

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:

1. Beschreiben Sie, wie mit einem Suchbaumobjekt der Klasse `BinarySearchTree` gearbeitet werden muss, wenn Elemente eingefügt oder gesucht werden sollen.
2. Vergleichen Sie das Prinzip der Arbeitsweise des `BinarySearchTree` für das Einfügen und Suchen mit den Klassen `BinaryTree` und `Stack` aus den Abiturvorgaben. Gehen Sie dabei besonders darauf ein, mit wie vielen Objekten kommuniziert werden muss, die für die Datenstruktur verantwortlich sind.
3. Die Klasse `BinarySearchTree` arbeitet im Gegensatz zur Klasse `Stack` oder `BinaryTree` nicht mit Inhaltsobjekten der Klasse `Object`, sondern mit Inhaltsobjekten der Klasse `Item`. Zählen Sie Eigenschaften auf, die Objekte der Klasse `Item` besitzen müssen und begründen Sie diese.
4. Die Beschreibung der Methode `getItem()` ist sehr knapp gehalten. Deshalb ist hier ihr Quelltext abgebildet:

```
class BinarySearchTree {  
  
    private BinaryTree bintree;  
  
    public Item getItem() {  
        if (this.isEmpty())  
            return null;  
        else  
            return (Item) bintree.getObject();  
    }  
  
    ...  
}
```

Erstellen Sie auf dieser Grundlage eine verbesserte Beschreibung der Methode, die darauf eingeht, wozu die Methode genutzt werden kann und wozu nicht.

5. Begründen Sie, weshalb auf die Methoden `getItem()`, `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.



Lösungen:

BinarySearchTree:

- 1.
- 2.
- 3.

