

Aufgabe Onlinehandel

Arbeitszeit 60 min

Einleitung:

In dieser Aufgabe ist schrittweise ein Suchbaum für die Produkte eines Onlinehandels zu programmieren. Dabei sind die Produkte im Suchbaum nach **Artikelnummern** sortiert.

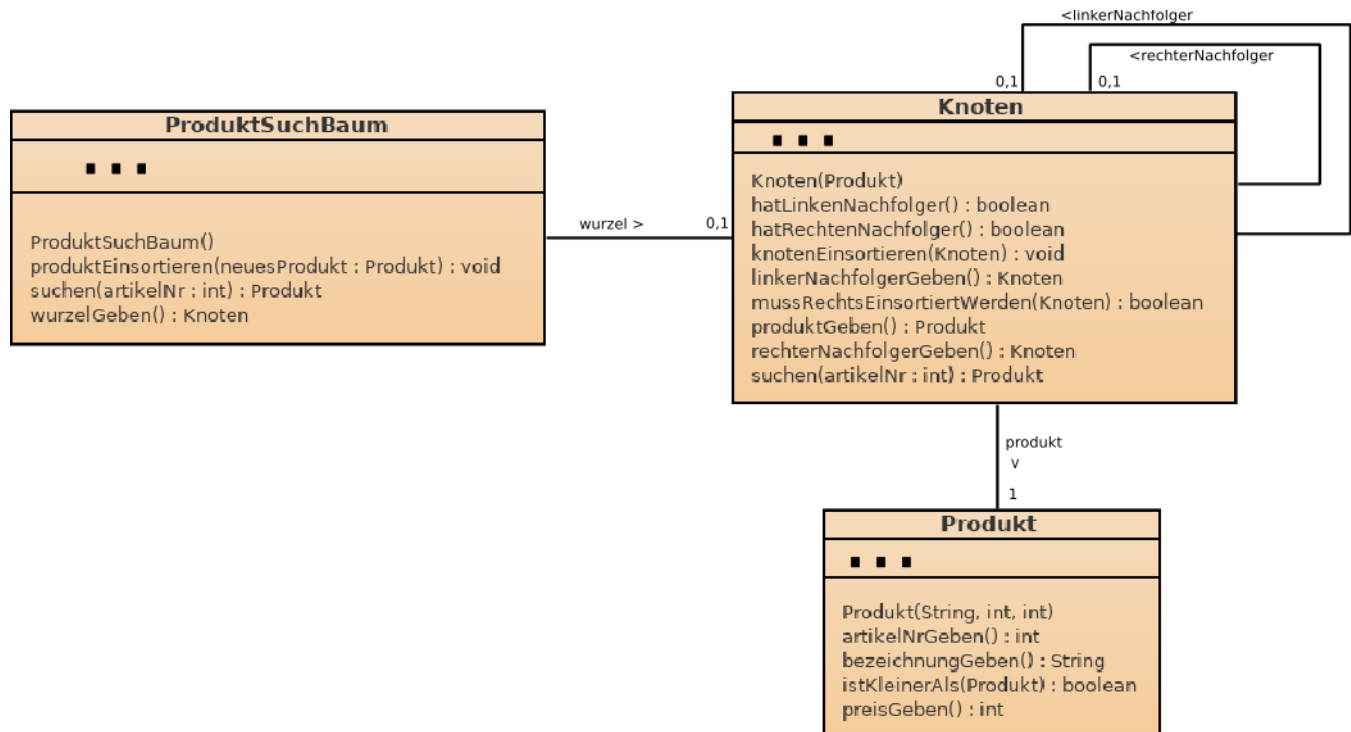
Anmerkungen:

Für alle Aufgaben sind Tests vorhanden, die nur korrekt funktionieren, wenn die Klassen und deren Methoden korrekt benannt werden. Die Tests für die Aufgaben können einzeln (Rechtsklick -> TestAll) oder im Gesamten ausgeführt werden. Tests, die momentan nicht kompiliert sind, werden nicht ausgeführt.

Die Aufgaben können nicht dadurch erfüllt werden, die Tests zu manipulieren! Daher sollten die Tests einfach unverändert bleiben.

Stimmige Lösungsansätze werden auch dann positiv gewertet, wenn sie die Tests nicht erfüllen!

Am Ende gilt folgendes Klassendiagramm:



Aufgabe 1:**4+3 BE**

Ein Produkt soll durch ein Objekt der Klasse **Produkt** repräsentiert werden. Dabei werden bei Aufruf des Konstruktors der Produktbezeichner (String), eine Artikelnummer (int) und der Preis (in Cent, daher auch int) in dieser Reihenfolge übergeben. Über die Methoden **String bezeichnungGeben()**; **int artikelNrGeben()** und **int preisGeben()** kann man diese Daten wieder abrufen.

1a) Implementiere diese Klasse wie angegeben.

1b) Erstelle die Methode **boolean istKleinerAls(Produkt anderesProdukt)**, die genau dann *true* zurück gibt, wenn die eigene Artikelnummer kleiner ist als die des anderen Produkts.

Aufgabe 2:**4+9+6 BE**

Die Struktur des Baums wird durch Objekte der Klasse "Knoten" repräsentiert.

2a) Implementiere den Konstruktor der Klasse Knoten derart, dass er ein Objekt der Klasse Produkt als Parameter empfängt. Schreibe ebenfalls die Methode **produktGeben()**, die es wieder ausgibt.

2b) Vorbereitend sind zwei Referenzattribute (Nachfolgeknoten) und fünf Methoden zu implementieren:

- **linkerNachfolgerGeben()** bzw. **rechterNachfolgerGeben()** gibt den entsprechenden Knoten aus.
 - **hatLinkenNachfolger()** bzw. **hatRechtenNachfolger()**, beide boolean, geben genau dann *true* zurück, wenn ein linker bzw. rechter Nachfolgeknoten existiert.
 - **mussRechtsEinsortiertWerden(Knoten andererKnoten)**, ebenfalls boolean, gibt genau dann *true* zurück, wenn das Produkt des anderenKnotens eine größere Artikelnummer hat, als das eigene Produkt.
- 2c) Weitere Knoten können mit der rekursiven Methode "**knotenEinsortieren(Knoten neuerKnoten)**" in die bestehende Struktur korrekt einsortiert werden. Implementiere diese Methode.

Aufgabe 3:**7+7 BE**

Die Klasse "**ProduktSuchBaum**" verwaltet die Produkte in einem binären Suchbaum.

3a) Implementiere diese Klasse mit folgenden Methoden:

- Der Konstruktor erzeugt einen leeren Baum.
- Die Methode **produktEinsortieren(Produkt neuesProdukt)** erzeugt einen Knoten für das hinzuzufügende Produkt und sortiert es in den Baum ein.
- Die Methode **wurzelGeben()** gibt den Knoten, der die Wurzel des Baums bildet, zurück.

3b) Die Methode **Produkt suchen(int artikelNr)** gibt das im Baum gespeicherte Produkt mit der gegebenen Artikelnummer aus. Falls kein Produkt mit dieser Artikelnummer vorhanden ist, wird *null* zurückgegeben.