

## Übungsblatt 10

Abgabe für ESE: bis Donnerstag, den 15. Januar um 10:00 Uhr

Abgabe für IEMS: bis Donnerstag, den 29. Januar um 10:00 Uhr

### Aufgabe 1 (14 Punkte)

Implementieren Sie eine Klasse *BinarySearchTree*, die einen binären Suchbaum mit Schlüsseln vom Typ *int* und Elementen von beliebigem Typ realisiert. Falls es Schwierigkeiten mit den Java-Generics, bzw. C++ templates gibt, können Sie für den Datentyp stattdessen einen String nutzen.

Designvorschlag für die Klasse + Details zur Spezifikation siehe Vorlesungs-Homepage. Beachten Sie, dass Sie nur *insert* (6 Punkte) und *lookup* (4 Punkte), aber kein *remove* implementieren müssen, und dass Sie die doppelt verkettete Liste zwischen den Knoten des Baumes weglassen können.

Schreiben Sie wie immer (sinnvolle) Tests für Ihre Methoden. Schreiben Sie dazu eine sinnvolle *toString* Methode (4 Punkte). Es reicht, wenn Sie *insert* und *lookup* zusammen beim Aufbauen eines nicht-trivialen Baumes testen. Achten Sie aber darauf, dass während des Tests insgesamt jede Zeile dieser beiden Funktionen mindestens einmal ausgeführt wird.

### Aufgabe 2 (6 Punkte)

Schreiben Sie ein Programm *BinarySearchTreeMain*, das die Zahlen  $1, 2, \dots, n$  in dieser Reihenfolge in einen zu Beginn leeren *BinarySearchTree* einfügt. Fügen Sie dann in einen neuen, wiederum zu Beginn leeren *BinarySearchTree* diese  $n$  Zahlen in zufälliger Reihenfolge ein.

Messen Sie die Laufzeit und die Tiefe des Baumes (jeweils nach den  $n$  Einfügungen) dieser beiden Varianten für  $n = 2^{10}, 2^{11}, 2^{12}, \dots$  (gehen Sie so hoch, dass die Rechenzeit auf Ihrer Maschine noch erträglich ist). Diskutieren Sie die Ergebnisse kurz in Ihrer *erfahrungen.txt*.

Committen Sie wie gehabt Ihren Code in das SVN, in einen neuen Unterordner *uebungsblatt\_10*, und ebenso Ihr Feedback in einer Textdatei *erfahrungen.txt*. Insbesondere: Wie lange haben Sie ungefähr gebraucht? An welchen Stellen gab es Probleme und wieviel Zeit hat Sie das gekostet?