert“: ja/nein
 - Kommentar
 - Zugehörigkeit von einer Aufgabe zu einer „Todo“-Liste
 - Ein Datum wann die Aufgabe fällig ist
- Eine „Todo“-Liste
 - Benutzer dem eine „Todo“-Liste gehört
 - Namen
 - Eine Anzahl von Aufgaben
 - Es sollte eine Default-Liste geben (Aufgaben ohne zugewiesene Liste)
- Arbeiten mit Aufgaben
 - Anlegen von Aufgaben/Listen
 - Löschen von Aufgaben/Listen
 - Zuweisen von einer Aufgabe zu einer Liste
 - Modifizieren von Aufgaben/Listen (z.B. erledigt setzen, Texte bearbeiten)

Im Hinterkopf behalten für eventuelle Erweiterung

- Zusätzliche Daten der Serienaufgaben
 - Dauer der Serie¹ oder unbegrenzt
 - Serienmuster²

Das Objektmodell der Todo-Liste soll zuerst mittels UML designt werden. Anschließend soll das Objektmodell in Java implementiert werden. Dabei ist die Klassenbibliothek von Java immer einzusetzen, wenn dies möglich ist. Grundlegend soll das Objektmodell die Daten gut kapseln, das heißt keinen direkten Zugriff auf die einzelnen Attribute erlauben, sondern stattdessen den Zugriff über get/set-Methoden zur Verfügung zu stellen.

Vorlage: www.wunderlist.com

¹ Anzahl der Termine festlegen (spätere Erweiterung hier möglich)

² Details zum Serienmuster (spätere Erweiterung hier möglich):

- Wöchentlich: Jede / Alle <n> Woche am <Montag, Dienstag, ...>
- Jährlich: Jeden <n>. <Januar, Februar, ...> (z.B. Geburtstage)