BinarySearchTree

Mit der Klasse BinarySearchTree aus den Abiturvorgaben lässt sich einfach ein binärer Suchbaum realisieren. Lesen Sie sich dazu die Beschreibung der Klasse durch und bearbeiten Sie folgende Aufgaben:

Aufgabe 1

Beschreiben Sie, wie mit einem Suchbaumobjekt der Klasse `BinarySearchTree` gearbeitet werden muss, wenn Elemente eingefügt oder gesucht werden sollen.

Aufgabe 2

Vergleichen Sie das Prinzip der Arbeitsweise des `BinarySearchTree` für das Einfügen und Suchen mit den Klassen `BinaryTree` und `List` aus den Abiturvorgaben. Gehen Sie dabei besonders darauf ein, mit wie vielen Objekten kommuniziert werden muss, die für die Datenstruktur verantwortlich sind.

Aufgabe 3

Während andere Abiturklassen für Datenstrukturen generisch für beliebige Datentypen sind, gibt es beim der Klasse `BinarySearchTree` die Einschränkung, dass die Inhaltsklasse das Interface `ComparableContent` implementieren muss. Schauen Sie sich dieses Interface an und zählen Sie die notwendigen Möglichkeiten auf, die an die Inhaltsobjekte gestellt werden. Begründen Sie diese Notwendigkeit.

Aufgabe 4

Begründen Sie, weshalb auf die Methoden `getContent()`, `getLeftTree()` und `getRightTree()` für die oben beschriebene Arbeitsweise mit der Klasse `BinarySearchTree` verzichtet werden kann. Geben Sie mit an, für welche Arbeitsweise diese Methoden benötigt werden.

Aufgabe 5

Erstellen Sie für den binären Suchbaum eine Klasse `Waren`, die nur eine ID als Zahl und den Namen der Ware speichert. Für die Vergleiche nehmen Sie nur die ID. Füllen Sie auf dieser Grundlage einen Suchbaum mit verschiedenen Waren und suchen Sie anschließend nach diesen Waren nur mit der Angabe der ID. Überprüfen Sie das Ergebnis der Suche durch die Ausgabe des Namens der zurückgegebenen Ware.

★ Zusatzaufgabe 1

Implementiere Sie eine Methode zur Traversierung des Suchbaumes und überprüfen Sie damit seine Füllung.

Lösungen zu den Aufgaben

Lösung 1

Beim Einfügen muss nur die Methode `insert` mit dem einzufügenden Objekt aufgerufen werden. Für das Suchen gibt es die Methode `search`, der man ein Objekt übergeben muss, bei dem die Attribute zum Gesuchten identisch sind, die für den Vergleich herangezogen werden.

Lösung 2

Während das Einfügen an die passende Stelle und das Suchen nach einem Objekt beim `BinaryTree` und `List` selbst durchgeführt werden muss, gibt es beim `BinarySearchTree` bereits entsprechende Methoden die diese Aufgabe übernehmen. Deshalb muss man hier auch nicht den Baum bzw. die Liste durchgehen, um an die gewünschten Stellen zu gelangen.

Lösung 3

Da der binäre Suchbaum wissen muss, wie die Inhaltsobjekte miteinander verglichen werden können, benötigt man entsprechende Methode, die genau dieses definieren. Im Falle von `ComparableContent` sind das die Überprüfung auf Gleichheit, sowie die der Vergleich auf größer und kleiner.

Lösung 4

Da es Methoden zum Suchen, Einfügen und Löschen gibt, muss man nicht selber auf dem Suchbaum operieren. Diese Methoden werden aber benötigt, wenn man z. B. den Baum traversieren will.