

Name: \_\_\_\_\_

Tutor: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

Punkte: \_\_\_\_\_

Gruppe: \_\_\_\_\_

Abgabe am: Di, 04. 04. 2006 12:00

---

**Wortliste als Binärbaum (Punkte: 24; Teilaufgaben: a=10, b=4, c=10)**

Um die Wortliste zu beschleunigen soll sie als binärer Suchbaum implementiert werden.

- a) Implementieren Sie einen binären Suchbaum (in der Klasse *WordList2*) laut Spezifikation in Übung 1. Testen Sie die Funktionalität.
- b) Testen Sie die Effizienz Ihrer Lösung. Welche Methoden werden wie viel schneller ausgeführt als bei der Übung 1? Gibt es auch Methoden, die gleich schnell/langsamer ausgeführt werden?  
Überlegen Sie (schriftlich!) die Gründe für die jeweiligen Änderungen der Effizienz der einzelnen Methoden. Wodurch ergeben sich die Unterschiede?
- c) Implementieren Sie das Balancieren des Binärbaumes. Das Balancieren soll dabei im Hintergrund erfolgen (d.h. der Benutzer der Wortliste soll sich darum nicht kümmern müssen).  
Überlegen Sie (schriftlich?): Von wo aus soll die Methode *balance* aufgerufen werden? Wie oft/unter welchen Kriterien müsste *balance* aufgerufen werden, damit es Vorteile bringt? Testen Sie die Wortliste mit und ohne balancieren. Werden die Erwartungen erfüllt?