

Übungsblatt 9

Abgabe bis Dienstag, den 17. Januar um 16 Uhr

Aufgabe 1 (12 Punkte)

Implementieren Sie eine Klasse *BinarySearchTree*, die einen binären Suchbaum mit Schlüsseln vom Typ *int* und Elementen vom Typ *Object* realisiert. Ein Gerüst der Klasse mit Kommentaren zu den einzelnen Methoden und Membervariablen ist auf dem Wiki verlinkt. Beachten Sie, dass Sie nur *insert* und *lookup*, 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. Sie können die *insert* und *lookup* Methode zusammen testen, und es reicht wenn Sie diese beiden Methoden beim Aufbau eines Baumes testen. Achten Sie aber darauf, dass Sie die Elemente in einer solchen Reihenfolge einfügen, dass jede Zeile Ihrer Implementierung der beiden Methoden während des Tests wenigstens 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 und dann die Tiefe des Baumes (nach den n Einfügungen) ausgibt. Fügen Sie dann in einen neuen, wiederum zu Beginn leeren *BinarySearchTree* diese n Zahlen in zufälliger Reihenfolge ein, und geben Sie die Tiefe nach diesen n Einfügungen aus. Wählen sie $n = 2^{16} = 65\,536$ oder so groß, dass die Rechenzeit auf Ihrer Maschine noch erträglich ist.

Schauen Sie, dass die Tiefen mit den in der Vorlesung vorhergesagten Werten ungefähr übereinstimmen. Warum dauert das Einfügen in dem einen Fall viel länger als in dem anderen?

Aufgabe 3 (2 Punkte)

Committen Sie alles wie gehabt in das SVN und stellen Sie sicher, dass auf Jenkins alles kompiliert, und die Tests und checkstyle ohne Fehler durchlaufen. Und vergessen Sie nicht Ihre *erfahrungen.txt* zum Übungsblatt und zur Vorlesung.