

Arzneiverwaltung

In Apotheken ersetzen immer häufiger Computersysteme mit angeschlossenen Roboterarmen die klassischen langen Schieberegale. Das System übernimmt hierbei automatisiert das Suchen und Ausgeben der Arzneien sowie das Ermitteln von knappen Arzneien, damit diese rechtzeitig nachbestellt werden können. Im Folgenden soll eine vereinfachte Arzneiverwaltung modelliert werden.

Bei den Arzneien wird der eindeutige Name der Arznei, die Anzahl der vorrätigen Packungen, ihr Preis und die Verschreibungspflicht erfasst. Der Name wird beim Anlegen einer neuen Arznei festgelegt und muss anschließend nicht mehr geändert werden. Die Anzahl der Packungen einer Arznei soll über eine Methode *ändere_anzahl_um(...)* um einen bestimmten Wert erhöht bzw. verringert werden können. Der Preis und die Verschreibungspflicht sollen über geeignete Methoden gesetzt und alle Eigenschaften sollen über geeignete Methoden abgefragt werden können.

Die Arzneiverwaltung verwaltet die Arzneien als Liste, verfügt über eine Methode *ausgeben(...)* zum Ausgeben einer einzelnen Packung einer Arznei mit einem bestimmten Namen (Material), eine Methode *auffüllen(...)* zum Auffüllen einer bestimmten Arznei mit einer Anzahl neuer Packungen, einer Methode *aufnehmen(...)* zum Aufnehmen einer neuen Arznei in das System und einer Methode *löschen(...)*, um eine Arznei aus dem Sortiment zu löschen.

Aufgaben

- 1.1 Modellieren Sie die Klassen *Arzneiverwaltung* und *Arznei* mit allen Attributen und Methoden inklusive Parametern als UML-Klassendiagramm. (11 BE)
- 1.2 Erläutern Sie die von Ihnen ausgewählte Klassenbeziehung. (2 BE)
- 2.1 Analysieren Sie die Methode *ausgeben(...)* aus dem Material. (8 BE)
- 2.2 Implementieren Sie die Methode *löschen(...)* der Klasse *Arzneimittel*, die anhand des Arzneinamens eine Arznei aus der Liste löscht. (4 BE)
- 2.3 Das System soll Arzneien ermitteln, deren Bestand geringer als ein festgelegter Grenzwert ist. Implementieren Sie eine Methode *suche_knappe_arzneien(...)* in der Klasse *Arzneiverwaltung*. Dieser soll ein gewünschter Grenzwert übergeben werden und sie soll eine Liste der gesuchten Arznamen zurückgeben. (7 BE)

- 3 Um ein schnelleres Auffinden einer Arznei zu ermöglichen, werden die Arzneien in regelmäßigen Abständen alphabetisch nach Arznamen sortiert.
- 3.1 Beschreiben Sie, wie nach dem Sortierverfahren Bubblesort die Arzneien alphabetisch nach Arznamen sortiert werden.
(7 BE)
- 3.2 Implementieren Sie die Methode *sortiere_arzneien()* der Klasse *Arzneiverwaltung*, die die Arzneien alphabetisch nach Arznamen sortiert.
Hinweis: Die Methoden *sort()* und *sorted()* für Listen sollen in dieser Aufgabe nicht verwendet werden.
(8 BE)
- 3.3 Beschreiben Sie einen rekursiven Algorithmus, welcher in einer sortierten Arzneiliste mittels binärer Suche eine bestimmte Arznei anhand ihres Arznames sucht und ihre Position in der Liste zurückgibt bzw. den Wert -1 für den Fall, dass die gesuchte Arznei nicht in der Arzneiliste gefunden wird.
(6BE)
- 3.4 Vergleichen Sie die lineare Suche mit der binären Suche bezüglich der Laufzeit.
(2 BE)
- 4 Bei der Arzneiverwaltung ist das Haltbarkeitsdatum der Arzneien von großer Bedeutung. Die Überwachung der Haltbarkeitsdaten soll ebenfalls vom Computersystem übernommen werden. Hierzu müssen zu jeder Packung einer Arznei die Packungsnummer und das Haltbarkeitsdatum gespeichert werden.
Erläutern Sie eine Möglichkeit, wie Sie Ihre Modellierung ändern können, um zusätzlich das Haltbarkeitsdatum jeder Packung einer Arznei zu speichern.
(5 BE)

Material**Die Methode *ausgeben(...)* der Klasse *Arzneiverwaltung***

```
01 def ausgeben(self, arzneiname: str) -> bool:
02     for arznei in self.__arzneien:
03         if arznei.get_name() == arzneiname:
04             if arznei.get_anzahl() > 0:
05                 arznei.ändere_anzahl_um(-1)
06                 return True
07             else:
08                 return False
09     return False
```