

## Testat 6 – 1

- Achte darauf, dass die vorgegebene Signatur in Deiner Lösung exakt eingehalten wird.
- Vervollständige die Datei `CharacterSearchTree.java` ohne eine `package`-Angabe.
- Gib nur die Datei mit dem Namen `CharacterSearchTree.java` im Moodle-Bereich ab.
- Die Bearbeitungsdauer beträgt 60 Minuten.
- Die Bearbeitungszeit beginnt mit der Ausgabe des Testats an Dich.

Bei diesem Testat erhältst Du drei Klassen:

- Die Klasse `CharacterSearchTree` ist bekannt und soll von Dir **ergänzt** werden.
- Die Klasse `HuffmanTriple` ist bekannt und soll von Dir nur benutzt werden.
- In der Klasse `Testumgebung` findest Du einen einfachen Test für die zu implementierende Methode. Die Klasse `Testumgebung` darfst Du beliebig ändern. Sie wird nicht abgegeben.

Ergänze in der Klasse `CharacterSearchTree` die Methode `int apply( int i )`.

Die Methode `int apply( int i )` soll Folgendes leisten:

- Die Methode `int apply( int i )` soll die Anzahl der im Baum vorkommenden Knoten bestimmen, für die die folgenden Bedingungen gelten:
  - der Knoten darf *keinen rechten* Nachfolgeknoten besitzen und
  - der Wert von `quantity` muss kleiner als der Wert von `i` sein.
- Es dürfen keine Attribute und keine weiteren Methoden angelegt werden.
- Die Methode `apply` muss in einem Programm mehrfach nacheinander aufgerufen werden können und bei jedem Aufruf das entsprechend der Aufgabenbeschreibung korrekte Ergebnis zurückgeben.

*Hinweis:*

Die Methode `main` in der Klasse `Testumgebung` führt einen Test für die Methode `apply` durch.

Bei einer korrekten Implementierung der Methode `apply` liefert der Aufruf für den in der Klasse `Testumgebung` bereits zur Verfügung gestellten Baum `bigTree` folgende Ergebnisse:

`apply( 3 )` gibt das Ergebnis 6 zurück – gezählt werden die Knoten `'b'`, `'f'`, `'h'`, `'k'`, `'u'` und `'x'`.

`apply( 5 )` gibt das Ergebnis 8 zurück – gezählt werden die Knoten `'b'`, `'f'`, `'h'`, `'k'`, `'n'`, `'p'`, `'u'` und `'x'`.

Der Baum `bigTree` besitzt die folgende Struktur:

